2013

Gödel's incompleteness theorem

Christopher Mullins
Eastern Washington University

GÖDEL'S INCOMPLETENESS THEOREM

A Thesis

Presented To

Eastern Washington University

Cheney, Washington

In Partial Fulfillment of the Requirements

for the Degree

Master of Science

By

Christopher Mullins

Spring 2013

THESIS OF CHRISTOPHER MULLINS APPROVED BY

____ DATE:________

W. DALE GARRAWAY, GRADUATE STUDY COMMITTEE

____ DATE:________

RONALD GENTLE, GRADUATE STUDY COMMITTEE

____ DATE:________

PARTHA SIRCAR, GRADUATE STUDY COMMITTEE

Abstract

This thesis gives a rigorous development of sentential logic and first-order logic as mathematical models of humanity's deductive thought processes. Important properties of each of these models are stated and proved including Compactness results (the ability to prove a statement from a finite set of assumptions), Soundness results (a proof given a set of assumptions will always be true given that set of assumptions), and Completeness results (a statement that is true given a set of assumptions must have a proof from that set of assumptions). Mathematical theories and axiomatizations or theories are discussed in a first-order logical setting. The ultimate aim of the thesis is to state and prove Gödel's Incompleteness Theorem for number theory.

Acknowledgements

First and foremost, I would like to acknowledge the one true God of the Bible for His great grace and mercy for the insight, understanding, wisdom, speed, and accuracy in writing this thesis. I have asked for His help throughout this project, and He has given His help in abundance beyond what I could imagine or deserve. All truth is God's truth.

Second, I would like to thank my loving and supportive parents Bill and Tammy Mullins who sacrificed so much to give me such a superb educational foundation by homeschooling me. I owe any and all academic success to them.

Third, I would like to thank the love of my life, Ashley Murray (soon to be Mullins) for her unconditional support and encouragement in this massive and time-consuming project. Thank you Love for being so understanding and patient and for being my continual cheerleader and supporter. We first started dating after I had chosen the topic for this thesis, and you have stuck by me as my best friend throughout the whole process. Thank you so much! I love you!

Lastly, I would like to thank Dr. Dale Garraway for his critical role as my thesis advisor. There were many times when I thought that I would not be able to get this thesis done on time, but you always encouraged me and gave me helpful feedback and helpful perspectives from which to analyze things. Thank you so much!

Contents

Chapter 1

Introduction

Statements that are true yet unprovable? Such a notion seems to be self-contradictory and yet that is exactly what Gödel's Incompleteness Theorem asserts.

Gödel's Incompleteness Theorem *Given a decidable set of first-order sentences that are true in number theory, then there is a statement true in number theory that is not provable from our decidable set.*

Of course there are terms in this statement that have as yet to be defined, but this statement gives the reader a first sense of the result that this thesis seeks to prove. Think of the set of sentences that the theorem mentions as the center of a circle in the plane. Think of the radius of that circle being associated with the extent of what can be proven from those sentences. Finally think of the entire plane representing the totality of all statements true of number theory. Then what Gödel's Theorem says is that the radius of the circle must be less than infinity.

It is clear that Kurt Gödel's Incompleteness Theorem states not only a mathematical result but also touches on meta-mathematics. It raises deep philosophical questions. How do we know mathematical truth? How do we know that we can prove true statements? Is mathematics realism or formalism?

Kurt Gödel went to university in Europe in the interim period between World Wars I and II. The philosophical air of logical positivism, whose foundational axiom is "Man is the measure of all things," abounded in Europe during this period. Hence, it is no surprise that mathematicians themselves were mostly formalists who believed that all of mathematics can be reduced to rule following. There is no objective reality of mathematical objects. There is no need to appeal to intuition.

David Hilbert was one of formalism's most vocal proponents in mathematics, and his desire was to spur his colleagues to the systematic formalizing of all mathematical branches. Among his 10 problems that he submitted to the mathematical community in 1900 was the goal of proving arithmetic consistent. This proof would basically demonstrate that arithmetic could be treated as a self-contained formal system without reference to anything else. Kurt Gödel's Incompleteness Theorem destroyed Hilbert's dream.

Gödel himself did not share the mathematical mainstream view. Instead Kurt Gödel was a Platonist believing that mathematical objects exist in reality with the mathematical project a project of discovery and not a meaningless exercise in reasoning and rule following. Gödel first announced that he had a proof of a version of the Incompleteness Theorem in 1930 in a conference in Königsberg (a conference at which Hilbert was present). Many at the conference paid no heed to the meek Gödel's understated announcement. One person of interest who did take a great deal of notice was John von Neumann, one of the patriarchs of computability theory. Von Neumann saw the application of Gödel's Theorem quite quickly.

Although one of the corollaries to the Incompleteness Theorem says that the consistency of arithmetic cannot be proved within arithmetic, solving in the negative one of Hilbert's problems and smashing the formalist enterprise, many took very little heed. In a strange twist, the logical positivists and post-modernists usurped Gödel's result as supporting their positions of relative knowledge and meaning. This conception continues to this day. Gödel believed that his work supported just the opposite view: that objective mathematical truth exists. (Note the author obtained much of the historical information given above from *Incompleteness–the Proof and Paradox of Kurt Gödel* by Rebecca Goldstein.)

Many have cited astonishing philosophical consequences of Gödel's Theorem including an intriguing argument that the mind must be more than a computer. To fully understand what Gödel's Incompleteness Theorem says and to judge the soundness of such claims we proceed on a journey through mathematical logic to prove this weighty theorem.

Gödel's Theorem is a statement about the mathematical model of logic called "first-order logic." In this thesis, we seek to establish a rigorous mathematical development of sentential and first-order logic as models of humanity's deductive thought processes.

As we proceed through the development of these two mathematical models of logic, we will state, prove, and discuss several important results, important not only for Gödel's Theorem but also powerful results in their own right. Among these results are soundness, completeness, and compactness results for both models.

After developing the model of first-order logic, we will discuss what formal theories are and what the formal axiomatization of a theory means. We will give concrete examples of these concepts with familiar mathematical structures on our journey to prove Gödel's Incompleteness Theorem.

Finally, as the main goal of this thesis, we will give a proof of Gödel's Incompleteness Theorem, discussing the concepts of Gödel numbering and representability within number theory which are necessary to prove the theorem. The development of this thesis closely follows that of Herbert Enderton in *A Mathematical Introduction to Logic*. Because this is the case, citation is suppressed for ease in reading.

Chapter 2

Sentential Logic: The Language and Truth Values

In our journey to prove Gödel's famous theorem, we must rigorously establish the mathematical system, first-order logic, in which it is proved. As a prototype to first-order logic, in this chapter we rigorously develop the mathematical system of sentential logic.

2.1 Intuition Behind the Sentential Language

Our goal behind creating a formal language is to create a model for humanity's deductive thought processes. In the real world, as reasoners, we may put forward the following arguments:

If Socrates is a man, then he is mortal.

Socrates is a man.

Therefore, Socrates is mortal.

If Gertrude is a purple oliphant, then it eats grapes.

Gertrude is a purple oliphant.

Therefore Gertrude eats grapes.

If it is not the case that you are reading this sentence and you are confused, then you should re-read it.

I am confused and I am no longer reading the former sentence.

Therefore, I should re-read the first sentence.

Now, as reasoners we can see that the first two arguments make sense *regardless* of whether men exist, Socrates, or purple oliphants named Gertrude. What makes the first two arguments understandable is the inherent structure of the propositions. In a very real way, what it *means* to be Socrates, a man, or a purple oliphant named Gertrude are inconsequential and merely cloud our thinking when we attempt to decide whether the argument is reasonable. Thus, it would be handy to have a formal system, with all the rigor of mathematics, that can capture the essential structure of arguments that we may make.

For instance $\{(\boldsymbol{A} \rightarrow \boldsymbol{B}), \boldsymbol{A}\} \vDash \boldsymbol{B}$ captures the essential structure of the first two arguments where we accept the two expressions in the set as "true" and where "$\vDash$" has an intended meaning of "therefore." So, we can interpret the symbolism to say that given the statements $(\boldsymbol{A} \rightarrow \boldsymbol{B})$ and $\boldsymbol{A}$ the conclusion $\boldsymbol{B}$ follows. Whether we translate "Socrates is a man" and "Man is mortal" as $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively or whether we translate "Gertrude is a purple oliphant" and "Gertrude eats grapes" as $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively, we have captured in just a few symbols the inherent structure in both arguments above.

In addition to capturing the essential structure of human reasoning with no frills, such a formal language to model human deductive thought processes also avoids the ambiguities of natural languages (languages spoken by humans to express thought). In the third argument, understanding the first premise is the key to knowing whether the argument makes sense. We need to know the conditions for re-reading the first sentence, and the stated condition is it not being the case that you are reading the first sentence and you are confused. Now, the problem is, there is ambiguity in the sentence. Does the "not" refer to *both* reading the first sentence and being confused or does the "not" refer to just reading the first sentence. Given how the sentence is written in our natural language (English in this case)–and in a perfectly acceptable way–it could be read with either emphasis. In the first case, either not reading the sentence or not being confused would be a sufficient condition for re-reading the sentence. In the second case, only by not reading the sentence and being confused will be sufficient for re-reading the sentence. Whether we take the intended meaning to be one or the other, the argument still makes sense, since the second premise was that I was no longer reading the sentence, and I was confused, and this premise will be sufficient for both cases. (So, since you, the reader, are no longer reading the first sentence in that argument–you are reading this one!–and you are probably still confused, you should go re-read the first sentence!) The point is that this ambiguity in the natural language clouds the analysis of whether the argument was reasonable.

Now, if we translate "you are reading this sentence" as $\boldsymbol{S}$ and "you are confused" as $\boldsymbol{C}$, and if we have "$\neg$" represent the idea of "not" and "$\wedge$" represent the idea of "and", then the ambiguity we have described above is whether "it is not the case that you are reading this sentence and you are confused" should be translated as $(\neg(\boldsymbol{S} \wedge \boldsymbol{C}))$ or as $((\neg \boldsymbol{S}) \wedge \boldsymbol{C})$. Having even the power to *describe* the ambiguity with just these few symbols indicates in and of itself the ability to also *avoid* such ambiguity if we use such a formal language to model deduction.

How would we analyze the reasonableness of arguments if we have such a nice formal language with which to work? The reader is probably already familiar with symbology we have already used and also the notion of truth tables. I can substitute values of true or false in for each symbol, and for all values that cause the premise expressions to be true, the conclusion expression should also be true. The truth table for the first two arguments above would be

$\boldsymbol{A}$	$\boldsymbol{B}$	$(\boldsymbol{A} \rightarrow \boldsymbol{B})$
T	T	T
T	F	F
F	T	T
F	F	T

The only line in the truth table for which both premises $\boldsymbol{A}$ and $(\boldsymbol{A} \rightarrow \boldsymbol{B})$ are true is the first, and we see that the conclusion, $\boldsymbol{B}$, is also true in this case. So, we would say that our argument is reasonable. All of the foregoing should be familiar to the reader from a mathematical foundations course.

2.2 The Need for Formalizing the Intuition

Since we will want to use the mathematical structure of our formal language to prove mathematical results about logic (ultimately, Gödel's Incompleteness Theorem), we need to develop the intuitive, foundations level ideas in a mathematically rigorous way. Above, we have simply thrown around symbols and ideas without any justification whatsoever. However, to establish our intuitive ideas rigorously, we may ask questions like, "What constitutes a valid formula in our language?", "How do we know that our valid formulas may be read unambiguously?", and "Do we (or will we) have sufficient structure in our language to express any proposition?". We will answer these and other such questions in the next couple of chapters. Note that throughout this thesis, to denote formal language symbols we put all such symbols in a bold typeface.

2.3 Valid Formulas

We begin by establishing the language of sentential logic and precisely describing what a valid formula or grammatically correct statement in that language will be. Note that we assume knowledge of basic set theory which we will use extensively in developing the language.

First, we assume that we are given a countably infinite number (cardinality $\aleph_0$) of symbols to use in the alphabet of our formal language. These symbols are listed in the following table.

Symbol	Type	Intended Meaning
(	Logical Grouping Symbol	
)	Logical Grouping Symbol	
$\neg$	Logical Connective	"not"
$\wedge$	Logical Connective	"and"
$\vee$	Logical Connective	"or"
$\rightarrow$	Logical Connective	"implies"
$\leftrightarrow$	Logical Connective	"is equivalent to"
$\boldsymbol{A_i}$ for each $i \in \mathbb{N}$	Sentence Symbol	

We assume that none of these symbols is a finite sequence of the others e.g. $\boldsymbol{A_{1989}} \neq \neg)(\wedge\vee$.

Definition 2.1 *An* **expression** *in the sentential language is a finite sequence of symbols from the sentential alphabet.*

Example 2.2 *Any symbol from the sentential alphabet is an expression. Each of the following are sentential expressions*

$$\rightarrow))))\boldsymbol{A_{1989}},$$

$$\rightarrow))) \leftrightarrow (((\neg, \quad \textit{and}$$

$$((\neg \boldsymbol{A_1}) \rightarrow (\boldsymbol{A_2} \wedge (\boldsymbol{A_3} \vee \boldsymbol{A_4})))$$

Of course, in our language we only want to consider expressions that have the appropriate structure to translate our natural language propositions. So, for instance, $\rightarrow))))\boldsymbol{A_{1989}}$ cannot translate any meaningful English proposition whereas $(\neg(\boldsymbol{A_1} \wedge \boldsymbol{A_2}))$ can meaningfully translate the proposition "It is not the case that I am both reading this sentence and am confused," if we let

$\boldsymbol{A_1}$ translate the statement "I am reading this sentence," and let $\boldsymbol{A_2}$ translate the statement "I am confused." So, in our language we want to be able to identify "grammatically correct" expressions (ones that can meaningful translate a natural language proposition). We will call these grammatically correct expressions *well-formed formulas* or *wffs* for short. There are two primary ways to define wffs.

We start with a top-down approach. Let $\mathfrak{S}$ be the family of all sets S of expressions in the sentential language that fulfill the following properties: (i): every sentence symbol is in S, and (ii): if expressions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are in S, then so are the expressions $\boldsymbol{(\neg\alpha)}$, $\boldsymbol{(\alpha \wedge \beta)}$, $\boldsymbol{(\alpha \vee \beta)}$, $\boldsymbol{(\alpha \rightarrow \beta)}$, and $\boldsymbol{(\alpha \leftrightarrow \beta)} \in S$.

Definition 2.3 *An expression is a* **well-formed formula (wff)** *if it is an expression in the set* $\bigcap_{S\in\mathfrak{S}} S$

Example 2.4 $\boldsymbol{((\neg A_1) \rightarrow (A_2 \wedge (A_3 \vee A_4)))}$ *is a wff under this definition. Since all of the sentence symbols are in every S (by property (i) of the family), this means that* $\boldsymbol{(\neg A_1)}$ *and* $\boldsymbol{(A_3 \vee A_4)}$ *are also in every S (by property (ii)). Thus,* $\boldsymbol{(A_2 \wedge (A_3 \vee A_4))}$ *is also in every S, and so* $\boldsymbol{((\neg A_1) \rightarrow (A_2 \wedge (A_3 \vee A_4)))}$ *is in every S, hence in the intersection, hence a wff under the definition.*

The idea with this definition is that we should include all expressions that take the form that we think our wffs should have, either a stand alone sentence symbol or one of the five other forms mentioned above, each of which uses one of our logical connective symbols. But, by including any set of expressions with these properties, we have more expressions than we want. For instance, if $\rightarrow \boldsymbol{A_1}\neg$ happens to be in one such S that has the two properties mentioned above, then we get infinitely many garbage expressions like

$\boldsymbol{(A_2 \wedge \rightarrow A_1 \neg)}$. We can eliminate such garbage expressions by using the intersection over all such sets to whiddle down to the smallest set that will have the two properties of each set in the family (that the intersection has the two properties of the family may be quickly verified by the reader).

Now for a bottom-up approach. Let $\mathbb{E}$ designate the set of all expressions using the sentential alphabet. We define several formula building operations on $\mathbb{E}$ and on $\mathbb{E} \times \mathbb{E}$ as follows (note that the bolded parentheses are part of the definition):

$$\mathcal{F}_{\neg}(\boldsymbol{\alpha}) = \boldsymbol{(\neg\alpha)}$$

$$\mathcal{F}_{\wedge}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{(\alpha \wedge \beta)}$$

$$\mathcal{F}_{\vee}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{(\alpha \vee \beta)}$$

$$\mathcal{F}_{\rightarrow}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{(\alpha \rightarrow \beta)}$$

$$\mathcal{F}_{\leftrightarrow}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{(\alpha \leftrightarrow \beta)}$$

Definition 2.5 *An expression is a* **wff** *if there is a finite sequence of expressions*

$$\langle \boldsymbol{\alpha_0}, \boldsymbol{\alpha_1}, \ldots, \boldsymbol{\alpha_n} \rangle$$

where each $\boldsymbol{\alpha_i}$ *is a sentence symbol,* $\boldsymbol{\alpha_i} = F_{\neg}(\boldsymbol{\alpha_j})$ *for* $j < i$*, or* $\boldsymbol{\alpha_i} = \mathcal{F}_{\sharp}(\boldsymbol{\alpha_j}, \boldsymbol{\alpha_k})$ *for* $j \leq k < i$ *and* $\sharp \in \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$.

Example 2.6 $\boldsymbol{((\neg A_1) \rightarrow (A_2 \wedge (A_3 \vee A_4)))}$ *is a wff under this definition.* $\mathcal{F}_{\neg}(\boldsymbol{A_1}) = \boldsymbol{(\neg A_1)}$ *and* $\mathcal{F}_{\vee}(\boldsymbol{A_3}, \boldsymbol{A_4}) = \boldsymbol{(A_3 \vee A_4)}$. *Also,*

$$\mathcal{F}_{\wedge}(\boldsymbol{A_2}, \boldsymbol{(A_3 \vee A_4)}) = \boldsymbol{(A_2 \wedge (A_3 \vee A_4))}, \text{ and}$$

$$\mathcal{F}_{\rightarrow}(\boldsymbol{(\neg A_1)}, \boldsymbol{(A_2 \wedge (A_3 \vee A_4))}) = \boldsymbol{((\neg A_1) \rightarrow (A_2 \wedge (A_3 \vee A_4)))}.$$

So we see that we can apply the formula building operations finitely many times, to three sentence symbols to build the formula under consideration. Thus, it must be a wff.

It turns out that both Definition 2.3 and Definition 2.5 are equivalent, as we should wish them to be. To prove the equivalence, we work with some more general notions from which we can prove the equivalence.

2.3.1 Induction

In a general setting we may be dealing with a set $\mathbb{U}$ and a set of operations (they could be binary, 3-ary, etc.) on $\mathbb{U}$, $\mathfrak{F}$, and we want to find the smallest subset of $\mathbb{U}$ containing some set B of initial elements of $\mathbb{U}$ that is closed under the operations in $\mathfrak{F}$. Think of the set B of initial elements as a basis, and the set we are trying to find is the set generated from the elements in B by the operations in $\mathfrak{F}$. In the context of wffs, $\mathbb{E}$ would be $\mathbb{U}$, the formula building operations the operations in $\mathfrak{F}$, and B would be the set of sentence symbols.

Now, we will simplify the generic situation for ease in grasping the key concepts involved by letting $\mathfrak{F}$ contain only two operations $f : \mathbb{U} \times \mathbb{U} \longrightarrow \mathbb{U}$ and $g : \mathbb{U} \longrightarrow \mathbb{U}$; however, the results that follow apply to any finite set of operations (and even beyond) as will be readily seen.

Definition 2.7 *A subset S of $\mathbb{U}$ is* **closed under** f **and** g *if whenever $x, y \in S$, then $f(x, y), g(x) \in S$.*

Definition 2.8 *A subset S of $\mathbb{U}$ is* **inductive** *if $B \subseteq S$ and S is closed under f and g.*

Example 2.9 *If S is inductive and $a, b \in B$, then b, $f(a,b)$, $f(b,b)$, $g(a)$, $f(g(a),b)$, $g(f(a,b))$, $f(g(f(b,b)), f(a,g(b)))$, etc. are in S.*

Let $\mathfrak{S} = \{S \in \mathcal{P}(\mathbb{U}) : S \text{ is inductive }\}$ and let $C^* = \bigcap_{S \in \mathfrak{S}} S$. In the context of wffs this C^* will be the set of wffs under Definition 2.3.

Theorem 2.10 *C^* is the smallest inductive subset of $\mathbb{U}$.*

Proof: If S is inductive, then $C^* \subseteq S$ by the very definition of C^*. So if C^* is inductive, it is in fact the smallest such inductive set. Since $B \subseteq S$ for every $S \in \mathfrak{S}$, $B \subseteq C^*$ by properties of intersection. Let $x, y \in C^*$, then $x, y \in S$ for every $S \in \mathfrak{S}$ by the definition of intersection and C^*. Since each $S \in \mathfrak{S}$ is inductive and hence closed under f and g, $f(x,y), g(x) \in S$ for every $S \in \mathfrak{S}$. Hence, $f(x,y), g(x) \in C^*$ by definition and thus, C^* is closed under f and g since the choice for $x, y \in C^*$ was arbitrary. Thus, C^* is inductive by definition and is the smallest inductive subset of $\mathbb{U}$ by the first couple of remarks. ∎

This theorem indicates that C^* will be precisely the elements of $\mathbb{U}$ that can be generated from the elements of B. This method gives the top-down approach to finding the smallest subset of $\mathbb{U}$ generated by f and g from the elements in B. For the bottom-up approach, we define the following.

Definition 2.11 *A* **construction sequence** *is a finite sequence $\langle x_0, \ldots, x_n \rangle$ of elements of $\mathbb{U}$ such that for each $i \leq n$ we have at least one of*

$$x_i \in B,$$

$$x_i = f(x_j, x_k) \text{ for some } j < i,\ k < i,$$

$$x_i = g(x_j) \text{ for some } j < i.$$

The idea behind this definition is that each element in the sequence either came from our set of initial elements or was formed by using f or g from some previous element or elements in the sequence. The final element in the sequence is the terminal element from the construction process.

Example 2.12 *For* $a, b \in B$*,* $\langle a \rangle$*,* $\langle a, b, g(a), g(b), f(a, b) \rangle$*,* $\langle b, b, f(b, b), a, f(a, f(b, b)), g(f(a, f(b, b))) \rangle$ *are construction sequences for* a*,* $f(a, b)$*, and* $g(f(a, f(b, b))$ *respectively (note that these construction sequences are not unique).*

Now let C_* be the set of all points x such that there is a construction sequence of some length $n \in \mathbb{N}$ ending with element x. Also, let C_n be the set of points x such that there is some construction sequence of length n that ends with x.

Theorem 2.13 $B = C_1 \subseteq C_2 \subseteq C_3 \subseteq \cdots$*, and* $C_* = \bigcup_{n \in \mathbb{N}} C_n$*.*

Proof: This theorem follows from the definitions involved and a simple induction argument. ∎

Note that we can extend the above arguments for just two operations to any finite number or even countably many operations with little to no difficulty.

Example 2.14 *Assuming* $a, b \in B$*, then*

$$\langle b, b, f(b, b), a, f(a, f(b, b)), g(f(a, f(b, b))) \rangle$$

is a construction sequence of length 6 for $g(f(a, f(b, b))$*. Hence,*

$$g(f(a, f(b, b)) \in C_6 \subseteq C_*.$$

Example 2.15 *Taking* $\mathbb{U}$ *to be the set of all expressions in the sentential language,* B *to be the set of all sentence symbols in the sentential language, and* $\mathfrak{F}$ *to be the class of the formula building operations* $\mathcal{F}_{\neg}$, $\mathcal{F}_{\wedge}$, $\mathcal{F}_{\vee}$, $\mathcal{F}_{\rightarrow}$, *and* $\mathcal{F}_{\leftrightarrow}$, *then* C_* *will be the set of wffs under Definition* 2.5*.*

Theorem 2.16 $C^* = C_*$

If we prove this general result, then we will also have proven that each of our definitions for the set of wffs in the sentential language are equivalent as a specific case of this more generic result. Again, we prove this result for a single unary operation and a single binary operation, but the extension to finitely many or even countably infinitely many operations is clear.

Proof: We first show that C_* is inductive. Clearly, $B \subseteq C_*$ as was mentioned above. Let $x, y \in C_*$. Then by definition we have construction sequences $\langle x_0, \ldots, x_n, x\rangle$ and $\langle y_0, \ldots, y_k, y\rangle$. Then

$$\langle x_0, \ldots, x_n, x, y_0, \ldots, y_k, y, f(x, y)\rangle$$

$$\langle x_0, \ldots, x_n, x, g(x)\rangle$$

are all construction sequences by definition. So, $f(x, y), g(x) \in C_*$ by definition. Thus, C_* is closed under f and g and is thus inductive by definition. Hence $C^* \subseteq C_*$ since C_* is one of the sets in the intersection used to define C^*.

Now let $x \in C_*$, by definition then, there is a construction sequence $\langle x_0, \ldots, x_n = x\rangle$ Note that x_0 must be in B by how we defined construction sequences. So, $x_0 \in C^*$ since $B \subseteq C^*$. Suppose that $x_0, \ldots x_i \in C^*$ for $i < n$. Now, $x_{i+1} \in B \subseteq C^*$ or $x_{i+1} = f(x_j, x_k)$ for $j \leq k < i$ or $x_{i+1} = g(x_j)$ for $j < i$. Since C^* is inductive and thus closed under f and g and since $x_j, x_k \in C^*$ by assumption, we must have $x_{i+1} = f(x_j, x_k) \in C^*$ or $x_{i+1} = g(x_j) \in C^*$. This shows then that $x_n = x \in C^*$ by the Principle of Mathematical Induction. Since our choice of $x \in C_*$ was arbitrary, $C_* \subseteq C^*$. Therefore, $C^* = C_*$. ∎

Since $C^* = C_*$ we can unambiguously speak of the set $C = C^* = C_*$ as being *generated* from B by the operations in the class $\mathfrak{F}$.

Before leaving our discussion of induction we present a powerful principle that will aid us greatly in many of the proofs we will present.

Theorem 2.17 (The Induction Principle) *Assume that C is the set generated from B by the operations in $\mathfrak{F}$. If $B \subseteq S \subseteq C$ and S is closed under the operations in $\mathfrak{F}$, then $S = C$.*

Proof: S is inductive by assumption, so $C = C^* \subseteq S$. Since we are given that $S \subseteq C$, then $S = C$. ∎

We will use this theorem often to show that if we have a set of wffs that contains the sentence symbols such that every element of that set fulfills some property P and if that set is closed under the formula building operations, then the entire set of wffs must have property P. However, we present the following example to show how this principle is a generalized form of a very familiar principle in mathematics.

Example 2.18 *Let $\mathbb{U} = \mathbb{R}$ and $B = \{0\}$ and let the class $\mathfrak{F}$ contain only the successor function $S(x) = x + 1$. The elements in $C_* = C$ will take the form 0 or $S^n(0) = S(S(S \cdots (S(0)) \cdots))$ where in the expression on the right, S is applied n times. That is, $S^n(0) = n$ (the natural number n), and so $C = \mathbb{N}$ in this case i.e. the natural numbers are generated from the set $\{0\}$ by the successor function. The Induction Principle in this case says that if $A \subseteq \mathbb{N}$ and $0 \in A$ and A is closed under the successor function S, (i.e.*

$n \in A \Rightarrow (n+1) \in A$), then $A = \mathbb{N}$. In this case then, our Induction Principle is our old friend the Principle of Mathematical Induction.

2.4 Unique Readability of Wffs

We have rigorously developed satisfactory and equivalent definitions for what the set of wffs is in our sentential language (we will denote this set as $\mathcal{W}$). But how do we know that each wff can only be read in one way? How do we know that with our formal language we do not have ambiguity problems like we had with the English sentence "If it is not the case that you are reading this sentence and you are confused, then you should re-read it."? Ultimately, guaranteeing the unique readability for our wffs will guarantee that we obtain only one possible truth value of a wff given a truth assignment for the sentence symbols involved in the wff.

Given an n-ary function the symbol $f\big|_{S^n}$ is the function f whose domain has been restricted to S^n

Definition 2.19 *C is* **freely generated** *from B by the n-ary operation f and k-ary operation g if in addition to the requirements for being generated, in the sense discussed in the section above, we have*

1. *$f\big|_{C^n}$ and $g\big|_{C^k}$ are one-to-one, and*

2. *$Im(f\big|_{C^n})$, $Im(g\big|_{C^k})$, and B are pairwise disjoint. We can define free generation in a similar way when there are any finite number or countably infinitely many operations involved.*

The one-to-one properties of the restrictions of f and g to C provide for the uniqueness of the construction of an element in C generated from f

or g. In other words, if an element was constructed via f then there is only one way that it could have been constructed via f and thus we can reverse engineer the construction to get back to the input (i.e. $f^{-1}(a)$ is unambiguous for $a \in Im(f\big|_{C^n})$). I cannot generate the same element in C from two different inputs into f.

Property 2 guarantees that an expression in C cannot be generated by both f and g. Neither can a initial element in B be generated by either f or g. If either of these things were possible, then we would have ambiguity in reading an element in C.

Example 2.20 *Suppose $B = \{a, b, c\}$. The element $f(f(a, g(b)), c)$ will be an element generated by f and g in $\mathbb{U}$. However, if $g(b) = g(a)$ (i.e. g fails property 1 of the definition above or if $f(a, g(b)) = c$ (i.e. f fails property 2 from the definition above), then we have*

$$f(f(a, g(b)), c) = f(f(a, g(a)), c) = f(c, c).$$

Thus, the structure of $f(f(a, g(b)), c)$ cannot be uniquely described using f and g and elements in B.

We can do this example in our particular setting. Suppose $B = \{\boldsymbol{A_1}, \boldsymbol{A_2}, \boldsymbol{A_3}\}$. The element

$$\mathcal{F}_\wedge(\mathcal{F}_\wedge(\boldsymbol{A_1}, \mathcal{F}_\neg(A_2)), \boldsymbol{A_3}) = \boldsymbol{((A_1 \wedge (\neg A_2)) \wedge A_3)}$$

will be an element generated by $F_\neg$ and $\mathcal{F}_\wedge$ in $\mathbb{E}$. However, if

$$\mathcal{F}_\neg(\boldsymbol{A_2}) = \mathcal{F}_\neg(\boldsymbol{A_1})$$

(i.e. $F_\neg$ fails property 1 of the definition above or if

$$\mathcal{F}_\wedge(\boldsymbol{A_1}, F_\neg(\boldsymbol{A_2})) = \boldsymbol{A_3}$$

(i.e. $\mathcal{F}_\wedge$ fails property 2 from the definition above), then we have

$$\mathcal{F}_\wedge(\mathcal{F}_\wedge(\boldsymbol{A_1}, \mathcal{F}_\neg(\boldsymbol{A_2})), \boldsymbol{A_3}) = \mathcal{F}_\wedge(\mathcal{F}_\wedge(\boldsymbol{A_1}, \mathcal{F}_\neg(\boldsymbol{A_1})), \boldsymbol{A_3})$$

$$= \mathcal{F}_\wedge(\boldsymbol{A_3}, \boldsymbol{A_3}) = \boldsymbol{A_3} \wedge \boldsymbol{A_3}.$$

If this were the case, the structure of $\boldsymbol{((A_1 \wedge (\neg A_2)) \wedge A_3)}$ could not be uniquely described using $\mathcal{F}_\neg$ and $\mathcal{F}_\wedge$ and elements in B.

Theorem 2.21 (The Unique Readability Theorem) *The set of wffs is freely generated from the set of sentence symbols.*

To position ourselves to prove this theorem, we need a couple preliminary results.

Definition 2.22 *An expression is* **balanced** *if it has the same number of left and right parentheses. If an expression has more left parentheses than right parentheses, we will say that the expression is* **left-heavy***.*

Lemma 2.4.1 *Every wff is balanced.*

Proof: Note that the set of balanced wffs is a subset of the set of wffs that contains the set of all sentence symbols since each sentence symbol $\boldsymbol{A_i}$ is balanced having no left or right parentheses. Let $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ be balanced wffs. Then

$$\mathcal{F}_\neg(\boldsymbol{\alpha}) = \boldsymbol{(\neg\alpha)},$$

$$\mathcal{F}_\wedge(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{(\alpha \wedge \beta)},$$

$$\mathcal{F}_\vee(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{(\alpha \vee \beta)},$$

$$\mathcal{F}_\to(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{(\alpha \to \beta)}, \text{ and}$$

$$\mathcal{F}_\leftrightarrow(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{(\alpha \leftrightarrow \beta)}$$

are all balanced wffs. They are wffs by definition since they are each generated from a formula building operation, and they are each balanced since parentheses are added in balance to already balanced wffs. Thus, the set of balanced wffs is closed under the five formula building operations. Hence, by the Induction Principle, the set of balanced wffs is exactly the set of wffs. ∎

Recall that an expression in the sentential language is a finite sequence of symbols in the sentential alphabet.

Example 2.23 *Formally,*

$$((\neg \boldsymbol{A_1}) \to (\boldsymbol{A_2} \wedge (\boldsymbol{A_3} \vee \boldsymbol{A_4})))$$

$$= \langle (, \neg, \boldsymbol{A_1},), \to, (, \boldsymbol{A_2}, \wedge, (, \boldsymbol{A_3}, \vee, \boldsymbol{A_4},),),) \rangle$$

Definition 2.24 *An* **initial segment** *of the finite sequence $\langle x_1, \ldots, x_n \rangle$ is a finite sequence $\langle x_1, x_2, \ldots, x_m \rangle$, where $m \leq n$. This initial segment is* **proper** *if $m < n$.*

Recall that a wff is a finite sequence of symbols from the alphabet.

Lemma 2.4.2 *Any proper initial segment of a wff is a left-heavy expression. Thus, no proper initial segment of a wff can itself be a wff.*

Proof: Let S be the following set of wffs:

$$\{\boldsymbol{\alpha} \in \mathcal{W} : \text{every proper initial segment of } \boldsymbol{\alpha} \text{ is left-heavy}\}.$$

We will show that S is inductive. Each sentence symbol (being part of the alphabet) is indecomposable and hence has no proper initial segment. Thus, the statement "If $\boldsymbol{\alpha_0}$ is a proper initial segment of a sentence symbol, then $\boldsymbol{\alpha_0}$ is left-heavy," is vacuously true since the antecedent is always false. So, each sentence symbol is in S.

We now seek to verify that S is closed under the formula building operations. We have essentially two types of formula building operations $\mathcal{F}_\neg$ and $\mathcal{F}_\sharp$ where $\sharp \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$. Let $\boldsymbol{\alpha} \in S$. Consider the proper initial segments of $\mathcal{F}_\neg(\boldsymbol{\alpha}) = (\neg\boldsymbol{\alpha})$ listed below:

1. $($
2. $(\neg$
3. $(\neg\boldsymbol{\alpha_0}$ where $\boldsymbol{\alpha_0}$ is any proper initial segment of $\boldsymbol{\alpha}$
4. $(\neg\boldsymbol{\alpha}$

In cases 1 and 2, we clearly have more left parentheses than right parentheses. In case 3, by assumption $\boldsymbol{\alpha} \in S$ and any proper initial segment $\boldsymbol{\alpha_0}$ of $\boldsymbol{\alpha}$ has more left parentheses than right parentheses. Tacking on $(\neg$ onto the front just adds to the unbalance and thus $(\neg\boldsymbol{\alpha_0}$ is left-heavy for any proper initial segment $\boldsymbol{\alpha_0}$ of $\boldsymbol{\alpha}$. In case 4, $\boldsymbol{\alpha}$ is balanced, so tacking on $(\neg$ gives the expression one more left parentheses than right parentheses, and the expression is thus left heavy. S is thus closed under $\mathcal{F}_\neg$. We now consider proper initial segments of $\mathcal{F}_\sharp(\boldsymbol{\alpha}, \boldsymbol{\beta}) = (\boldsymbol{\alpha}\sharp\boldsymbol{\beta})$.

1. $($
2. $(\boldsymbol{\alpha_0}$ where $\boldsymbol{\alpha_0}$ is a proper initial segment of $\boldsymbol{\alpha}$
3. $(\boldsymbol{\alpha}$
4. $(\boldsymbol{\alpha}\sharp$
5. $(\boldsymbol{\alpha}\sharp\boldsymbol{\beta_0}$ where $\boldsymbol{\beta_0}$ is a proper initial segment of $\boldsymbol{\beta}$
6. $(\boldsymbol{\alpha}\sharp\boldsymbol{\beta}$

The first expression is clearly left-heavy. By assumption $\boldsymbol{\alpha}, \boldsymbol{\beta} \in S$ so initial segments $\boldsymbol{\alpha_0}$ and $\boldsymbol{\beta_0}$ will be left-heavy. Tacking on a left parentheses to $\boldsymbol{\alpha_0}$ will only add to the unbalance and so $(\boldsymbol{\alpha_0}$ will be left-heavy. For case three, $\boldsymbol{\alpha}$ will be balanced so $(\boldsymbol{\alpha}$ has one more left parentheses than the number of right parentheses. A similar statement holds for $(\boldsymbol{\alpha}\sharp$ since $\boldsymbol{\alpha}\sharp$ has the same number of left and right parentheses. Also, since $\boldsymbol{\alpha\sharp}$ is a balanced expression, $\boldsymbol{\alpha\sharp\beta_0}$ will be left-heavy since $\boldsymbol{\beta_0}$ is an initial segment of $\boldsymbol{\beta}$ and is thus left-heavy by assumption. Hence, $\boldsymbol{(\alpha\sharp\beta_0}$ will also be left-heavy since we have added another left parentheses to an already left-heavy expression. Finally, each of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are balanced being wffs. Thus $\boldsymbol{\alpha\sharp\beta}$ will also be balanced. Clearly then, $\boldsymbol{(\alpha\sharp\beta}$ will be left-heavy. So, S is closed under $\mathcal{F}_\sharp$ where $\sharp \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, and hence S is closed under all of the formula building operations. Since S contains B, S is inductive, thus by the Induction Principle, S is exactly the set of all wffs. This means that any initial segment of any wff is left-heavy and thus cannot itself be a wff by the previous lemma. ∎

We are now in a position to prove the Unique Readability Theorem for the set of wffs. Again, this theorem states that the set of all wffs is freely generated from the set of sentence symbols. To show this result, we must show that the the formula-building operations restricted to the set of wffs within the set of expressions are each one-to-one, and we must also show that $\mathcal{F}_\sharp(\mathcal{W}^n) \cap \mathcal{F}_\flat(\mathcal{W}^k) = \emptyset$ (for $n, k \in \{1, 2\}$) when $\sharp \neq \flat$ for

$$\sharp, \flat \in \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$$

and $\mathcal{F}_\sharp(\mathcal{W}^n) \cap \mathcal{S} = \emptyset$ where $\sharp \in \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ and where $\mathcal{S}$ is the set of sentence symbols. To show how each of these could fail when the formula-building operations are *not* restricted to $\mathcal{W}^n$, we present the following examples.

Example 2.25 *Let* $\boldsymbol{\alpha} = \boldsymbol{A_1}$, $\boldsymbol{\beta} = \boldsymbol{A_2} \vee \boldsymbol{A_3} \wedge \boldsymbol{A_4}$, $\boldsymbol{\gamma} = \boldsymbol{A_1} \wedge \boldsymbol{A_2} \vee \boldsymbol{A_3}$, *and* $\boldsymbol{\delta} = \boldsymbol{A_4}$. *Then,*

$$\begin{aligned} \mathcal{F}_\wedge(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= (\boldsymbol{\alpha} \wedge \boldsymbol{\beta}) \\ &= (\boldsymbol{A_1} \wedge \boldsymbol{A_2} \vee \boldsymbol{A_3} \wedge \boldsymbol{A_4}) \\ &= (\gamma \wedge \boldsymbol{\delta}) \\ &= \mathcal{F}_\wedge(\boldsymbol{\gamma}, \boldsymbol{\delta}) \end{aligned}$$

Hence, $\mathcal{F}_\wedge$ is not one-to-one since $\boldsymbol{\alpha} \neq \boldsymbol{\beta}$ and $\boldsymbol{\gamma} \neq \boldsymbol{\delta}$. Notice that $\boldsymbol{\alpha}$ is an initial segment of $\boldsymbol{\gamma}$.

Example 2.26 *Let $\boldsymbol{\alpha} = \boldsymbol{A_1}$, $\boldsymbol{\beta} = \boldsymbol{A_2} \vee \boldsymbol{A_3}$, $\boldsymbol{\gamma} = \boldsymbol{A_1} \rightarrow \boldsymbol{A_2}$, and $\boldsymbol{\delta} = \boldsymbol{A_3}$. Then, $\mathcal{F}_\rightarrow(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{(A_1 \rightarrow A_2 \vee A_3)} = \mathcal{F}_\vee(\boldsymbol{\gamma}, \boldsymbol{\delta})$. Thus, $Im(\mathcal{F}_\rightarrow) \cap Im(\mathcal{F}_\vee) \neq \emptyset$. Notice that again $\boldsymbol{\alpha}$ is an initial segment of $\boldsymbol{\gamma}$.*

These examples illustrate the huge role that parentheses play in wffs to make them work the way we want them to. We make two quick notes before proving the theorem: (1) No sentence symbol begins with $\boldsymbol{(}$, and (2) no wff starts with the logical symbol $\boldsymbol{\neg}$.

Proof: (of the Unique Readability Theorem) We first show that the images of the operations $\mathcal{F}_\sharp\big|_{\mathcal{W}^2}$ for $\sharp \in \{\boldsymbol{\wedge}, \boldsymbol{\vee}, \boldsymbol{\rightarrow}, \boldsymbol{\leftrightarrow}\}$ are pairwise disjoint. Assume

$$\mathcal{F}_\sharp\big|_{\mathcal{W}^2}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{(\alpha \sharp \beta)} = \boldsymbol{(\gamma \flat \delta)} = \mathcal{F}_\flat\big|_{\mathcal{W}^2}(\boldsymbol{\gamma}, \boldsymbol{\delta})$$

for $\sharp, \flat \in \{\boldsymbol{\wedge}, \boldsymbol{\vee}, \boldsymbol{\rightarrow}, \boldsymbol{\leftrightarrow}\}$. By definition, we may write $\boldsymbol{(\alpha \sharp \beta)}$ as a finite sequence

$$\langle \boldsymbol{(}, \boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_n}, \sharp, \boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_k}, \boldsymbol{)} \rangle$$

where $\boldsymbol{\alpha} = \langle \boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_n} \rangle$ and $\boldsymbol{\beta} = \langle \boldsymbol{b_1}, \boldsymbol{b_2}, \ldots \boldsymbol{b_k} \rangle$ and the $\boldsymbol{a_i}$'s and $\boldsymbol{b_i}$'s are indecomposable symbols from our alphabet. Similarly

$$\boldsymbol{(\gamma \flat \delta)} = \langle \boldsymbol{(}, \boldsymbol{g_1}, \boldsymbol{g_2}, \ldots, \boldsymbol{g_m}, \flat, \boldsymbol{d_1}, \boldsymbol{d_2}, \ldots \boldsymbol{d_r}, \boldsymbol{)} \rangle.$$

Without loss of generality $n \geq m$. Suppose $n > m$. Then for all $1 \leq i \leq m$, $\boldsymbol{a_i} = \boldsymbol{g_i}$, and $\boldsymbol{a_{m+1}} = \flat$. So, $\langle \boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_m} \rangle = \boldsymbol{\gamma}$, a wff. However, since $\langle \boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_m} \rangle$ is a proper initial segment of $\boldsymbol{\alpha}$, a wff, it must be left-heavy by the preceding lemma, and cannot be a wff, a contradiction. Hence, $m = n$ ($\boldsymbol{\alpha} = \boldsymbol{\gamma}$), implying that $\sharp = \flat$. Since the lengths of the wffs we assumed to be equal must be the same we must have $\boldsymbol{\beta} = \boldsymbol{\delta}$. We have not only shown that the images of $\mathcal{F}_\sharp$ and $\mathcal{F}_\flat$ must be pairwise disjoint, but also (if we replace $\flat$ with $\sharp$ in our initial assumption of equality), that $\mathcal{F}_\sharp$ is one-to-one for each $\sharp \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$. It is clear that $\mathcal{F}_\neg$ will also be one-to-one.

Suppose that $\mathcal{F}_\sharp\big|_{\mathcal{W}^2}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathcal{F}_\neg\big|_{\mathcal{W}}(\boldsymbol{\gamma})$ where $\sharp$ is as above. Then we have $\mathbf{(}\boldsymbol{\alpha}\sharp\boldsymbol{\beta}\mathbf{)} = \mathbf{(}\neg\boldsymbol{\gamma}\mathbf{)}$. Thus, $\boldsymbol{\alpha}\sharp\boldsymbol{\beta}\mathbf{)} = \neg\boldsymbol{\gamma}\mathbf{)}$. This cannot happen by the second of the two notes we made before the start of the proof. In fact then, $\mathcal{F}_\sharp\big|_{\mathcal{W}^2}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \neq \mathcal{F}_\neg\big|_{\mathcal{W}}(\boldsymbol{\gamma})$ for any $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{W} \times \mathcal{W}$ and $\boldsymbol{\gamma} \in \mathcal{W}$.

By the first note before the start of the proof, the images of the formula-building operations together with the set of sentence symbols are all pairwise disjoint since a formula-building operation always adds a left parenthesis. Hence we have shown that $\mathcal{W}$ is freely generated from the set of sentence symbols by the five formula-building operations. ∎

Essentially the wffs being freely generated from the set of sentence symbols by the five formula building operations says that given a wff, it can be uniquely deconstructed back into its constituent wffs i.e. there is one and only one way to build that particular wff using the formula-building operations. This result will be crucial in developing the notion of truth assignments, which we now do.

2.5 Truth Assignments

Now that we have developed our alphabet, defined what grammatically correct statements in the sentential language are, and shown that there is no ambiguity in reading these statements, we want to talk about how one wff will follow logically from another wff. Remember that with sentential logic, we're modeling deductive thought. If we think of a wff as providing the structure into which we can translate an English statement or proposition, we want to be able to model what a deduction in English would look like in our language. A deduction in English happens when given a statement, if we know the statement is true, it must also be true that another statement is true. We deduce the second statement from the first. So, in our formal language, we may think of one wff being deducible from another if when the first wff is true, the second wff *must* also be true. For example, if the wff $\boldsymbol{(A_1 \wedge A_2)}$ is a translation from English into our formal sentential language of a true proposition, we should be able to deduce $\boldsymbol{A_1}$ from our initial wff. Notice that *what* proposition we are translating into the wff $\boldsymbol{(A_1 \wedge A_2)}$ is inconsequential but merely the *truth value* of that proposition.

In a mathematics foundations course, we are taught to assign truth values (usually symbolized with T or F) to each of the sentence symbols involved in the wffs at hand and then calculate the truth values of the constituent wffs that make up the wff or wffs under consideration. This process can be achieved by filling out a row in a truth table, and we may fill out such a row for each possible combination of truth values for the sentence symbols involved. We then say that wff $\boldsymbol{\alpha}$ implies wff $\boldsymbol{\beta}$ if whenever there is a row in the truth table where $\boldsymbol{\alpha}$ is true, in the same row the wff $\boldsymbol{\beta}$ is also true. Since we are attempting to mathematically model deductive thought, we must rigorously develop these notions. First, we need to mathematically develop how to assign truth values to wffs, formalizing our more intuitive foundations notions. To do this, we develop more general notions of recursive functions.

2.5.1 Recursion

Our idea of assigning truth values to our wffs is a recursive idea. To figure out the truth value of the current wff under consideration, we need to look at the truth value of the wff last in line. In a more general setting we have a set $\mathbb{U}$ (like the set of all expressions), a subset B of $\mathbb{U}$ (like the set of all sentence symbols) and two functions $f : \mathbb{U} \times \mathbb{U} \longrightarrow \mathbb{U}$ and $g : \mathbb{U} \longrightarrow \mathbb{U}$ where $C \subseteq \mathbb{U}$ is the set generated from B by f and g (C would correspond to the set of all wffs). Our goal is to define a function on C recursively, that is we want a function $\overline{h}$ with domain C such that

1. We have rules for computing $\overline{h}(x)$ for $x \in B$

2a. We have rules for computing $\overline{h}(f(x, y))$ based on $\overline{h}(x)$ and $\overline{h}(y)$

2b. We have rules for computing $\overline{h}(g(x))$ based on $\overline{h}(x)$.

These will be our only three cases we need to explore since C is generated from B and thus elements in C take the form, $x \in B$, $f(x, y)$ for $x, y \in C$, or $g(x)$ for $x \in C$. The idea is that given some $c \in C$ and if we desire to compute $\overline{h}(c)$, then we merely need to see how c is generated by f and g and reverse engineer that construction until we eventually reach the "bottom" where we see what $\overline{h}$ does to the elements of B from which c is generated.

Example 2.27 *Let* $\mathbb{U} = \mathbb{R}$, $B = \{0\}$, $S(x) = x + 1$, *and* $C = \mathbb{N}$ *as discussed above. Now define*

$$\overline{h}(0) = 0$$

$$\overline{h}(S(x)) = S(x) + \overline{h}(x)$$

So,

$$\begin{aligned}
\overline{h}(4) &= \overline{h}(S(3)) \\
&= S(3) + \overline{h}(3) \\
&= 4 + \overline{h}(S(2)) \\
&= 4 + (S(2) + \overline{h}(2)) \\
&= 4 + 3 + \overline{h}(S(1)) \\
&= 4 + 3 + S(1) + \overline{h}(1) \\
&= 4 + 3 + 2 + S(0) + \overline{h}(0) \\
&= 4 + 3 + 2 + 1 + 0
\end{aligned}$$

So, $\overline{h}(x) = \dfrac{x(x+1)}{2}$

The question arises as to whether such a function always exists that fulfils the rules given? It does not.

Example 2.28 *Let* $U = \mathbb{R}$, $B = \{0\}$, $f(x, y) = x \cdot y$, *and* $g(x) = x + 1$. *Then* $C = \mathbb{N}$. *We attempt to define* $\overline{h}$ *as a function as follows:*

1. $\overline{h}(0) = 0$

2a. $\overline{h}(f(x, y)) = f(\overline{h}(x), \overline{h}(y))$

2b. $\overline{h}(g(x)) = \overline{h}(x) + 2$

On one hand

$$\begin{aligned}
\overline{h}(1) &= \overline{h}(f(g(0), g(0))) \\
&= f(\overline{h}(g(0)), \overline{h}(g(0))) \\
&= f(\overline{h}(0) + 2, \overline{h}(0) + 2) \\
&= f(2, 2) = 4
\end{aligned}$$

On the other hand $\overline{h}(1) = \overline{h}(g(0)) = \overline{h}(0) + 2 = 2$. *Thus,* $\overline{h}$ *cannot be a function. The main problem here is that 1 is not uniquely generated from* f *and* g *since* $Im(g) \cap Im(f) \neq \emptyset$.

Since it will not always be the case that such a recursive function exists, we seek to establish conditions for when such a recursive function will exist. These conditions are given by the following Theorem.

Theorem 2.29 (The Recursion Theorem) *Assume that the subset C of $\mathbb{U}$ is freely generated from B by f and g, where*

$$f : \mathbb{U} \times \mathbb{U} \longrightarrow \mathbb{U} \qquad g : \mathbb{U} \longrightarrow \mathbb{U}.$$

Further assume that V is a set and that F, G, and h are functions such that

$$h : B \longrightarrow V \qquad F : V \times V \longrightarrow V \qquad G : V \longrightarrow V.$$

Then there is a unique function

$$\overline{h} : C \longrightarrow V$$

such that

(i) For $x \in B$, $\overline{h}(x) = h(x)$.

(ii) For $x, y \in C$,

$$\overline{h}(f(x, y)) = F(\overline{h}(x), \overline{h}(y)) \qquad \overline{h}(g(x)) = G(\overline{h}(x)).$$

Viewed algebraically, the conclusion of this theorem says that any map h of B into V can be extended to a homomorphism $\overline{h}$ from C (with operations f and g) into V (with operations F and G). The idea behind the theorem is that we can assign values from a new set V to our basis elements in B and then compute the value of my input into $\overline{h}$ using operations on the new set V following the same structure for how my input into $\overline{h}$ was uniquely constructed in C. For us, the h function will be a truth assignment for the sentence symbols, and then $\overline{h}$ will give us the ability to calculate the truth value of any sentential wff given this truth assignment.

Proof: We will call a function v *acceptable* if the following statements hold for v:

1. $Dom(v) \subseteq C$ and $Im(v) \subseteq V$.

2. $x \in B \cap Dom(v)$ implies that $v(x) = h(x)$

3. $f(x,y), g(x) \in Dom(v)$ implies that $x, y \in Dom(v)$ and

$$v(f(x,y)) = F(v(x), v(y)) \text{ and } v(g(x)) = G(v(x))$$

Let K be the set of all acceptable functions. We must first show that K is non-empty. Consider $h : B \longrightarrow V$. Property (1) is certainly fulfilled as is (2). Since C is freely generated from B, B, $Im(f\big|_{C^2})$, and $Im(g\big|_{C})$ are pairwise disjoint. Thus, no element of B can take the form $f(x,y)$ or $g(x)$ where $x, y \in C$. Hence, (3) is vacuously true. Hence, h is an acceptable function, and hence, $K \neq \emptyset$.

Now let $\overline{h} = \bigcup_{v \in K} v$. So $(x,y) \in \overline{h}$ if and only if $v(x) = y$ for some $v \in K$. We seek to verify that $\overline{h}$ is acceptable. First we must show that it is indeed a function. Certainly we may say that it is a relation, that is $\overline{h} \subseteq C \times V$. Let $S = \{x \in C : \text{For at most one } y,\ (x,y) \in \overline{h}\}$. We seek to show that S is inductive. Let $x \in B$. Suppose $(x,y) \in \overline{h}$. Then there is $v \in K$ such that $v(x) = y$. Since $x \in B$ and v is acceptable, $y = h(x)$. Thus, there is at most one y such that $(x,y) \in \overline{h}$ for $x \in B$. Thus, $B \subseteq S$. Now let x_1, x_2 in S Suppose $(f(x_1,x_2), y) \in \overline{h}$. Then there exists $v \in K$ such that
$v(f(x_1,x_2)) = y$. Since v is acceptable it fulfils (2) above. Thus we have

$$y = v(f(x_1,x_2)) = F(v(x_1), v(x_2)).$$

So, $x_1, x_2 \in Dom(v)$ i.e. there exist $y_1, y_2 \in V$ such that $v(x_1) = y_1$ and $v(x_2) = y_2$. Thus $(x_1, y_1), (x_2, y_2) \in \overline{h}$ and since $x_1, x_2 \in S$, y_1 and y_2 are the only values paired with x_1 and x_2 respectively. Now if $v' \in K$ such that $v'(f(x_1, x_2)) = y'$ (in which case $(f(x_1, x_2), y') \in \overline{h}$), we have

$$F(v(x_1), v(x_2)) = y'$$

which implies that $F(y_1, y_2) = y = y'$ since F is assumed to be a function. Thus, for at most one y we have $(f(x_1, x_2), y) \in \overline{h}$. Hence, $f(x_1, x_2) \in S$. Similarly for $g(x_1)$. So, S is closed under f and g, and by the Induction Principle, we must have $S = C$. Since $Dom(\overline{h}) \subseteq C$, we may say that for every $x \in Dom(\overline{h})$ there exists a unique y such that $(x, y) \in \overline{h}$. Thus, $\overline{h}$ is by definition, a function.

We next, seek to show that $\overline{h}$ is acceptable. Property (1) certainly holds. For property (2), let $x \in Dom(\overline{h}) \cap B$. Then $(x, y) \in \overline{h}$ for some unique $y \in V$. By the definition of $\overline{h}$, there exists $v \in K$ such that $(x, y) \in v$. Since $x \in Dom(v) \cap B$ and v is acceptable, $y = v(x) = h(x)$. Thus, $\overline{h}(x) = h(x)$ for $x \in Dom(\overline{h}) \cap B$, and property (2) holds.

For property (3), let $f(x_1, x_2) \in Dom(\overline{h})$ for $x_1, x_2 \in C$. Thus, there is a unique $y \in V$ such that $(f(x_1, x_2), y) \in \overline{h}$, and by definition of $\overline{h}$, there is $v \in K$ such that $v(f(x_1, x_2)) = y$. Since v is acceptable, $x_1, x_2 \in Dom(v) \subseteq Dom(\overline{h})$, and

$$v(f(x_1, x_2)) = F(v(x_1), v(x_2)) = y.$$

So $(x_1, v(x_1)), (x_2, v(x_2)) \in v \subseteq \overline{h}$. Hence $\overline{h}(x_1) = v(x_1)$ and $\overline{h}(x_2) = v(x_2)$. Thus $\overline{h}(f(x_1, x_2)) = y = v(f(x_1, x_2)) = F(v(x_1), v(x_2)) = F(\overline{h}(x_1), \overline{h}(x_2))$. Similarly, if $g(x_1) \in Dom(\overline{h})$ for $x_1 \in C$, then $x_1 \in Dom(\overline{h})$ and $\overline{h}(g(x_1)) = G(\overline{h}(x_1))$. So, property (3) holds, and $\overline{h}$ is acceptable.

Now we show that $Dom(\overline{h})$ is C. To do this, we show that $Dom(\overline{h})$ is inductive. As shown earlier, h is acceptable with $B = Dom(h) \subseteq Dom(\overline{h})$. Now let $x_1, x_2 \in Dom(\overline{h})$. Our goal is to demonstrate that both $f(x_1, x_2)$ and $g(x_1)$ are in $Dom(\overline{h})$. We do this by creating a new function that has $f(x_1, x_2)$ in its domain and then show that this new function is a subset of the old function.

Let $\overline{h}' = \overline{h} \cup \{(f(x_1, x_2), F(\overline{h}(x_1), \overline{h}(x_2))\}$. We show that $\overline{h}'$ is an acceptable function. It will be a function since $\overline{h}$ is a function and by Property (3) applied to $\overline{h}$. Property (1) of acceptable functions clearly holds for $\overline{h}'$ as does property (2) since $Im(f\big|_{C^2}) \cap B = \emptyset$, so that $f(x_1, x_2) \notin B$. Hence, $\overline{h}'(x) = \overline{h}(x) = h(x)$ for $x \in B$. It is also clear that property (3) of acceptable functions holds for $\overline{h}'$ since the property holds for all $(x, y) \in \overline{h} \subseteq \overline{h}'$ and

$$\overline{h}'(f(x_1, x_2)) = F(\overline{h}'(x_1), \overline{h}'(x_2)).$$

Hence, $\overline{h}'$ is acceptable, and is in K. But then, $\overline{h}' \subseteq \overline{h}$, and we have $f(x_1, x_2) \in Dom(\overline{h})$. Similarly, $g(x_1) \in Dom(\overline{h})$. Thus, by the Induction Principle, $Dom(\overline{h}) = C$.

At this point we have shown that we have a function $\overline{h} : C \longrightarrow V$ that is an acceptable function. We now must show that this function is unique. Suppose we have $\overline{h}' : C \longrightarrow V$ that is also acceptable. Let S be the set on which $\overline{h}$ and $\overline{h}'$ agree. $B \subseteq S$ since for all $x \in B$, $\overline{h} = h(x) = \overline{h}'$. Let $x_1, x_2 \in S$, then $f(x_1, x_2) \in C = Dom(\overline{h}) = Dom(\overline{h}')$, and

$$\overline{h}(f(x_1, x_2)) = F(\overline{h}(x_1), \overline{h}(x_2)) = F(\overline{h}'(x_1), \overline{h}'(x_2)) = \overline{h}'(f(x_1, x_2))$$

since $\overline{h}(x_i) = \overline{h}'(x_i)$ for $i = 1, 2$ by assumption. Hence $f(x_1, x_2) \in S$. Similarly, $g(x_1) \in S$. Thus, by the Induction Principle $S = C$, and hence $\overline{h} = \overline{h}'$. Thus, we have proved the Recursion Theorem. ∎

We have proved the Recursion Theorem for free generation by two functions f and g, but of course the results may be extended to free generation by any finite number of functions. To get a sense of what the Recursion Theorem gives us, we consider the following examples.

Example 2.30 *Recall that* $\mathbb{N}$ *is generated from* $\{0\}$ *by the successor function* S. *Since* S *is one-to-one and* $Im(S) \cap \{0\} = \emptyset$, *then we may say that* $\mathbb{N}$ *is freely generated from* $\{0\}$ *by* S. *So, by the Recursion Theorem, for any set* V, *any* $a \in V$, *and any* $F : V \longrightarrow V$ *with* $h(0) = a$, *there will exist a unique* $\overline{h}$ *such that* $\overline{h}(0) = a$ *and for* $n \in \mathbb{N}$, $\overline{h}(S(n)) = F(\overline{h}(n))$

As a specific case, take V *to be the set of prime numbers, and let*

$$h(0) = p_0 = 2.$$

Let $F(p_i) = p_{i+1}$ *(*p_i *is the* i*th prime number). Then,*

$$\overline{h}(4) = \overline{h}(S(3)) = F(\overline{h}(3)) = F(F(F(F(h(0))))) = F(F(F(F(2)))) = p_5 = 11.$$

Example 2.31 *The wffs are freely generated from the set of sentence symbols by the five formula building operations. Let* $\mathcal{S}$ *denote the set of sentence symbols,* $\mathcal{W}$ *the set of wffs, and* $V = \mathbb{Z}^+$. *Let* $h : \mathcal{S} \longrightarrow \mathbb{Z}^+$ *defined by* $h(\boldsymbol{A}) = 1$ *for* $\boldsymbol{A} \in \mathcal{S}$. *Let* $G_\neg : \mathbb{Z}^+ \longrightarrow \mathbb{Z}^+$ $G_\sharp : \mathbb{Z}^+ \times \mathbb{Z}^+ \longrightarrow \mathbb{Z}^+$ *be defined by* $G_\neg(n) = 3 + n$ *and* $G_\sharp(n, m) = 3 + n + m$ *for* $\sharp \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$. *The Recursion Theorem then states that there exists a unique* $\overline{h} : \mathcal{W} \longrightarrow \mathbb{Z}^+$ *such that* $\overline{h}(\boldsymbol{A}) = 1$ *for each* $\boldsymbol{A} \in \mathcal{S}$ *and* $\overline{h}((\neg\boldsymbol{\alpha})) = \overline{h}(\mathcal{F}_\neg(\boldsymbol{\alpha})) = G_\neg(\overline{h}(\boldsymbol{\alpha})) = 3 + \overline{h}(\boldsymbol{\alpha})$ *and* $\overline{h}((\boldsymbol{\alpha}\sharp\boldsymbol{\beta})) = \overline{h}(\mathcal{F}_\sharp(\boldsymbol{\alpha}, \boldsymbol{\beta})) = G_\sharp(\overline{h}(\boldsymbol{\alpha}), \overline{h}(\boldsymbol{\beta})) = 3 + \overline{h}(\boldsymbol{\alpha}) + \overline{h}(\boldsymbol{\beta})$. $\overline{h}$ *simply gives the length of each wff, finding the length by "peeling" off and counting symbols until it reaches the sentence symbols.*

Having now established some general recursion results, we now return to our primary discussion of formalizing the notion of truth assignments for wffs in the sentential language.

We will consider the set $\{T, F\}$ where T is thought of as *truth* and F is thought of as *falsity*. We could just as easily think of this set as $\{1, 0\}$, the way a computer "thinks" of truth and falsity. Let $\mathcal{S}$ be the set of sentence symbols, and suppose we are given $v : \mathcal{S} \longrightarrow \{T, F\}$. Such a function will be what we mean when we use the term *truth assignment*. We of course have the set of wffs $\mathcal{W}$ being freely generated by the five formula building operations: $\mathcal{F}_\neg, \mathcal{F}_\sharp$ where $\sharp \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$. Let $\mathcal{G}_\neg : \{T, F\} \longrightarrow \{T, F\}$ and $\mathcal{G}_\sharp : \{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$ for $\sharp \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$ where

$$\mathcal{G}_\neg(V) = \begin{cases} T & \text{if } V = F \\ F & \text{if } V = T \end{cases}$$

$$\mathcal{G}_\vee(V_1, V_2) = \begin{cases} F & \text{if } V_1 = F \text{ and } V_2 = F \\ T & \text{otherwise} \end{cases}$$

$$\mathcal{G}_\wedge(V_1, V_2) = \begin{cases} T & \text{if } V_1 = T \text{ and } V_2 = T \\ F & \text{otherwise} \end{cases}$$

$$\mathcal{G}_\rightarrow(V_1, V_2) = \begin{cases} F & \text{if } V_1 = T \text{ and } V_2 = F \\ T & \text{otherwise} \end{cases}$$

$$\mathcal{G}_\leftrightarrow(V_1, V_2) = \begin{cases} T & \text{if } V_1 = V_2 \\ F & \text{otherwise} \end{cases}$$

These are the natural truth value operations for "not", "and", etc. Applying the Recursion Theorem, we are guaranteed the existence of unique $\overline{v} : \mathcal{W} \longrightarrow \{T, F\}$ where $\overline{v}(\boldsymbol{A}) = v(\boldsymbol{A})$ for $\boldsymbol{A} \in \mathcal{S}$, and where

$\overline{v}(\mathcal{F}_\neg(\boldsymbol{\alpha})) = \mathcal{G}_\neg(\overline{v}(\boldsymbol{\alpha}))$ and $\overline{v}(\mathcal{F}_\sharp(\boldsymbol{\alpha}, \boldsymbol{\beta})) = \mathcal{G}_\sharp(\overline{v}(\boldsymbol{\alpha}), \overline{v}(\boldsymbol{\beta}))$ for $\sharp \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$. Equivalently, $\overline{v}((\neg\boldsymbol{\alpha})) = \mathcal{G}_\neg(\overline{v}(\boldsymbol{\alpha}))$ and $\overline{v}((\boldsymbol{\alpha}\sharp\boldsymbol{\beta})) = \mathcal{G}_\sharp(\overline{v}(\boldsymbol{\alpha}), \overline{v}(\boldsymbol{\beta}))$.

The moral of the story is that given a wff, and a given assignment of truth and falsity to each of the sentence symbols involved in the wff, we can compute a unique truth value of the wff by looking at the truth value of its pieces. The existence of a unique truth value for any wff given truth assignments for each of its sentence symbols comes as a consequence of the wffs being freely generated from the sentence symbols i.e. the Unique Readability Theorem. We can only read each wff in one way and so when we assign truth values to each of its sentence symbols, we can only obtain one truth value for the wff.

Example 2.32 *Consider the wff* $(\boldsymbol{A_1} \wedge \boldsymbol{A_2})$ *and let* $v : \mathcal{S} \longrightarrow \{T, F\}$ *where* $v(\boldsymbol{A_1}) = T$, $v(\boldsymbol{A_2}) = F$, *and* $v(\boldsymbol{A_i}) = T$ *for* $i > 2$. *By the Recursion Theorem, we have* $\overline{v} : \mathcal{W} \longrightarrow \{T, F\}$, *and*

$$\overline{v}((\boldsymbol{A_1} \wedge \boldsymbol{A_2})) = \mathcal{G}_\wedge(\overline{v}(\boldsymbol{A_1}), \overline{v}(\boldsymbol{A_2})) = \mathcal{G}_\wedge(v(\boldsymbol{A_1}), v(\boldsymbol{A_2})) = \mathcal{G}_\wedge(T, F) = F$$

Example 2.33 *Consider the wff*

$$(((\neg(\boldsymbol{A_1} \wedge \boldsymbol{A_2})) \rightarrow ((\neg\boldsymbol{A_1}) \vee (\neg\boldsymbol{A_2}))) \leftrightarrow (((\neg\boldsymbol{A_1}) \vee (\neg\boldsymbol{A_2})) \rightarrow (\neg(\boldsymbol{A_1} \wedge \boldsymbol{A_2}))))$$

We use the same v *as in the previous example. To speed computation, instead of writing, say* $\mathcal{G}_\neg(T) = F$ *or* $\mathcal{G}_\wedge(T, F) = F$ *we will write* $(\neg T) = F$ *and*

$(T \wedge F) = F$. *So with the wff at hand*

$$\begin{aligned}
& \overline{v}((((\neg(\boldsymbol{A_1} \wedge \boldsymbol{A_2})) \rightarrow ((\neg \boldsymbol{A_1}) \vee (\neg \boldsymbol{A_2}))) \leftrightarrow (((\neg \boldsymbol{A_1}) \vee (\neg \boldsymbol{A_2})) \rightarrow \\
& (\neg(\boldsymbol{A_1} \wedge \boldsymbol{A_2})))))) \\
= \; & (((\neg(T \wedge F)) \rightarrow ((\neg T) \vee (\neg F))) \leftrightarrow (((\neg T) \vee (\neg F)) \rightarrow \\
& (\neg(T \wedge F)))) \\
= \; & (((\neg F) \rightarrow (F \vee T)) \leftrightarrow ((F \vee T) \rightarrow (\neg F))) \\
= \; & ((T \rightarrow T) \leftrightarrow (T \rightarrow T)) \\
= \; & (T \leftrightarrow T) \\
= \; & T
\end{aligned}$$

Given these examples we see at once that each row of a truth table represents a truth assignment from the sentence symbols to the set $\{T, F\}$.

Example 2.34 *The following table lists all possible truth assignments for the wff in the bottom column.*

$\boldsymbol{A_1}$	$\boldsymbol{A_2}$	$(\neg \boldsymbol{A_1})$	$(\neg \boldsymbol{A_2})$	$(\boldsymbol{A_1} \wedge \boldsymbol{A_2})$	$(\neg(\boldsymbol{A_1} \wedge \boldsymbol{A_2}))$	$((\neg \boldsymbol{A_1}) \vee (\neg \boldsymbol{A_2}))$
T	F	F	T	F	T	T
T	T	F	F	T	F	F
F	F	T	T	F	T	T
F	T	T	F	F	T	T

$((\neg(\boldsymbol{A_1} \wedge \boldsymbol{A_2})) \leftrightarrow ((\neg \boldsymbol{A_1}) \vee (\neg \boldsymbol{A_2}))$
T
T
T
T

Each row corresponds to a truth assignment from $\mathcal{S}$ to $\{T, F\}$ with the values in the first two columns giving the assignment on the sentence symbols of interest. As we move from left to right in this particular truth table, we are computing the truth value of larger and larger constituent pieces of the wff in *the top row of the bottom column. Row 1 corresponds to v in the previous two examples, and our computations from columns 3 to 6 are via $\overline{v}$ given to us by the Recursion Theorem.*

Of course, the truth table above indicates that no matter what truth assignment we make for the sentence symbols in the wff

$$((\neg(\boldsymbol{A_1} \wedge \boldsymbol{A_2})) \leftrightarrow ((\neg \boldsymbol{A_1}) \vee (\neg \boldsymbol{A_2})),$$

the wff will always be true. Analyzing the truth value of wffs given a truth assignments for the sentence symbols involved is at the heart of our design of modeling human deductive thought processes. We have characterized a deduction as occurring when the truth of one statement guarantees the truth of another statement. The deduction comes from the inherent structure of the statement. Thus, in the case above, no matter what natural language (English) statements the sentence symbols $\boldsymbol{A_1}$ and $\boldsymbol{A_2}$ are intended to translate, the wff $((\neg(\boldsymbol{A_1} \wedge \boldsymbol{A_2})) \leftrightarrow ((\neg \boldsymbol{A_1}) \vee (\neg \boldsymbol{A_2}))$ will always be a truism. Truth assignments give us the tool for modeling and analyzing the raw logical structure in statements and hence the deducibility of one statement from another. We now formalize the notion of deducibility within our sentential language.

2.6 Tautological Implication

We begin with some definitions which will aid our discussion.

Definition 2.35 *We say that $v : \mathcal{S} \longrightarrow \{T, F\}$* **satisfies** *wff $\boldsymbol{\varphi}$ if $\overline{v}(\boldsymbol{\varphi}) = T$.*

Definition 2.36 *The set of wffs Σ* **tautologically implies** *the wff $\boldsymbol{\tau}$ and we write $\Sigma \models \boldsymbol{\tau}$ if every truth assignment on the set of sentence symbols that satisfies every wff in Σ also satisfies the wff $\boldsymbol{\tau}$.*

Notice, as we have implicitly done in the truth table in Example 2.37, we need only consider what a particular truth assignment will do to the sentence symbols in the wffs in question. There will of course be infinitely many truth assignments on the set of *all* sentence symbols such that $v(\boldsymbol{A_1}) = T$ and $v(\boldsymbol{A_2}) = T$, but we only really care about what these truth assignments will do to the sentence symbols $\boldsymbol{A_1}$ and $\boldsymbol{A_2}$. More formally, we could define an equivalence relation between truth assignments. Let $\mathcal{K}$ denote the set of all sentence symbols in Σ and $\boldsymbol{\tau}$. Then say that truth assignments v_1 and v_2 are equivalent ($v_1 \sim v_2$) if and only if $v_1\big|_{\mathcal{K}} = v_2\big|_{\mathcal{K}}$. This will clearly be an equivalence relation. We see then that if we take a representative from each equivalence class and compute the truth value for each wff in $\Sigma \cup \{\boldsymbol{\tau}\}$ with each representative truth assignment, we will have computed the truth value of each wff in question for *all* truth assignments.

So in our truth table above, although we really only computed the truth value of $\boldsymbol{((\neg(A_1 \wedge A_2)) \leftrightarrow ((\neg A_1) \vee (\neg A_2))}$ under finitely many truth assignments, this is sufficient since the truth assignments chosen and represented in the table are each representatives of the equivalence classes under the equivalence relation discussed above.

Tautological implication in sentential logic is our model for a logical deduction. It models our intuitive idea that if the premises (the set Σ) of a valid implication are true, then the conclusion of the implication we are trying to establish ($\boldsymbol{\tau}$) cannot fail to be true.

Example 2.37 *Consider* $\Sigma = \{\boldsymbol{A_1}, \boldsymbol{(A_1 \to A_2)}\}$ *and* $\boldsymbol{\tau} = \boldsymbol{A_2}$.

We see here that only for row 1 in the truth table are all of the elements in Σ *satisfied, and we have that* $\boldsymbol{\tau}$ *is also satisfied. We never have a situa-*

A_1	A_2	$(A_1 \to A_2)$
T	T	T
T	F	F
F	T	T
F	F	T

tion where the elements of Σ *are true but* $\boldsymbol{\tau}$ *is false. Thus, we may say that* $\{\boldsymbol{A_1}, (\boldsymbol{A_1} \to \boldsymbol{A_2})\} \vDash \boldsymbol{A_2}$. *This is the logical rule "modus ponens."*

Example 2.38 *Consider* $\Sigma = \{(\neg \boldsymbol{A_1}), (\boldsymbol{A_1} \to \boldsymbol{A_2})\}$ *and* $\boldsymbol{\tau} = (\neg \boldsymbol{A_2})$.

A_1	A_2	$(\neg A_1)$	$(\neg A_2)$	$(A_1 \to A_2)$
T	T	F	F	T
T	F	F	T	F
F	T	T	F	T
F	F	T	T	T

Now we have two rows where the elements of Σ *are satisfied, rows 3 and 4. Notice however that* $(\neg \boldsymbol{A_2})$ *is not satisfied in row 3. So* Σ *does not tautologically imply* $\boldsymbol{\tau}$ *in this case. This represents the fallacy of denying the consequent.*

Example 2.39 *Consider* $\Sigma = \{(\neg \boldsymbol{A_2}), (\boldsymbol{A_1} \to \boldsymbol{A_2})\}$ *and* $\boldsymbol{\tau} = (\neg \boldsymbol{A_1})$.

A_1	A_2	$(\neg A_1)$	$(\neg A_2)$	$(A_1 \to A_2)$
T	T	F	F	T
T	F	F	T	F
F	T	T	F	T
F	F	T	T	T

Row 4 is the only row of the truth table for which the elements of Σ *are satisfied. For this row,* $(\neg \boldsymbol{A_1})$ *is also satisfied and so*

$$\{(\neg \boldsymbol{A_2}), (\boldsymbol{A_1} \to \boldsymbol{A_2})\} \vDash (\neg \boldsymbol{A_1}).$$

This is contraposition.

Having given a few examples that give a sense of what tautological implication is, we now consider a few special cases.

Case 1: What if Σ is empty? It is then a vacuously true statement to say that every truth assignment v satisfies every member of Σ (i.e the implication "If $\boldsymbol{\varphi} \in \Sigma$, then v satisfies $\boldsymbol{\varphi}$" can never be false since the antecedent is false, and thus the statement holds true for any truth assignment v). So, we may say "$\emptyset \vDash \boldsymbol{\tau}$" if and only if every truth assignment satisfies $\boldsymbol{\tau}$ since every truth assignment satisfies the elements of $\emptyset$. In this case we write $\vDash \boldsymbol{\tau}$ and say that $\boldsymbol{\tau}$ is a *tautology*, a statement that is always true.

Example 2.40 $((\neg(\boldsymbol{A_1} \wedge \boldsymbol{A_2})) \leftrightarrow ((\neg \boldsymbol{A_1}) \vee (\neg \boldsymbol{A_2}))$ *is a tautology since it is always true, no matter what truth assignment we use i.e. no matter what row of the truth table (see Example* 2.34*) we are considering. Of course this is one of DeMorgan's Laws. We may verify all of the standard tautologies presented in a mathematics foundations course in a similar fashion using truth tables.*

Case 2: What if the wffs in Σ cannot be satisfied? If this is the case, then *any* wff $\boldsymbol{\tau}$ is tautologically implied by Σ. This is so because of our definition of tautological implication. For $\boldsymbol{\tau}$ to be tautologically implied by Σ, *if* a truth assignment v satisfies every wff in Σ, then v must also satisfy $\boldsymbol{\tau}$. So if a truth assignment *cannot* satisfy all the elements of Σ it is vacuously true that $\boldsymbol{\tau}$ is tautologically implied by Σ.

Example 2.41 $\{\boldsymbol{A_1}, (\neg \boldsymbol{A_1})\} \vDash \boldsymbol{A_2}$ *since no truth assignment can satisfy both* $(\neg \boldsymbol{A_1})$ *and* $\boldsymbol{A_1}$*. Therefore, every truth assignment which satisfies all the ele-*

ments of $\{\boldsymbol{A_1}, (\neg \boldsymbol{A_1})\}$ *(there are none) will also satisfy* $\boldsymbol{A_2}$*. The idea is that any statement follows from a contradiction.*

Case 3: What if Σ is a singleton i.e. $\Sigma = \{\boldsymbol{\sigma}\}$? Instead of writing $\{\boldsymbol{\sigma}\} \models \boldsymbol{\tau}$, we write $\boldsymbol{\sigma} \models \boldsymbol{\tau}$. If $\boldsymbol{\sigma} \models \boldsymbol{\tau}$ and $\boldsymbol{\tau} \models \boldsymbol{\sigma}$ we say that $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ are *tautologically equivalent*, and we write $\boldsymbol{\sigma} \models\!\models \boldsymbol{\tau}$. For sentential logic, this gives us our notion of logical equivalence.

Example 2.42 *From Example* 2.34*, it is clear that*

$$((\neg(\boldsymbol{A_1} \wedge \boldsymbol{A_2})) \models ((\neg \boldsymbol{A_1}) \vee (\neg \boldsymbol{A_2})) \textit{ and}$$

$$((\neg \boldsymbol{A_1}) \vee (\neg \boldsymbol{A_2})) \models ((\neg(\boldsymbol{A_1} \wedge \boldsymbol{A_2})).$$

Thus, $((\neg \boldsymbol{A_1}) \vee (\neg \boldsymbol{A_2})) \models\!\models ((\neg(\boldsymbol{A_1} \wedge \boldsymbol{A_2}))$.

We also saw in Example 2.40 that

$$((\neg(\boldsymbol{A_1} \wedge \boldsymbol{A_2})) \leftrightarrow ((\neg \boldsymbol{A_1}) \vee (\neg \boldsymbol{A_2}))$$

was a tautology. Our examples with one of DeMorgan's Laws suggest the following results.

Theorem 2.43 $\boldsymbol{\alpha} \models \boldsymbol{\beta}$ *iff* $\models (\boldsymbol{\alpha} \rightarrow \boldsymbol{\beta})$

Proof:

$$\begin{aligned} \boldsymbol{\alpha} \models \boldsymbol{\beta} \quad &\text{iff} \quad \overline{v}(\boldsymbol{\alpha}) = T \text{ implies } \overline{v}(\boldsymbol{\beta}) = T \\ &\text{iff} \quad \overline{v}(\boldsymbol{\alpha}) = T \text{ and } \overline{v}(\boldsymbol{\beta}) = F \text{ cannot both happen.} \\ &\text{iff} \quad \overline{v}((\boldsymbol{\alpha} \rightarrow \boldsymbol{\beta})) = T \text{ for every truth assignment.} \\ &\text{iff} \quad \models (\boldsymbol{\alpha} \rightarrow \boldsymbol{\beta}) \end{aligned}$$

∎

Corollary 2.43.1 $\boldsymbol{\alpha} \models\!\!\!\!\dashv \boldsymbol{\beta}$ *iff* $\models (\boldsymbol{\alpha} \leftrightarrow \boldsymbol{\beta})$

Proof:

$$\begin{aligned}
\boldsymbol{\alpha} \models\!\!\!\!\dashv \boldsymbol{\beta} \quad &\text{iff} \quad \boldsymbol{\alpha} \models \boldsymbol{\beta} \text{ and } \boldsymbol{\beta} \models \boldsymbol{\alpha} \\
&\text{iff} \quad \models (\boldsymbol{\alpha} \to \boldsymbol{\beta}) \text{ and } \models (\boldsymbol{\beta} \to \boldsymbol{\alpha}) \\
&\text{iff} \quad \text{neither } \overline{v}(\boldsymbol{\alpha}) = T \text{ and } \overline{v}(\boldsymbol{\beta}) = F \text{ nor } \overline{v}(\boldsymbol{\beta}) = T \text{ and } \overline{v}(\boldsymbol{\alpha}) = F \\
&\text{iff} \quad \overline{v}(\boldsymbol{\alpha}) = \overline{v}(\boldsymbol{\beta}) \text{ for every truth assignment.} \\
&\text{iff} \quad \models (\boldsymbol{\alpha} \leftrightarrow \boldsymbol{\beta})
\end{aligned}$$

∎

Theorem 2.44 $\Sigma \cup \{\boldsymbol{\alpha}\} \models \boldsymbol{\beta}$ *iff* $\Sigma \models (\boldsymbol{\alpha} \to \boldsymbol{\beta})$

Proof: Since this proof should follow easily from the appropriate definitions, it is left to the reader. ∎

The interesting thing about these theorems and their proofs is the perspective they yield about talking in a logical mathematical way about the mathematical topic of logic. In Theorem 2.43.1, the statement on the left is a statement about the sentential model of logical equivalence existing between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in the formal language, but this statement exists *outside* of the formal sentential language. However, the statement on the right gives a statement about it always being true that $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are equivalent *within* the formal language. (The intended meaning of $\leftrightarrow$ being logical equivalence). The theorem thus indicates that logical equivalence (in the sentential model) between two statements in the formal sentential language, can be translated into the formal language itself where the logical symbol $\leftrightarrow$ is the formal language symbol for equivalence.

Perhaps even more of a mind-bender is the fact that this theorem *itself* which is a statement about logical equivalence can be translated into the formal language. We will let $\boldsymbol{A_1}$ translate the statement "There are two wffs $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ that are tautologically equivalent" (i.e. "$\boldsymbol{\alpha} \models\mid \boldsymbol{\beta}$"). Now we will let $\boldsymbol{A_2}$ translate the statement "The wff $\boldsymbol{(\alpha \leftrightarrow \beta)}$ is a tautology" (i.e. "$\models \boldsymbol{(\alpha \leftrightarrow \beta)}$"). The theorem above says that $\boldsymbol{A_1} \models \boldsymbol{A_2}$ and $\boldsymbol{A_2} \models \boldsymbol{A_1}$ (or the same thing $\boldsymbol{A_1} \models\mid \boldsymbol{A_2}$). But then the theorem says that this is logically equivalent to $\models \boldsymbol{(A_1 \leftrightarrow A_2)}$. So our theorem can be modeled in the formal sentential language. At this point the infinite "hall of mirrors" effect is beginning to set in. To combat the disorientation, we must recognize that we are operating at two levels, a meta-level and the formal level. When we say, "The theorem above says $\boldsymbol{A_1} \models\mid \boldsymbol{A_2}$," we are thinking at the formal level *within* our sentential model. Any formal satisfaction of the wff $\boldsymbol{A_1}$ will also formally satisfy the wff $\boldsymbol{A_2}$. When we say "But then the theorem says that this is logically equivalent to $\models \boldsymbol{(A_1 \leftrightarrow A_2)}$," we are stepping *outside* of the formal model of sentential logic to the meta-level of us being rational beings who understand logical equivalence and what that *means*. In this case, we understand that the statement "$\boldsymbol{A_1} \models\mid \boldsymbol{A_2}$" is logically equivalent (this equivalence is at the meta-level) to the statement "$\models \boldsymbol{(A_1 \leftrightarrow A_2)}$" given the theorem we proved. At the meta-level, we are deducing things about our model of deduction (in this case, sentential logic). At the formal level, we are modeling within a formal language the deduction about our model of deduction. We as the readers and mathematicians stand in the real world making real deductions about the models of our real deductions.

This self-reference is the sort that comes about from studying logic mathematically when mathematics itself is based on logic. It is also just this sort of self-reference that is the very key to proving Gödel's Incompleteness Theorem.

We have established the basics of sentential logic as a model for humanity's deductive thought processes. We are now in a position to explore some of the properties of this model.

Chapter 3

Properties of Sentential Logic

Having developed sentential logic as a model of humanity's deductive thought processes, we are in a position to ask about what nice properties sentential logic has. Recall that our goal is to develop mathematically rigorous models of deduction, and like other mathematical constructs, we would like to ask what nice properties a particular model has and what properties it lacks. Really, what we are asking is how well does sentential logic model humanity's deductive thought processes?

3.1 The Compactness Theorem

We recall first that saying "$\Sigma \models \boldsymbol{\tau}$" is our model for saying "If the hypotheses of the statements (wffs) in Σ are satisfied, then the statement (wff) $\boldsymbol{\tau}$ must also be satisfied." Now, nothing was said about the cardinality of the set Σ. By assumption, we have a countable alphabet of logical symbols and sentence symbols, and expressions are finite sequences of indecomposable alphabet symbols. Hence there are countably many possible wffs, and the set Σ of wffs is countable.

The question becomes, if it is the case that $\Sigma \vDash \boldsymbol{\tau}$ where Σ is countably infinite, is it possible to give a *proof* that $\Sigma \vDash \boldsymbol{\tau}$?

It is instructive at this point to consider what we mean by the term *proof*. Remember that at our current sentential level, we are thinking of "$\Sigma \vDash \boldsymbol{\tau}$" as a model for saying that the statements or premises of Σ logically imply the statement $\boldsymbol{\tau}$. To give a proof of "$\Sigma \vDash \boldsymbol{\tau}$", we use the properties of sentential logic that we have developed as part of it being a mathematical structure to fulfill the definition for what it means to say "$\Sigma \vDash \boldsymbol{\tau}$." Here, our proof consists of finitely many statements at the meta-level discussed at the end of Chapter 2. What we are doing as reasoners in the real world (the meta-level) is examining a statement in a mathematical structure (sentential logic) and giving sufficient meta-reasons to be able to say "$\Sigma \vDash \boldsymbol{\tau}$" in our mathematical structure. This is no different than if we were working in number theory seeking to establish a proof of the statement "There are infinitely many prime numbers." Our proof of "$\Sigma \vDash \boldsymbol{\tau}$" is not a statement within the formal sentential logic structure, just as our proof of there being infinitely many prime numbers does not take place in the formal system of number theory itself. We only use the *structure* of number theory to carry out a reasonable proof.

The question then arises, if we carry out proofs as part of our deductive thought processes as humans, can we model our proofs in the sentential structure itself? What is characteristic about a proof is a *demonstration* of *how* the truth of finitely many premises guarantees the truth of a conclusion. At the sentential logic level we can handle the truth of finitely many premises guaranteeing the truth of a conclusion. This structure we can express in the sentential model of deduction as "$\Sigma_f \vDash \boldsymbol{\tau}$" where $\Sigma_f = \{\boldsymbol{\sigma_0}, \boldsymbol{\sigma_1}, \ldots, \boldsymbol{\sigma_n}\}$ (a finite set of wffs that translates our finitely many premises as $\boldsymbol{\sigma_0}$ through $\boldsymbol{\sigma_n}$).

We may even express our tautological implication in the symbolism of the sentential language as $(\boldsymbol{\sigma_0} \rightarrow (\boldsymbol{\sigma_1} \rightarrow (\ldots \rightarrow \boldsymbol{\sigma_n}))\ldots) \rightarrow \boldsymbol{\tau}$ since this wff will be a tautology using Theorem 2.44 finitely many times.

This representation of a proof is the best we can do with sentential logic. Given the *meaning* we have intended for the symbols involved, the above wff definitely models the guarantee of the truth of the statement $\boldsymbol{\tau}$ given the truth of the premises of Σ_f. Analysing the wff via truth table will show that given any truth assignment v, $\overline{v}$ will always return an output of T given the wff as an input. Or from the computer science perspective, if we program a computer to return F for an always false wff, T for an always true wff, or "inconclusive" for a sometimes true and sometimes false wff, given our wff $(\boldsymbol{\sigma_0} \rightarrow (\boldsymbol{\sigma_1} \rightarrow (\ldots \rightarrow \boldsymbol{\sigma_n}))\ldots) \rightarrow \boldsymbol{\tau}$ as an input into the computer, the computer would return T.

We now measure against our intuitions whether our model for a proof in the sentential realm is a good one. Our model seems limited. A proof should clearly demonstrate the *how* and the *why* behind the guarantees of truth. In other words the ultimate *"why"* behind writing $\rightarrow$ in $(\boldsymbol{\alpha} \rightarrow \boldsymbol{\beta})$ where this wff is a tautology. Any hints as to this "why" within the formal language itself must come from the very structure of the symbolism being used and interaction of truth values of the wffs. So, perhaps at the most rudimentary level, our model for a proof in sentential approaches what a proof actually is, but it seems unsatisfying.

However, at this point we recall that sentential logic is a *model* for the deductive process that human beings use. Of course any mathematical model will have limitations since it is just that: a model. It is an approximation to the real world where we compress the real world's complexities into something simpler and easier to think about. The properties we then derive from the model may give us some useful information to use in the real world. We also recall that sentential logic, is only our *first* model for humanity's deductive thought processes. We see that refinements of the model may be in order. We now return to our original discussion.

Again, the question is, if it is the case that "$\Sigma \models \boldsymbol{\tau}$" where can we *always* give a proof (at the real world level) of this fact? The interesting case occurs when Σ is countably infinite. Our main tool at this point for establishing a statement of this kind is to use the definition; i.e. if an arbitrary truth assignment satisfies the wffs in Σ, then it must also satisfy the wff $\boldsymbol{\tau}$. Proving this statement becomes a problem if Σ is infinite for then we necessarily have infinitely many sentence symbols $\boldsymbol{A_i}$ and thus infinitely many truth assignments to check (or equivalently, infinitely many rows in our truth table). Thus, if Σ is infinite, we can never prove $\Sigma \models \boldsymbol{\tau}$ directly from the definition+. We need other results which we now develop.

First, recall that we say that truth assignment $v : \mathcal{S} \longrightarrow \{T, F\}$ *satisfies* wff $\boldsymbol{\varphi}$ if and only if $\overline{v}(\boldsymbol{\varphi}) = T$.

Definition 3.1 *A set Σ of wffs is* **satisfiable** *if there is a truth assignment which satisfies every member of Σ.*

Lemma 3.1.1 *If every finite subset of a set of wffs Σ is satisfiable, then the same is true of at least one of the sets $\Sigma \cup \{\boldsymbol{\alpha}\}$ and $\Sigma \cup \{(\neg\boldsymbol{\alpha})\}$ for any wff $\boldsymbol{\alpha}$.*

Proof: If either $\boldsymbol{\alpha}$ or $(\neg\boldsymbol{\alpha})$ is in Σ, the result holds trivially. We assume then that neither $\boldsymbol{\alpha}$ nor $(\neg\boldsymbol{\alpha})$ are in Σ, and thus $(\neg\boldsymbol{\alpha}) \notin \Sigma \cup \{\boldsymbol{\alpha}\}$ and $\boldsymbol{\alpha} \notin \Sigma \cup \{(\neg\boldsymbol{\alpha})\}$. Suppose, by way of contradiction that there are finite sets F_1

and F_2 with $F_1 \subseteq \Sigma \cup \{\boldsymbol{\alpha}\}$ and $F_2 \subseteq \Sigma \cup \{(\neg\boldsymbol{\alpha})\}$ such that no truth assignments satisfy either F_1 or F_2. Note that since F_1 and F_2 are finite, $F_1 \cup F_2$ will also be finite as will $F_1 \cup F_2 - \{\boldsymbol{\alpha}, (\neg\boldsymbol{\alpha})\}$. This set is a subset of Σ. Since Σ has the property that every finite subset of Σ is satisfiable, there is a truth assignment v satisfying every member of $F_1 \cup F_2 - \{\boldsymbol{\alpha}, (\neg\boldsymbol{\alpha})\}$. In particular, v satisfies every member of $F_1 - \{\boldsymbol{\alpha}, (\neg\boldsymbol{\alpha})\}$ and of $F_2 - \{\boldsymbol{\alpha}, (\neg\boldsymbol{\alpha})\}$ which are each subsets of $F_1 \cup F_2 - \{\boldsymbol{\alpha}, (\neg\boldsymbol{\alpha})\}$. Since $(\neg\boldsymbol{\alpha}) \notin F_1 \subseteq \Sigma \cup \{\boldsymbol{\alpha}\}$ and $\boldsymbol{\alpha} \notin F_2 \subseteq \Sigma \cup \{(\neg\boldsymbol{\alpha})\}$, $F_1 - \{\boldsymbol{\alpha}, (\neg\boldsymbol{\alpha})\} = F_1 - \{\boldsymbol{\alpha}\}$, and $F_2 - \{\boldsymbol{\alpha}, (\neg\boldsymbol{\alpha})\} = F_2 - \{(\neg\boldsymbol{\alpha})\}$.

Now, either $\overline{v}(\boldsymbol{\alpha}) = T$ or $\overline{v}(\boldsymbol{\alpha}) = F$. In the first case, F_1 would have to be satisfiable since $(F_1 - \{\boldsymbol{\alpha}\}) \cup \{\boldsymbol{\alpha}\} = F_1$. This contradicts our assumption of F_1 not being satisfiable, and we conclude that $\overline{v}(\boldsymbol{\alpha}) = F$. But by properties of $\overline{v}$ established in Chapter 2, this must mean that $\overline{v}((\neg\boldsymbol{\alpha})) = T$. In this case F_2 must be satisfiable by similar reasoning to that in the previous case. But this is another contradiction to what we have assumed. We thus conclude that what we have supposed is false and every finite subset is satisfiable for at least one of $\Sigma \cup \{\boldsymbol{\alpha}\}$ and $\Sigma \cup \{(\neg\boldsymbol{\alpha})\}$. ∎

We need one more lemma before we prove a result that will help us to answer our question under consideration.

Lemma 3.1.2 *Let Δ be a set of wffs such that (i) every finite subset of Δ is satisfiable, and (ii) for every wff $\boldsymbol{\alpha}$, either $\boldsymbol{\alpha} \in \Delta$ or $(\neg\boldsymbol{\alpha}) \in \Delta$ (this will be an exclusive "or" since $\{\boldsymbol{\alpha}, (\neg\boldsymbol{\alpha})\}$ would not be a satisfiable subset of Δ). Define the truth assignment v by*

$$v(\boldsymbol{A}) = \begin{cases} T & \text{if } \boldsymbol{A} \in \Delta \\ F & \text{if } \boldsymbol{A} \notin \Delta \end{cases}$$

for each sentence symbol $\boldsymbol{A}$. Then for every wff $\boldsymbol{\varphi}$, $\overline{v}(\boldsymbol{\varphi}) = T$ iff $\boldsymbol{\varphi} \in \Delta$.

Proof: Recall that for sentential logic, we let $\mathcal{S}$ denote the set of sentence symbols and $\mathcal{W}$ denote the set of wffs. Let

$$I = \{\boldsymbol{\varphi} \in \mathcal{W} | \boldsymbol{\varphi} \in \Delta \text{ iff } \overline{v}(\boldsymbol{\varphi}) = T\}$$

(v is the truth assignment defined above). We wish to use induction, that is, the Induction Principle, to prove that $I = \mathcal{W}$.

Note that $\mathcal{S} \subseteq I$ since for every sentence symbol $\boldsymbol{A}$, $\overline{v}(\boldsymbol{A}) = v(\boldsymbol{A}) = T$ if and only if $\boldsymbol{A} \in \Delta$ by how v was defined. If we can demonstrate that I is closed under the formula building operations, we will have fulfilled the Induction Principle.

Let $\boldsymbol{\alpha}, \boldsymbol{\beta} \in I$. Note that this means that $\overline{v}(\boldsymbol{\alpha}) = T$ iff $\boldsymbol{\alpha} \in \Delta$ and $\overline{v}(\boldsymbol{\beta}) = T$ iff $\boldsymbol{\beta} \in \Delta$. We show that $\boldsymbol{(\neg\alpha)} \in I$. We know from our results using the Recursion Theorem with truth assignments that $\overline{v}(\boldsymbol{(\neg\alpha)}) = T$ iff $\overline{v}(\boldsymbol{\alpha}) = F$ iff $\boldsymbol{\alpha} \notin \Delta$. We show that $\boldsymbol{\alpha} \notin \Delta$ iff $\boldsymbol{(\neg\alpha)} \in \Delta$. By property (ii) of Δ, we have that $\boldsymbol{\alpha} \notin \Delta$ implies that $\boldsymbol{(\neg\alpha)} \in \Delta$. Conversely, if $\boldsymbol{(\neg\alpha)} \in \Delta$, then $\boldsymbol{\alpha} \notin \Delta$. Thus, we have that $\boldsymbol{(\neg\alpha)} \in \Delta$ iff $\boldsymbol{\alpha} \notin \Delta$. So, we have that $\overline{v}(\boldsymbol{(\neg\alpha)}) = T$ iff $\boldsymbol{\alpha} \notin \Delta$ iff $\boldsymbol{(\neg\alpha)} \in \Delta$. Hence $\boldsymbol{(\neg\alpha)} \in I$ by definition, and I is closed under $\mathcal{F}_{\neg}$.

The same reasoning used here to show that I is closed under $\mathcal{F}_{\neg}$ may be used in the remaining cases of the formula building operations. The demonstration of all of these cases is very tedious. We will illustrate the proofs for the remaining formula building operations with one two-variable formula building operation, and then we leave it to the reader to establish the remaining cases.

We seek to demonstrate that $\boldsymbol{(\alpha \vee \beta)} \in I$. By the Recursion Theorem, we note that $\overline{v}(\boldsymbol{(\alpha \vee \beta)}) = T$ iff $\overline{v}(\boldsymbol{\alpha}) \vee \overline{v}(\boldsymbol{\beta}) = T$ iff $\overline{v}(\boldsymbol{\alpha}) = T$ or $\overline{v}(\boldsymbol{\beta}) = T$ iff $\boldsymbol{\alpha} \in \Delta$ or $\boldsymbol{\beta} \in \Delta$, since $\boldsymbol{\alpha}, \boldsymbol{\beta} \in I$. The following piece of a truth table will be helpful in establishing the claim that $\boldsymbol{\alpha} \in \Delta$ or $\boldsymbol{\beta} \in \Delta$ iff $\boldsymbol{(\alpha \vee \beta)} \in \Delta$.

$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	$(\neg\boldsymbol{\alpha})$	$(\neg\boldsymbol{\beta})$	$(\boldsymbol{\alpha} \vee \boldsymbol{\beta})$	$(\neg(\boldsymbol{\alpha} \vee \boldsymbol{\beta}))$
T	T	F	F	T	F
T	F	F	T	T	F
F	T	T	F	T	F
F	F	T	T	F	T

We see from the Recursion Theorem that all possibilities for the truth values for the key wffs $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ have been exhausted under any truth assignment. We see immediately from the truth table that the finite sets

$$\{\boldsymbol{\alpha}, (\neg(\boldsymbol{\alpha} \vee \boldsymbol{\beta}))\} \text{ and } \{\boldsymbol{\beta}, (\neg(\boldsymbol{\alpha} \vee \boldsymbol{\beta}))\}$$

are each not satisfiable under any possible truth assignment since the sets' two wffs never have the value of truth under the same truth assignment. If we assume that either $\boldsymbol{\alpha} \in \Delta$ or $\boldsymbol{\beta} \in \Delta$, then we must conclude by property (i) of Δ that $(\neg(\boldsymbol{\alpha} \vee \boldsymbol{\beta})) \notin \Delta$, otherwise one of the above sets would be satisfiable. So, by property (ii) of Δ, $(\boldsymbol{\alpha} \vee \boldsymbol{\beta}) \in \Delta$. Conversely, if we suppose that $(\boldsymbol{\alpha} \vee \boldsymbol{\beta}) \in \Delta$, then we note that $\{(\neg\boldsymbol{\alpha}), (\neg\boldsymbol{\beta}), (\boldsymbol{\alpha} \vee \boldsymbol{\beta})\}$ is not satisfiable as seen by the truth table above. Since we are assuming that $(\boldsymbol{\alpha} \vee \boldsymbol{\beta}) \in \Delta$, by property (i), we have that either $(\neg\boldsymbol{\alpha}) \notin \Delta$ (in which case $\boldsymbol{\alpha} \in \Delta$ by property (ii)) or $(\neg\boldsymbol{\beta}) \notin \Delta$ (in which case $\boldsymbol{\beta} \in \Delta$ by property (ii)). So $(\boldsymbol{\alpha} \vee \boldsymbol{\beta}) \in \Delta$ implies that $\boldsymbol{\alpha} \in \Delta$ or that $\boldsymbol{\beta} \in \Delta$. Thus, $(\boldsymbol{\alpha} \vee \boldsymbol{\beta}) \in \Delta$ iff $\boldsymbol{\alpha} \in \Delta$ or $\boldsymbol{\beta} \in \Delta$. So, we conclude that $\overline{v}((\boldsymbol{\alpha} \vee \boldsymbol{\beta})) = T$ iff $\boldsymbol{\alpha} \in \Delta$ or $\boldsymbol{\beta} \in \Delta$ iff $(\boldsymbol{\alpha} \vee \boldsymbol{\beta}) \in \Delta$. Hence, $(\boldsymbol{\alpha} \vee \boldsymbol{\beta}) \in I$ by definition and I is closed under $\mathcal{F}_{\vee}$. The cases with the other formula building operations are similar and are left to the reader.

Since I contains $\mathcal{S}$ and is closed under the formula building operations, we conclude by the Induction Principle that $I = \mathcal{W}$. So, for any wff $\boldsymbol{\varphi}$, $\overline{v}(\boldsymbol{\varphi}) = T$ iff $\boldsymbol{\varphi} \in \Delta$. We thus accept this lemma as proved. ∎

We are now in position to prove the result that will answer our question at hand.

Theorem 3.2 (The Compactness Theorem for Sentential Logic)
A set of wffs is satisfiable if and only if every finite subset is satisfiable.

Proof: Let Σ be a set of wffs. If Σ is satisfiable, then of course, every finite subset will be satisfiable. Just take the truth assignment that satisfies Σ and this will satisfy all of Σ's subsets, finite or not. Conversely, suppose that every finite subset of Σ is satisfiable. If Σ is itself finite, we are done since Σ is a subset of itself, and being finite, is satisfiable by assumption. So, we suppose that Σ is infinite and that every finite subset of Σ is satisfiable. The idea is to construct a maximal set of wffs which contains Σ and define a truth assignment which satisfies every member of this maximal set. Such a truth assignment will of course satisfy Σ as well. We proceed.

Let $\boldsymbol{\alpha_1},\ \boldsymbol{\alpha_2},\ \boldsymbol{\alpha_3}, \ldots$ be an enumeration of the wffs (the set of wffs is countable). Define the following:

$$\Delta_0 = \Sigma$$

$$\Delta_{n+1} = \begin{cases} \Delta_n \cup \{\boldsymbol{\alpha_{n+1}}\} & \text{if every finite subset of this set is satisfiable} \\ \Delta_n \cup \{(\neg \boldsymbol{\alpha_{n+1}})\} & \text{otherwise} \end{cases}$$

Using a simple induction argument and the Lemma 3.1.1, we know that every finite subset of Δ_n for each natural number n is satisfiable. Every finite subset of $\Delta_0 = \Sigma$ is satisfiable by assumption. If every finite subset of Δ_{n-1} for $n \geq 1$ is satisfiable, then the same holds true for at least one of $\Delta_{n-1} \cup \{\boldsymbol{\alpha_n}\}$ and $\Delta_{n-1} \cup \{(\neg \boldsymbol{\alpha_n})\}$ by the lemma. Let $\Delta = \bigcup_{n=0}^{\infty} \Delta_n$. First, it is clear that $\Sigma = \Delta_0 \subseteq \Delta$. Let F be a finite subset of Δ. Now, $F \subseteq \Delta_n$ for some $n \in \mathbb{N}$. Since every finite subset of Δ_n is satisfiable, then F is satisfiable. Since our choice for F as a finite subset of Δ was arbitrary, then we know that every finite subset of Δ is satisfiable.

Now let $\boldsymbol{\varphi}$ be a wff. By our enumeration of the set of wffs, $\boldsymbol{\varphi} = \boldsymbol{\alpha_i}$ for some natural number i. By construction either $\boldsymbol{\varphi} = \boldsymbol{\alpha_i} \in \Delta_i$ or $(\neg\boldsymbol{\varphi}) = (\neg\boldsymbol{\alpha_i}) \in \Delta_i$. In either case, we have that either $\boldsymbol{\varphi} \in \Delta$ or $(\neg\boldsymbol{\varphi}) \in \Delta$. Since our choice for $\boldsymbol{\varphi}$ was arbitrary, this is true for any wff $\boldsymbol{\varphi}$. We define the truth assignment v as in Lemma 3.1.2, and using the lemma we find that for any wff $\boldsymbol{\varphi}$, $\overline{v}(\boldsymbol{\varphi}) = T$ if and only if $\boldsymbol{\varphi} \in \Delta$. Since for every wff $\boldsymbol{\varphi} \in \Sigma$, $\boldsymbol{\varphi} \in \Delta$, we see that $\overline{v}(\boldsymbol{\varphi}) = T$ for every $\boldsymbol{\varphi} \in \Sigma$. Hence, the truth assignment v satisfies Σ by definition. Therefore, the assertion of the Compactness Theorem holds. ∎

Note that there are two alternative proofs to the Compactness Theorem to the one given above that may be of interest to the reader. The first uses Zorn's Lemma (equivalent to the Axiom of Choice), and in this case the Compactness Theorem can be shown to hold for an uncountable alphabet of sentential symbols. The second alternative uses general topological concepts. The compactness theorem asserts the compactness of a particular topological space called a Stone Space. The interested reader may wish to do some research on this line.

Corollary 3.2.1 *If $\Sigma \vDash \boldsymbol{\tau}$, then there is a finite $\Sigma_f \subseteq \Sigma$ such that $\Sigma_f \vDash \boldsymbol{\tau}$.*

Proof: Assume that $\Sigma \vDash \boldsymbol{\tau}$. A moment's reflection shows that $\Sigma \vDash \boldsymbol{\tau}$ if and only if $\Sigma \cup \{(\neg\boldsymbol{\tau})\}$ is unsatisfiable. Suppose by way of contradiction that for every finite $\Sigma_f \subseteq \Sigma$ it is not the case that $\Sigma_f \vDash \boldsymbol{\tau}$. By our first observation then $\Sigma_f \cup \{(\neg\boldsymbol{\tau})\}$ is satisfiable for every finite $\Sigma_f \subseteq \Sigma$. Note that a finite subset of $\Sigma \cup \{(\neg\boldsymbol{\tau})\}$ will either be of the form Σ_f or $\Sigma_f \cup \{(\neg\boldsymbol{\tau})\}$ for finite $\Sigma_f \subseteq \Sigma$. Since $\Sigma_f \cup \{(\neg\boldsymbol{\tau})\}$ is satisfiable for any finite $\Sigma_f \subseteq \Sigma$ and $\Sigma_f \subseteq \Sigma_f \cup \{(\neg\boldsymbol{\tau})\}$, we see that every finite subset of $\Sigma \cup \{(\neg\boldsymbol{\tau})\}$ is satisfiable. By the Compactness Theorem then, $\Sigma \cup \{(\neg\boldsymbol{\tau})\}$ is itself satisfiable. But, by our initial observation, this is so if and only if it is not the case that $\Sigma \models \boldsymbol{\tau}$, a contradiction. So, in fact it must be the case that $\Sigma_f \models \boldsymbol{\tau}$ for some finite $\Sigma_f \subseteq \Sigma$. ∎

Thus, we are guaranteed that if $\Sigma \models \boldsymbol{\tau}$, we should be able to find finite $\Sigma_f \subseteq \Sigma$ to establish $\Sigma_f \models \boldsymbol{\tau}$. Hence, we are always able to establish $\Sigma \models \boldsymbol{\tau}$ using our truth table method in the meta-realm. Since $\Sigma \models \boldsymbol{\tau}$ is our model in the sentential language for the truth of the statements of Σ guaranteeing the statements of $\boldsymbol{\tau}$, the above theorem also guarantees that we need only finitely many premises (Σ_f) to make this guarantee. That is, this corollary to the Compactness Theorem guarantees that our intuitive notion that a proof of a true statement should be finite in length (and thus physically doable) is matched in the sentential model for deductive thought. Hence our sentential does a decent job at mirroring our intuitive notions of proof construction. We now ponder another desirable property.

3.2 Expressibility in Sentential Logic

Another question we wish to ask about our model for humanity's deductive thought processes is "How do we know that we can't enrich the sentential language by adding more connectives?" Have we included enough sentential connectives (our current connective symbols are $\neg$, $\vee$, $\wedge$, $\rightarrow$, and $\leftrightarrow$) to express all possible logical constructions that could occur given our intended meanings for our connectives?

An example clarifies exactly what we mean by this question. Suppose we enrich our sentential language with a connective symbol $\boldsymbol{\nabla}$. That is, we have a new ternary formula building operation $\mathcal{F}_{\nabla}$ which we define on the set of expressions in the enriched sentential language by $\mathcal{F}_{\nabla}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = (\boldsymbol{\nabla}\boldsymbol{\alpha}\boldsymbol{\beta}\boldsymbol{\gamma})$.

So in our new language, $(\boldsymbol{\nabla\alpha\beta\gamma})$ is now an expression for expressions $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$. Now, we also must have an intended meaning for $\boldsymbol{\nabla}$ as we do with our original connectives (i.e. the intended meaning for $\wedge$ is "and", the intended meaning for $\rightarrow$ is "implies", etc.). We will call $\boldsymbol{\nabla}$ the "too good to be true" operator. To understand our intended meaning, let us define $\mathcal{G}_{\nabla}\{T,F\}^3 \longrightarrow \{T,F\}$ as follows:

$$\mathcal{G}_{\nabla}(V_1,V_2,V_2) = \begin{cases} F & \text{if } V_i = V_j = T \text{ where } i \neq j \\ T & \text{if } V_i = V_j = F \text{ where } i \neq j \end{cases}$$

(This function is well defined since the truth value of two of the inputs must match.) So, for instance $(\boldsymbol{\nabla}TTT) = (\boldsymbol{\nabla}TTF) = F$ (too good to be true), and $(\boldsymbol{\nabla}TFF) = (\boldsymbol{\nabla}FFF) = T$ (we could also call this the "pessimist function"). So, using the Recursion Theorem, for a truth assignment v,

$$\overline{v}((\boldsymbol{\nabla\alpha_1\alpha_2\alpha_3})) = \begin{cases} F & \text{if } \overline{v}(\alpha_i) = \overline{v}(\alpha_j) = T \text{ where } i \neq j \\ T & \text{if } \overline{v}(\alpha_i) = \overline{v}(\alpha_j) = F \text{ where } i \neq j \end{cases}$$

So, having enriched our original sentential language with the "too good to be true" symbol, can we express more logical structure now in our new language than we could before?

First, we need to make the notion of "expressing more logical structure" more precise. Recall that two wffs $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ are said to be *tautologically equivalent* if and only if $\boldsymbol{\sigma} \models\!\!\!\!\!\dashv \boldsymbol{\tau}$. We will declare that if any wff in an extended language (extended in the sense just exemplified) is tautologically equivalent (in the extended language) to a wff in the original language, then our extended language can express precisely the same logical structure as can the original language and so we have gained nothing by our extension.

Example 3.3 $(\boldsymbol{\nabla\alpha\beta\gamma}) \models\!\!\!\!\!\dashv (\neg(((\boldsymbol{\alpha} \wedge \boldsymbol{\beta}) \vee (\boldsymbol{\alpha} \wedge \boldsymbol{\gamma})) \vee (\boldsymbol{\beta} \wedge \boldsymbol{\gamma})))$.

The reader can check this assertion via a truth table.

The former example indicates that our "too good to be true" connective does nothing significant to improve the expressibility of our language since there is a tautologically equivalent wff in the original language. However, we would like to see a language (hopefully the sentential language we have developed!) for which *any* extension by *any* connective symbol will result in no improvement of expressibility. The sentential language in fact fits the bill (in fact a much smaller language than the sentential language would suffice), but we need to shift our focus to a new tool to be able to prove this fact.

Definition 3.4 *A function $B^k : \{T, F\}^k \longrightarrow \{T, F\}$ with $k \geq 1$ is a* ***k*-place Boolean function***. A* **Boolean function** *is a k-place Boolean function for some $k \geq 1$.*

Example 3.5 *The following catalog is a list of Boolean functions.*

$$I_i^k(V_1, V_2, \ldots, V_i, \ldots, V_k) = V_i$$

$$\begin{array}{cc} N(T) = F & N(F) = T \\ A(T,T) = T & A(V,F) = F \\ O(F,F) = F & O(V,T) = T \\ C(T,F) = F & C(F,V) = C(V,T) = T \\ E(V_1,V_2) = T & E(V_1,V_2) = F \\ \textit{if } V_1 = V_2 & \textit{if } V_1 \neq V_2 \end{array}$$

The use of Boolean functions becomes apparent in the following truth table.

$\boldsymbol{A_1}$	$\boldsymbol{A_2}$	$(\boldsymbol{A_1} \wedge \boldsymbol{A_2})$	$A(v(\boldsymbol{A_1}), v(\boldsymbol{A_2}))$
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F

We see that the 2-place Boolean function A evaluated at the truth values of the sentence symbols involved in the wff $(\boldsymbol{A_1} \wedge \boldsymbol{A_2})$ gives the same output in every case that $\overline{v}((\boldsymbol{A_1} \wedge \boldsymbol{A_2}))$ does. The Boolean function A expresses the same intended logical structure as $\wedge$ does. It becomes readily apparent that any wff realizes a Boolean function. Given a wff $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}$ involves a finite number of sentence symbols $\boldsymbol{A_1}, \boldsymbol{A_2}, \ldots, \boldsymbol{A_k}$. There will be 2^k possible distinct truth assignments, v for these sentence symbols, but we may define $B^k_{\boldsymbol{\alpha}} : \{T, F\}^k \longrightarrow \{T, F\}$ as follows: $B^k_{\boldsymbol{\alpha}}(v(\boldsymbol{A_1}), v(\boldsymbol{A_2}), \ldots, v(\boldsymbol{A_k})) = \overline{v}(\boldsymbol{\alpha})$ for every truth assignment v (this will exhaust all possible inputs into our defined function since $|\{T, F\}^k| = 2^k$). So, given any wff $\boldsymbol{\alpha}$ we may define a Boolean function whose output exactly matches that of $\overline{v}$ for every truth assignment v. So, the entire structure represented in our formal sentential language can be completely expressed with Boolean functions.

Example 3.6 *We readily see the following:*

$I^k_i = B^k_{\boldsymbol{A_i}}$

$N = B^1_{(\neg \boldsymbol{A_1})}$

$A = B^2_{(\boldsymbol{A_1} \wedge \boldsymbol{A_2})}$

$O = B^2_{(\boldsymbol{A_1} \vee \boldsymbol{A_2})}$

$C = B^2_{(\boldsymbol{A_2} \rightarrow \boldsymbol{A_2})}$

$E = B^2_{(\boldsymbol{A_2} \leftrightarrow \boldsymbol{A_2})}$

Even better than saying that each wff $\boldsymbol{\alpha}$ realizes a Boolean function $B^k_{\boldsymbol{\alpha}}$, we may say that our Boolean function is some composition of the 6 functions above (as long as $\boldsymbol{\alpha}$ is in the sentential language developed in Chapter 2). To see this, we recall our development of truth assignments in which we used the

Recursion Theorem back in Chapter 2. We used the following functions:

$$\mathcal{G}_{\neg}(V) = \begin{cases} T & \text{if } V = F \\ F & \text{if } V = T \end{cases}$$

$$\mathcal{G}_{\vee}(V_1, V_2) = \begin{cases} F & \text{if } V_1 = F \text{ and } V_2 = F \\ T & \text{otherwise} \end{cases}$$

$$\mathcal{G}_{\wedge}(V_1, V_2) = \begin{cases} T & \text{if } V_1 = T \text{ and } V_2 = T \\ F & \text{otherwise} \end{cases}$$

$$\mathcal{G}_{\rightarrow}(V_1, V_2) = \begin{cases} F & \text{if } V_1 = T \text{ and } V_2 = F \\ T & \text{otherwise} \end{cases}$$

$$\mathcal{G}_{\leftrightarrow}(V_1, V_2) = \begin{cases} T & \text{if } V_1 = V_2 \\ F & \text{otherwise} \end{cases}$$

These are precisely the Boolean functions N, A, O, C, and E. The Recursion Theorem told us the following:

$$\overline{v}(\boldsymbol{\alpha}) = \begin{cases} v(\boldsymbol{\alpha}) & \text{if } \boldsymbol{\alpha} = \boldsymbol{A_i} \text{ for some } i \\ \mathcal{G}_{\neg}(\overline{v}(\boldsymbol{\beta})) & \text{if } \boldsymbol{\alpha} = \mathcal{F}_{\neg}(\boldsymbol{\beta}) \\ \mathcal{G}_{\sharp}(\overline{v}(\boldsymbol{\beta}), \overline{v}(\boldsymbol{\gamma})) & \text{where } \boldsymbol{\alpha} = \mathcal{F}_{\sharp}(\boldsymbol{\beta}, \boldsymbol{\gamma}) \text{ and } \sharp \in \{\vee, \wedge, \leftarrow, \leftrightarrow\} \end{cases}$$

Since $B^k_{\boldsymbol{\alpha}}(I^k_1(\overrightarrow{V}), I^k_2(\overrightarrow{V}), \ldots, I^k_k(\overrightarrow{V})) = \overline{v}(\boldsymbol{\alpha})$ where $\overrightarrow{V} = (V_1, V_2, \ldots, V_k)$ and where $V_i = v(\boldsymbol{A_i})$ for the truth assignment v, and since $\mathcal{G}_{\sharp} = B$ where $B \in \{N, A, O, C, E\}$, we may rewrite the statement given to us by the Recursion Theorem as follows:

$$B^k_{\boldsymbol{\alpha}}(I^k_1(\overrightarrow{V}), I^k_2(\overrightarrow{V}), \ldots, I^k_k(\overrightarrow{V})) = \begin{cases} I^k_i(\overrightarrow{V}) & \text{if } \boldsymbol{\alpha} = \boldsymbol{A_i} \text{ for some } i \\ N(\overline{v}(\boldsymbol{\beta})) & \text{if } \boldsymbol{\alpha} = \mathcal{F}_{\neg}(\beta) \\ B(\overline{v}(\boldsymbol{\beta}), \overline{v}(\boldsymbol{\gamma})) & \text{where } \boldsymbol{\alpha} = \mathcal{F}_{\sharp}(\boldsymbol{\beta}, \boldsymbol{\gamma}) \text{ and} \\ & \sharp \in \{\wedge, \vee, \leftarrow, \leftrightarrow\} \text{ and} \\ & \text{where } B \in \{A, O, C, E\} \end{cases}$$

The Recursion Theorem applied here tells us how to write each boolean function determined by the wff $\boldsymbol{\alpha}$ as a composition of the five boolean functions $N,\ A,\ O,\ C,\ E$.

Example 3.7 $B^2_{((\neg \boldsymbol{A_1}) \leftrightarrow \boldsymbol{A_2})}(V_1, V_2) = E(N(I^2_1(V_1, V_2)), I^2_2(V_1, V_2))$

The Boolean functions have changed nothing. They just give a clearer perspective on the essential structure of the intended meaning of the logical connectives used in the sentential language. How do Boolean functions help us answer our primary question? It would make sense that wffs which realize the same Boolean function should be equivalent in every meaningful sense since they are able to express the same logical structure. We could easily create an equivalence relation such that wffs $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are equivalent if and only if $B^k_{\boldsymbol{\alpha}} = B^k_{\boldsymbol{\beta}}$. Now, a Boolean function is able to express any logical structure that we might dream up. If we are able to say that given a Boolean function, we can find a wff $\boldsymbol{\alpha}$ in some enriched language such that this $\boldsymbol{\alpha}$ realizes our Boolean function, and if we are able to say that for any wff $\boldsymbol{\beta}$ in any enriched language that there is a wff $\boldsymbol{\alpha}$ in our original language such that $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are equivalent ($B^k_{\boldsymbol{\alpha}} = B^k_{\boldsymbol{\beta}}$), then we would seem to have achieved our goal. However, we must first ask whether this notion of equivalence coincides with the notion of tautological equivalence, for that is how we originally stated our aim. We now prove that the two notions of equivalence in fact coincide with each other.

We put an ordering on $\{T, F\}$ by declaring that $F < T$ (this is completely natural if we think of F as 0 and T as 1). We first prove a lemma that will aid us in the next theorem.

Lemma 3.2.1

$$\{T, F\}^k = \{(v(\boldsymbol{A_{m+1}}), v(\boldsymbol{A_{m+2}}), \ldots, v(\boldsymbol{A_{m+k}})) : v \textit{ is a truth assignment}\}$$

for any $m \geq 0$ *where the* $\boldsymbol{A_{m+i}}$*'s are sentence symbols.*

Proof: Certainly the right set is a subset of the left since $v(\boldsymbol{A_i}) \in \{T, F\}$ for each $m+1 \leq i \leq m+k$. An element in $\{T, F\}^k$ takes the form $(V_1, V_2, \ldots, V_k)$ where $V_i \in \{T, F\}$. Let $v : \mathcal{S} \longrightarrow \{T, F\}$ be defined as follows:

$$v(A_i) = \begin{cases} V_{i-m} & \text{if } m+1 \leq i \leq m+k \\ T & \text{otherwise} \end{cases}$$

Then $(V_1, V_2, \ldots, V_k) = (v(\boldsymbol{A_{m+1}}), v(\boldsymbol{A_{m+2}}), \ldots, v(\boldsymbol{A_{m+k}}))$ and hence

$$(V_1, V_2, \ldots, V_k) \in \{(v(\boldsymbol{A_{m+1}}), \ldots, v(\boldsymbol{A_{m+k}})) : v \text{ is a truth assignment}\}.$$

Thus, the sets are the same. ∎

Theorem 3.8 *Let* $\boldsymbol{\alpha}$ *and* $\boldsymbol{\beta}$ *be wffs whose sentence symbols are among* $\boldsymbol{A_1}, \ldots, \boldsymbol{A_k}$. *Then*

(i) $\boldsymbol{\alpha} \models \boldsymbol{\beta}$ *if and only if for all* $\overrightarrow{V} \in \{T, F\}^k$, $B^k_{\boldsymbol{\alpha}}(\overrightarrow{V}) \leq B^k_{\boldsymbol{\beta}}(\overrightarrow{V})$ *(this follows the imposed ordering of* $T < F$.

(ii) $\boldsymbol{\alpha} \models\!\!\!\dashv \boldsymbol{\beta}$ *if and only if* $B^k_{\boldsymbol{\alpha}} = B^k_{\boldsymbol{\beta}}$.

(iii) $\models \boldsymbol{\alpha}$ *if and only if* $Im(B^k_{\boldsymbol{\alpha}}) = \{T\}$.

We can see that this theorem (part (ii)) provides the equivalence of our two notions of equivalence of expressibility of wffs.

Proof: (i) By definition $\boldsymbol{\alpha} \models \boldsymbol{\beta}$ if and only if for all truth assignments v, whenever $\overline{v}(\boldsymbol{\alpha}) = T$, then $\overline{v}(\boldsymbol{\beta}) = T$. Notice that $\overline{v}(\boldsymbol{\alpha}) \leq \overline{v}(\boldsymbol{\beta})$ if $\overline{v}(\boldsymbol{\alpha}) = F$ no matter the value of $\overline{v}(\boldsymbol{\beta})$, and we are guaranteed by the above statement that (assuming $\boldsymbol{\alpha} \models \boldsymbol{\beta}$) $\overline{v}(\boldsymbol{\alpha}) \leq \overline{v}(\boldsymbol{\beta})$ when $\overline{v}(\boldsymbol{\alpha})$ is true. On the other hand, if we know that $\overline{v}(\boldsymbol{\alpha}) \leq \overline{v}(\boldsymbol{\beta})$ for all truth assignments v, then we know that if $\overline{v}(\boldsymbol{\alpha}) = T$, that $\overline{v}(\boldsymbol{\beta}) \neq F$, that is, $\overline{v}(\boldsymbol{\beta}) = T$ otherwise the inequality would not hold. So $\boldsymbol{\alpha} \models \boldsymbol{\beta}$ if and only if $\overline{v}(\boldsymbol{\alpha}) \leq \overline{v}(\boldsymbol{\beta})$ for all truth assignments v. Using Lemma 3.2.1 and the fact that $B^k_{\boldsymbol{\tau}}(\overrightarrow{V}) = \overline{v}(\boldsymbol{\tau})$ where $\overrightarrow{V}$ corresponds to the truth assignment v at the end of the above lemma, we can say that this statement is equivalent to $B^k_{\boldsymbol{\alpha}}(\overrightarrow{V}) \leq B^k_{\boldsymbol{\beta}}(\overrightarrow{V})$ for all $\overrightarrow{V} \in \{T, F\}^k$.

(ii) By definition $\boldsymbol{\alpha} \models\!\!\!\!\dashv \boldsymbol{\beta}$ if and only if $\boldsymbol{\alpha} \models \boldsymbol{\beta}$ and $\boldsymbol{\beta} \models \boldsymbol{\alpha}$. Using part(i), we can say that that the former statement is true if and only if for all $\overrightarrow{V} \in \{T, F\}^k$, $B^k_{\boldsymbol{\alpha}}(\overrightarrow{V}) \leq B_{\boldsymbol{\beta}}(\overrightarrow{V})$ and $B^k_{\boldsymbol{\beta}}(\overrightarrow{V}) \leq B_{\boldsymbol{\alpha}}(\overrightarrow{V})$. This is of course equivalent to saying $B^k_{\boldsymbol{\alpha}} = B^k_{\boldsymbol{\beta}}$.

(iii) $\emptyset \models \boldsymbol{\alpha}$ if and only if for every truth assignment v, $\overline{v}(\boldsymbol{\alpha}) = T$. Using Lemma 3.2.1, this is true if and only if $B^k_{\boldsymbol{\alpha}}(\overrightarrow{V}) = T$ for all $\overrightarrow{V} \in \{T, F\}^k$ i.e. if $Im(B^k_{\boldsymbol{\alpha}}) = \{T\}$. ∎

Example 3.9 *Since* $(\boldsymbol{\nabla\alpha\beta\gamma}) \models\!\!\!\!\dashv (\neg(((\boldsymbol{\alpha} \wedge \boldsymbol{\beta}) \vee (\boldsymbol{\alpha} \wedge \boldsymbol{\gamma})) \vee (\boldsymbol{\beta} \wedge \boldsymbol{\gamma})))$.

$$B_{(\boldsymbol{\nabla\alpha\beta\gamma})} = B_{(\neg(((\boldsymbol{\alpha} \wedge \boldsymbol{\beta}) \vee (\boldsymbol{\alpha} \wedge \boldsymbol{\gamma})) \vee (\boldsymbol{\beta} \wedge \boldsymbol{\gamma})))}.$$

We can thus say that wffs $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are tautologically equivalent if and only if they realize equivalent Boolean functions. To attain our goal, it remains to be shown that any logical structure that can be captured by a Boolean function can be expressed by a wff in the sentential language that we developed in Chapter 2 i.e. we would like to show that each Boolean function is realizable by some wff in that language.

Theorem 3.10 *Let B^k be a k-place Boolean function where $k \geq 1$. Then there exists $\boldsymbol{\alpha}$, a wff in the sentential language which uses $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$ as logical connective symbols (with their normally intended meanings) such that $B^k = B^k_{\boldsymbol{\alpha}}$ (i.e. such that $\boldsymbol{\alpha}$ realizes B^k).*

Before proving the Theorem, we illustrate our proof with the following example. Suppose we are given the Boolean function B^3 defined by the following maps (we have suppressed commas and parentheses):

$$\begin{array}{ll} TTT \longmapsto T & TTF \longmapsto F \\ TFT \longmapsto T & FTT \longmapsto T \\ FFT \longmapsto T & FTF \longmapsto F \\ TFF \longmapsto F & FFF \longmapsto T \end{array}$$

Our goal is to find a wff $\boldsymbol{\alpha}$ such that $\boldsymbol{\alpha}$ realizes B^k. Note that B^3 returns the value T only for the inputs

$$\overrightarrow{V}_1 = (T, T, T)$$
$$\overrightarrow{V}_2 = (T, F, T)$$
$$\overrightarrow{V}_3 = (F, F, T)$$
$$\overrightarrow{V}_4 = (F, T, T)$$
$$\overrightarrow{V}_5 = (F, F, F)$$

First, we wish to design wffs that will be true precisely when the sentence symbols $\boldsymbol{A_1}, \boldsymbol{A_2}, \boldsymbol{A_3}$ take on the truth values of the input triples that cause the Boolean function to give an output of true. Using $\wedge$ and $\neg$ will serve our purpose as demonstrated below (note that we suppress some of the usage of parentheses for ease of reading using the fact that $\wedge$ and $\vee$ are associative when thought of as operations).

$\boldsymbol{\gamma_1} = \boldsymbol{A_1} \wedge \boldsymbol{A_2} \wedge \boldsymbol{A_3}$

$\boldsymbol{\gamma_2} = \boldsymbol{A_1} \wedge (\neg \boldsymbol{A_2}) \wedge \boldsymbol{A_3}$

$\boldsymbol{\gamma_3} = (\neg \boldsymbol{A_1}) \wedge (\neg \boldsymbol{A_2}) \wedge \boldsymbol{A_3}$

$\boldsymbol{\gamma_4} = (\neg \boldsymbol{A_1}) \wedge \boldsymbol{A_2} \wedge \boldsymbol{A_3}$

$\boldsymbol{\gamma_5} = (\neg \boldsymbol{A_1}) \wedge (\neg \boldsymbol{A_2}) \wedge (\neg \boldsymbol{A_3})$

So, $\boldsymbol{\gamma_i}$ is true for the corresponding unique element of $\{T, F\}^3$ and false for the remaining $8 - 1 = 7$ triples. Hence, if we let

$$\boldsymbol{\alpha} = \boldsymbol{\gamma_1} \vee \boldsymbol{\gamma_2} \vee \boldsymbol{\gamma_3} \vee \boldsymbol{\gamma_4} \vee \boldsymbol{\gamma_5},$$

it is clear that $\overline{v}(\boldsymbol{\alpha})$ will be true if and only if $\overline{v}(\boldsymbol{\gamma_i}) = T$ for a unique index i since the $\boldsymbol{\gamma_i}$'s are all distinct wffs and cannot be satisfied with the same truth assignment. So, $\overline{v}(\boldsymbol{\alpha}) = T$ for exactly five distict truth assignments v for the sentence symbols involved, and each distinct truth assignment will correspond to exactly one of the five triples for which B^3 returns an output of true. This means that $\overline{v}(\boldsymbol{\alpha})$ will return false exactly when B^3 returns false. So, we must have that $B^3_{\boldsymbol{\alpha}} = B^3$.

Proof: (Note: we again suppress some usage of parentheses for ease of reading, noting that $\vee$ and $\wedge$ are associate when considered as operations.)
<u>Case 1:</u> If $Im(B^k) = \{F\}$, we let $\boldsymbol{\alpha} = (\boldsymbol{A_1} \wedge (\neg \boldsymbol{A_1})) \wedge \boldsymbol{A_2} \wedge \cdots \wedge \boldsymbol{A_k}$. $\overline{v}(\boldsymbol{\alpha}) = F$ for every truth assignment v, so that $B^k_{\boldsymbol{\alpha}}(\overrightarrow{V}) = F = B^k(\overrightarrow{V})$ for all $\overrightarrow{V} \in \{T, F\}^k$ and hence $B^k = B^k_{\boldsymbol{\alpha}}$.
<u>Case 2:</u> If $Im(B^k) \neq \{F\}$, then there exists n and $\overrightarrow{V_i} = (V_{i1}, V_{i2}, \ldots, V_{ik})$ such

that for $1 \leq i \leq n$, $B^k(\overrightarrow{V_i}) = T$. Now we let

$$\boldsymbol{\beta_{ij}} = \begin{cases} \boldsymbol{A_j} & \text{if } V_{ij} = T \\ (\neg \boldsymbol{A_j}) & \text{if } V_{ij} = F \end{cases}$$

$$\boldsymbol{\gamma_i} = \boldsymbol{\beta_{i1}} \wedge \boldsymbol{\beta_{i2}} \wedge \cdots \wedge \boldsymbol{\beta_{ik}}$$

$$\boldsymbol{\alpha} = \boldsymbol{\gamma_1} \vee \boldsymbol{\gamma_2} \vee \cdots \vee \boldsymbol{\gamma_n}.$$

By construction, it is clear that $B^k_{\boldsymbol{\alpha}}(\overrightarrow{V}) = T$ if and only if $\overrightarrow{V} = \overrightarrow{V_i}$ for some $1 \leq i \leq n$ (refer back to the example above to see how this plays out). ∎

Note that the wff $\boldsymbol{\alpha}$ in the above theorem need not be unique; any tautological equivalent $\boldsymbol{\beta}$ to $\boldsymbol{\alpha}$ will also realize B^k by Theorem 3.8.

Example 3.11 *Suppose we enrich the Chapter 2 sentential language with* $\boldsymbol{\nabla}$ *as a logical connective symbol as discussed at the beginning of this section, and suppose we have the wff in the extended language*

$$\boldsymbol{\varphi} = ((\boldsymbol{\nabla A_1 A_2 A_3}) \wedge ((\neg \boldsymbol{A_2}) \rightarrow \boldsymbol{A_3})).$$

We have the following truth table for this wff.

$\boldsymbol{A_1}$	$\boldsymbol{A_2}$	$\boldsymbol{A_3}$	$(\neg \boldsymbol{A_2})$	$(\boldsymbol{\nabla A_1 A_2 A_3})$	$((\neg \boldsymbol{A_2}) \rightarrow \boldsymbol{A_3})$	$((\boldsymbol{\nabla A_1 A_2 A_3}) \wedge ((\neg \boldsymbol{A_2}) \rightarrow \boldsymbol{A_3}))$
T	*T*	*T*	*F*	*F*	*T*	*F*
T	*T*	*F*	*F*	*F*	*T*	*F*
T	*F*	*T*	*T*	*F*	*T*	*F*
T	*F*	*F*	*T*	*T*	*F*	*F*
F	*T*	*T*	*F*	*F*	*T*	*F*
F	*T*	*F*	*F*	*T*	*T*	*T*
F	*F*	*T*	*T*	*T*	*T*	*T*
F	*F*	*F*	*T*	*T*	*F*	*F*

$\boldsymbol{\varphi}$ *realizes the boolean function* $B^3_{\boldsymbol{\varphi}}$ *where*

$TTT \longmapsto F$

$TTF \longmapsto F$

$TFT \longmapsto F$

$TFF \longmapsto F$

$FTT \longmapsto F$

$FTF \longmapsto T$

$FFT \longmapsto T$

$FFF \longmapsto F$

Using the construction in Theorem 3.10 *we have*

$\boldsymbol{\gamma_1} = (\neg \boldsymbol{A_1}) \wedge \boldsymbol{A_2} \wedge (\neg \boldsymbol{A_3})$

$\boldsymbol{\gamma_2} = (\neg \boldsymbol{A_1}) \wedge (\neg \boldsymbol{A_2}) \wedge \boldsymbol{A_3}$

So, letting

$$\boldsymbol{\alpha} = \boldsymbol{\gamma_1} \vee \boldsymbol{\gamma_2} = (\neg \boldsymbol{A_1}) \wedge \boldsymbol{A_2} \wedge (\neg \boldsymbol{A_3}) \vee (\neg \boldsymbol{A_1}) \wedge (\neg \boldsymbol{A_2}) \wedge \boldsymbol{A_3}$$

$\boldsymbol{\alpha}$ *will realize* $B^3_{\boldsymbol{\alpha}} = B^3_{\boldsymbol{\varphi}}$*, or equivalently,* $\boldsymbol{\alpha} \models\!\!\!\!=\!\!\mid \boldsymbol{\varphi}$ *in the extended language.*

We have achieved our goal. For suppose that we enrich the sentential language developed in Chapter 2 with a (or many) new logical connective symbols. We have shown that any wff $\boldsymbol{\varphi}$ in this new language will realize a Boolean function $B^k_{\boldsymbol{\varphi}}$. But, by the theorem we have just proved $B^k_{\boldsymbol{\varphi}} = B^k_{\boldsymbol{\alpha}}$ where $\boldsymbol{\alpha}$, by construction, uses only the logical connectives $\wedge$, $\vee$, and $\neg$. By results we proved above, we know that it must then be the case that $\boldsymbol{\varphi}$ and $\boldsymbol{\alpha}$ are tautologically equivalent in our enriched language. Thus, our enriched language is able to express just as much logical structure as the old language. We give this property of expressing as much logical structure as we need a name.

Definition 3.12 *A set of logical connectives in a sentential language is* **complete** *if every k-place Boolean function for* $k \geq 1$ *is realizable by a wff which uses only the logical connectives contained within the set.*

Example 3.13 $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ *is complete as shown by Theorem* 3.10.

Example 3.14 *Any k-place Boolean function with $k \geq 1$ was shown to be realizable by a wff $\boldsymbol{\alpha}$ using only the symbols $\neg, \vee,$ and $\wedge$. Thus, $\{\neg, \wedge, \vee\}$ is complete.*

Example 3.15 $(\boldsymbol{A_1} \rightarrow \boldsymbol{A_2})$ *realizes the Boolean function*

$$\mathcal{G}_{\rightarrow}(V_1, V_2) = \begin{cases} F & \text{if } V_1 = T \text{ and } V_2 = F \\ T & \text{otherwise} \end{cases}.$$

In this case,

$TT \longmapsto T$

$FT \longmapsto T$

$FF \longmapsto T$

Our construction in the theorem thus shows that $(\boldsymbol{A_1} \rightarrow \boldsymbol{A_2})$ is tautologically equivalent to $(((\boldsymbol{A_1} \wedge \boldsymbol{A_2}) \vee ((\neg \boldsymbol{A_1}) \wedge \boldsymbol{A_2})) \vee ((\neg \boldsymbol{A_1}) \wedge (\neg \boldsymbol{A_2})))$. There may of course be shorter wffs to which $(\boldsymbol{A_1} \wedge \boldsymbol{A_2})$ is tautologically equivalent, but this is the one the theorem constructs for us. Similarly, $(\boldsymbol{A_1} \leftrightarrow \boldsymbol{A_2})$ is tautologically equivalent to $((\boldsymbol{A_1} \wedge \boldsymbol{A_2}) \vee ((\neg \boldsymbol{A_1}) \wedge (\neg \boldsymbol{A_2})))$.

Example 3.16 $\{\neg, \wedge\}$, $\{\neg, \vee\}$, *and* $\{\neg, \rightarrow\}$ *are each complete.*

We do not prove these propositions because they are not terribly enlightening. Essentially, we would use DeMorgan's laws ($(\boldsymbol{\alpha} \vee \boldsymbol{\beta}) \models\!\!\!\!\!\dashv (\neg((\neg \boldsymbol{\alpha}) \wedge (\neg \boldsymbol{\beta})))$ and $(\boldsymbol{\alpha} \wedge \boldsymbol{\beta}) \models\!\!\!\!\!\dashv (\neg((\neg \boldsymbol{\alpha}) \vee (\neg \boldsymbol{\beta})))$), the fact that $(\boldsymbol{\alpha} \rightarrow \boldsymbol{\beta}) \models\!\!\!\!\!\dashv ((\neg \boldsymbol{\alpha}) \vee \boldsymbol{\beta})$.

We have developed this idea of completeness at length, but to truly get a feel for what this completeness is, we will demonstrate a set that is not complete.

Theorem 3.17 *The set $\{\wedge, \rightarrow\}$ is not complete.*

Notice that if only one sentence symbol is involved, say $\boldsymbol{A_1}$, a truth assignment that assigns $\boldsymbol{A_1}$ a value of T will assign the wffs $\boldsymbol{(A_1 \wedge A_1)}$ and $\boldsymbol{(A_1 \rightarrow A_1)}$ the value of true. What is key is that neither one of these wffs can yield the value of false for an assignment of true for $\boldsymbol{A_1}$ which $\boldsymbol{(\neg A_1)}$ can.

Proof: Suppose $\{\wedge, \rightarrow\}$ is complete. Then every k-place Boolean function with $k \geq 1$ is realizable by a wff which uses the connectives in the complete set. This statement must also hold true for the Boolean function N where $N(T) = F$ and $N(F) = T$. So, there is a wff $\boldsymbol{\varphi}$ using only logical connectives from the set $\{\wedge, \rightarrow\}$ such that $B^1_{\boldsymbol{\varphi}} = N$ ($\boldsymbol{\varphi}$ only involves one sentence symbol).

Let $\boldsymbol{A}$ be an arbitrary sentence symbol. We wish now to establish the claim that $\boldsymbol{A} \vDash \boldsymbol{\alpha}$ for any wff $\boldsymbol{\alpha}$ that uses only the sentence symbol $\boldsymbol{A}$ in the sentential language under consideration. We will do this by an induction argument. Note: all wffs are wffs in the restricted language under consideration. Let $I = \{\boldsymbol{\alpha} : \text{If } \boldsymbol{\alpha} \text{ involves only the sentence symbol } \boldsymbol{A}, \text{ then } \boldsymbol{A} \vDash \boldsymbol{\alpha}\}$. The sentence symbols are a subset of I since it is trivially true to say $\boldsymbol{A} \vDash \boldsymbol{A}$, and for any other sentence symbol $\boldsymbol{B}$, since $\boldsymbol{B}$ does not involve the sentence symbol $\boldsymbol{A}$, it is a vacuously true statement so say "If $\boldsymbol{B}$ involves only the sentence symbol $\boldsymbol{A}$, then $\boldsymbol{A} \vDash \boldsymbol{B}$." Let $\boldsymbol{\alpha}, \boldsymbol{\beta} \in I$. So, $\boldsymbol{A} \vDash \boldsymbol{\alpha}$ and $\boldsymbol{A} \vDash \boldsymbol{\beta}$. Suppose v is a truth assignment such that $\overline{v}(\boldsymbol{A}) = v(\boldsymbol{A}) = T$. Since $\boldsymbol{A} \vDash \boldsymbol{\alpha}$ and $\boldsymbol{A} \vDash \boldsymbol{\beta}$, by definition, we know that $\overline{v}(\boldsymbol{\alpha}) = \overline{v}(\boldsymbol{\beta}) = T$. So, we must also have $\overline{v}(\boldsymbol{(\alpha \wedge \beta)}) = T = \overline{v}(\boldsymbol{(\alpha \rightarrow \beta)})$ by the Recursion Theorem. By definition then, $\boldsymbol{A} \vDash \boldsymbol{(\alpha \wedge \beta)}$ and $\boldsymbol{A} \vDash \boldsymbol{(\alpha \rightarrow \beta)}$. Thus, $\boldsymbol{(\alpha \wedge \beta)}, \boldsymbol{(\alpha \rightarrow \beta)} \in I$, and by the Induction Principle, the set I must be exactly the set of wffs. This statement holds for any sentence symbol $\boldsymbol{A}$ since the choice for the sentence $\boldsymbol{A}$ was arbitrary to begin with. Therefore, our claim is established.

Now, above we established that there is a wff $\boldsymbol{\varphi}$ in this restricted language such that $B^1_{\boldsymbol{\varphi}} = N$. Since $\boldsymbol{\varphi}$ involves only one sentence symbol, say $\boldsymbol{A_1}$, we know by what we just showed that $\boldsymbol{A_1} \vDash \boldsymbol{\varphi}$. By results established earlier, $B^1_{\boldsymbol{A_1}}(V) \leq B^1_{\boldsymbol{\varphi}}(V) = N$ for all $V \in \{T, F\}$. If $V = T$, then this statement says that $T \leq F$, a contradiction under our ordering of $\{T, F\}$. Thus, what we supposed initially is false, and $\{\wedge, \rightarrow\}$ is not complete. ∎

This particular theorem helps to show what completeness means in this context. A complete set of logical connectives will guarantee that we have sufficient structure in our sentential language to express any logical structure that we wish, a nice property to have in a model of humanity's deductive thought processes.

3.3 Effectiveness in the Sentential Language

Earlier, we established that if $\Sigma \vDash \boldsymbol{\tau}$ for the set $\Sigma \cup \{\boldsymbol{\tau}\}$ of wffs, then we only need finite $\Sigma_f \subseteq \Sigma$ to establish $\Sigma_f \vDash \boldsymbol{\tau}$. So, if a statement $\boldsymbol{\tau}$ is a consequence of the statements of Σ, even if Σ is infinite, we can find a finite number of statements that guarantee $\boldsymbol{\tau}$. This fact encourages us in our endeavor to find proofs (for which "$\Sigma_f \vDash \boldsymbol{\tau}$" serves as a model in the sentential language) for facts that are true without us yet knowing that they are true.

Yet, in the project of mathematics, we would like to know not just that we can prove true facts but also that certain statements *do not* follow from other statements. In other words, we would like not only to be able to demonstrate that "$\Sigma \vDash \boldsymbol{\tau}$" if this is a true statement, but would also like to be able to say "$\Sigma \vDash \boldsymbol{\tau}$" is a false statement when this is so. If we are guaranteed the ability to do the latter as well as the former, then with every statement and with every set of premises, we are guaranteed the ability of being able to *decide* whether our statement follows from the set of premises or not. That is, given a set of premises (perhaps a set of axioms) and given a sentential statement, we would be able to definitely determine whether the statement did or did not follow from the set of premises. We now discuss effective procedures and how they relate to our discussion of these ideas.

To remind the reader that the definitions, results, and questions here posed are of an informal character, we term our "definitions" as proclamations and our "theorems" as propositions.

Proclamation 3.17.1 *We will say a procedure is* **effective** *if it meets the following (informal) criteria:*

1. *There must be a finite list of exact instructions with each instruction being a finite string of symbols, explaining how to execute the procedure. These instructions should demand no cleverness on the part of the person (or machine) following them.*

2. *The procedure must avoid random devices (such as the flipping of a coin), or any such device which can, in practice, only be approximated.*

3. *In the case of a decision procedure, the procedure must be such that after a finite number of steps the procedure produces a "yes" or "no" answer.*

Intuitively, an effective procedure is one that can be carried out as a computer program. Note: our "definition" for an effective procedure is not mathematically rigorous, but questions of whether certain effective procedures exist are in fact central to the main result that this thesis wishes to discuss, namely, Gödel's Incompleteness theorem. We will discuss effective procedures here in an informal way, but will define an "effective procedure" in a mathematically rigorous way in a later chapter. In particular, it is easy enough to say that a procedure **is** effective given our informal definition above, but it is difficult to say when a procedure would **not** be effective with our informal definition. Later, when we rigorously develop these intuitive notions, we will be able to solve this problem.

Proclamation 3.17.2 *A set of expressions* Σ *(within the sentential language) is* **decidable** *if there is an effective procedure such that for each expression* $\boldsymbol{\varepsilon}$ *the procedure yields the statement "*$\boldsymbol{\varepsilon} \in \Sigma$*" or "*$\boldsymbol{\varepsilon} \notin \Sigma$*."*

Proposition 3.17.1 *The set of wffs,* $\mathcal{W}$*, in the sentential language is decideable.*

Proof: We outline the procedure (we assume the operator employing the procedure can recognize distinct expressions):

Step 1. If the expression is of the form $(\boldsymbol{\alpha}\sharp\boldsymbol{\beta})$ or $(\neg\boldsymbol{\alpha})$ where $\sharp \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ and where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are expressions, proceed to Step 2. Otherwise, stop and report that $\boldsymbol{\varepsilon} \notin \mathcal{W}$

Step 2. If $\boldsymbol{\varepsilon} = (\boldsymbol{\alpha}\sharp\boldsymbol{\beta})$ and $\boldsymbol{\alpha} = \boldsymbol{A_i}$ and $\boldsymbol{\beta} = \boldsymbol{A_j}$ for some i and j, stop and report that $\boldsymbol{\varepsilon} \in \mathcal{W}$. Otherwise, repeat Step 1 with $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. If $\boldsymbol{\varepsilon} = (\neg\boldsymbol{\alpha})$ and $\boldsymbol{\alpha} = A_i$ for some i, stop and report $\boldsymbol{\varepsilon} \in \mathcal{W}$. Otherwise, repeat Step 1 with $\boldsymbol{\alpha}$.

The procedure outlined clearly fulfills the three criteria for an effective procedure. The procedure instructions are finite, there is no guess work involved in the procedure outlined, and given the discussion in Chapter 2 of how wffs are constructed, the procedure outlined above must eventually cease and give a "yes, it is in the set of wffs" or "no, it is not in the set of wffs" response. ∎

The above fact, gives a flavor of what we mean by decidable set and also how we can show that a effective procedure exists for doing such-and-such a task. Essentially, we try to outline an algorithm that could be run by a computer which when the algorithm is run will accomplish the task we desire.

Proposition 3.17.2 *Every finite set of expressions is decidable.*

Proof: This statement is obvious. ∎

Proposition 3.17.3 *Not every set of expressions is decidable.*

Proof: There are countably many ($\aleph_0$) expressions as was shown earlier in the chapter, thus there are $2^{\aleph_0}$ (the cardinality of the power set of expressions) sets of expressions. On the other hand, since each effective procedure has a finite list of instructions which characterizes that procedure, and since we put no limit on how long that list of instructions is, the set of all effective procedures will have a cardinality of $\aleph_0$. Since $2^{\aleph_0} > \aleph_0$, we see that we have more sets of expressions than we have effective procedures that could be used to show that a particular set is decidable. ∎

Of course, the above results are to familiarize us with effective procedures and their limitations. Our interest is in results that will shed light on our search for mathematical deductions i.e. our search for mathematical knowledge. Our question is, given a set of wffs (premises), Σ, and another wff (statement), $\boldsymbol{\tau}$, can we always demonstrate that $\Sigma \models \boldsymbol{\tau}$ if this is true or demonstrate that $\Sigma \not\models \boldsymbol{\tau}$ if this is true.

At this point, we again take a moment to think about just exactly what we are doing. Sentential Logic is a mathematical model of humanity's deductive thought processes. Within this model. $\Sigma \models \boldsymbol{\tau}$ is our model for a deduction or proof. Thus, if we can answer the above question about our model, we can shed some meta-light on our endeavor as mathematicians: Given any set of premises, do we *know* we can always disprove or prove *any other statement* from our given set of premises? We now address this question.

Proposition 3.17.4 *Given any finite set $\Sigma \cup \{\boldsymbol{\tau}\}$ there is an effective procedure to decide whether or not $\Sigma \models \boldsymbol{\tau}$.*

Proof: Step 1. Create a truth table for all the wffs in $\Sigma \cup \{\boldsymbol{\tau}\}$. [Since $\Sigma \cup \{\boldsymbol{\tau}\}$ is finite, this step is doable in finitely many operations.]

Step 2. Find all rows of the truth table which correspond to truth assignments which satisfy all wffs in the set Σ. If there are none, return "$\Sigma \models \boldsymbol{\tau}$" (this corresponds to our second main case for tautological implications described in Chapter 2). [Since there are finitely many cells in our truth table created in Step 1 this step is doable in finitely many operations.]

Step 3. For each of the rows in Step 2, look at the cell corresponding to $\boldsymbol{\tau}$. If for any row, $\boldsymbol{\tau}$ has a value of F, return $\Sigma \not\models \boldsymbol{\tau}$. Otherwise, if for every row $\boldsymbol{\tau}$ has a value of T, return $\Sigma \models \boldsymbol{\tau}$. [This step is doable in finitely many operations.]

There is no guesswork or cleverness in the steps outlined above, and since the instructions are finite and the process produces a "yes/no" response, the procedure outlined is effective. ∎

Corollary 3.17.1 *For a finite set of wffs Σ, the set of tautological consequences, $\{\boldsymbol{\tau} \in \mathcal{W} | \Sigma \models \boldsymbol{\tau}\}$, is decidable.*

The above tells us that given finitely many premises, and any other statement, we can definitively prove whether or not that statement is a consequence of the premises. In particular, in axiomatic mathematics, we begin with a set of statements that we assume to be true, and the theory of those axioms (e.g. group theory, field theory, Zermelo-Fraenkel set theory, etc.) is all statements that we can prove assuming the truth of those axioms, including statements that say that certain things *do not follow* from the axioms. If we have finitely many axioms, then our results about the sentential logic model indicate that we can completely decide which statements follow from our axioms and which do not. *There are no undecidable statements in a theory with finitely many axioms.* However, many of our most beloved and familiar axiomatic systems implicitly involve *infinitely many axioms.*

For example, one of the axioms of Peano arithmetic states: "If $n \in \mathbb{N}$, then $S(n) \in \mathbb{N}$" where S is the successor function. As seasoned mathematicians, our concept of this statement is that it asserts one thing, one of the foundational properties that defines the set $\mathbb{N}$. Now, we could translated this statement into our sentential language as the sentence symbol, say, $\boldsymbol{A_1}$. The problem is that if we wish to express the structure of Peano arithmetic with our sentential language, then such a translation is unhelpful. This unhelpfulness comes from the fact that our sentence symbols are our most basic propositions with no inherent meaning behind the symbol. We never look at the meaning of what the sentence symbol is intended to translate. The power in the above axiom of Peano arithmetic comes in the "if/then" structure. By translating the whole statement as one sentence symbol, we have lost the deducing power of the axiom.

We can go to the other extreme and intend for $\boldsymbol{A_0}$ to translate "$0 \in \mathbb{N}$, $\boldsymbol{A_1}$ to translate "$1 \in \mathbb{N}$, etc. So, $\boldsymbol{(A_0 \rightarrow A_1)}$ could be used to support the statement "If $0 \in \mathbb{N}$, then $S(0) \in \mathbb{N}$," and similarly for all other natural numbers. Thus, if we wish to support the structure of Peano arithmetic using sentential logic, we could include the infinitely many wffs $\boldsymbol{(A_i \rightarrow A_{i+1})}$ to support the statement "The successor of every natural number is a natural number."

Thus, if we are attempting to represent the proof of a statement that follows from the axioms of Peano arithmetic in the sentential language, we would do so by stating that "$\Sigma \models \boldsymbol{\tau}$" where Σ is the set of all Peano axioms translated into the sentential language, including the wffs $\boldsymbol{(A_i \rightarrow A_{i+1})}$ for each $i \in \mathbb{N}$, and where $\boldsymbol{\tau}$ is the statement we are proving, i.e, a statement of the theory of Peano arithmetic. This goes to show that as far as Sentential Logic is concerned we may wish to include the possibility of infinitely many axioms for the theory under consideration. Thus, if our desire is, given the axioms which determine a theory, to be able to decide exactly which statements follow from those axioms and which do not, it is insufficient to say that $\{\boldsymbol{\tau} \in \mathcal{W} | \Sigma \models \boldsymbol{\tau}\}$ is decidable for Σ finite. We should also consider cases where Σ is infinite as with the axioms of Peano arithmetic.

We already know from the discussion above that some infinite sets *must* be undecidable. In fact, Gödel's Incompleteness Theorem will show us that, in general, the set of tautological consequences of an infinite set Σ (for example the axioms of Peano arithmetic) will be undecidable. So, our goal of decidability for every particular theory in mathematics is unattainable. However, "half" of decidability is attainable in the following sense.

Proclamation 3.17.3 *We will say that a set of expressions Σ within the sentential language is effectively enumerable if and only if there is an effective procedure that will list in order the elements of Σ.*

If Σ is infinite, then the process which lists the elements of Σ will never finish, but for any element in Σ, the process must eventually (given a sufficient finite amount of time) output the element as an element from Σ.

We present the next proposition in order to discuss the difference between effective enumerability and decidability.

Proposition 3.17.5 *A set Σ of expressions is effectively enumerable if and only if there is an effective procedure which, given any expression $\boldsymbol{\varepsilon}$ produces the answer "Yes" if and only if $\boldsymbol{\varepsilon} \in \Sigma$.*

The last piece of our proposition bears some remark before proceeding to the proof. Saying, "Yes" if and only if $\boldsymbol{\varepsilon} \in \Sigma$ *does not mean* that the procedure will answer "No" if $\boldsymbol{\varepsilon} \notin \Sigma$. It *may*, or the process may give no answer but enter an infinite loop instead. The only stipulation is that the procedure cannot answer "Yes" if $\boldsymbol{\varepsilon} \notin \Sigma$.

Proof: If we are given the expression $\boldsymbol{\varepsilon}$, we will use our given effective procedure for listing the elements of Σ. Our procedure will give a return of "Yes" if $\boldsymbol{\varepsilon}$ is encountered. If $\boldsymbol{\varepsilon}$ is not in Σ, our procedure will continue to list elements of Σ forever (we are of course assuming that Σ is an infinite set) never answering "Yes."

If now we are given the effective procedure that will answer "Yes" if and only if expression $\boldsymbol{\varepsilon} \in \Sigma$, we design the following procedure to list the elements of Σ. Note that our effective procedure must give an answer of "Yes" when fed $\boldsymbol{\varepsilon}$, but may take a time of k minutes or less to report this answer. So, in the following procedure we re-test expressions for greater and greater amounts of time to ensure that they end up in our listing.

Step 1. Examine all 1-tuples which use symbols from the following subset of the Sentential alphabet: $\{(,), \neg, \vee, \wedge, \rightarrow, \leftrightarrow, \boldsymbol{A_1}\}$. Test each 1-tuple expression under the effective procedure for 1 minute. If the procedure yields an answer of "Yes", add that expression to the list of Σ.[Since there are only finitely many 1-tuples using these symbols, this will take a finite amount of time.]

Step 2. Examine all 1-tuples and 2-tuples which use symbols from the following subset of the Sentential alphabet: $\{(,), \neg, \vee, \wedge, \rightarrow, \leftrightarrow, \boldsymbol{A_1}, \boldsymbol{A_2}\}$. Test each 1-tuple expression and 2 tuple expression under the effective procedure for 2 minutes. If the procedure yields an answer of "Yes", add that expression to the list of Σ.[Since there are only finitely many 1-tuples and 2-tuples using these symbols, this will take a finite amount of time.]

⋮

Step i. Examine all 1-tuples, 2-tuples,..., and i-tuples which use symbols from the following subset of the Sentential alphabet:

$$\{(,), \neg, \vee, \wedge, \rightarrow, \leftrightarrow, \boldsymbol{A_1}, \boldsymbol{A_2}, \ldots, \boldsymbol{A_i}\}.$$

Test each 1-tuple expression, 2-tuple expression,..., and i-tuple expression under the effective procedure for i-minutes. If the procedure yields an answer of "Yes", add that expression to the list of Σ.[Since there are only finitely many 1-tuples, 2-tuples,..., and i-tuples using these symbols, this will take a finite amount of time.]

This procedure consists of finite instructions which could easily be carried out by a computer program, and is thus effective. Given any expression $\boldsymbol{\varepsilon}$ in Σ, this expression is some n-tuple and thus involves only finitely many symbols from our sentential alphabet. It will come under examination at all steps subsequent to and including Step n. Our effective procedure must give an answer of "Yes" when fed $\boldsymbol{\varepsilon}$, but may take a time of k minutes or less to report this answer. So, this expression will be added to the list of elements of Σ for some (and possibly all) steps from Step n to Step k (if $k \geq n$). Also, each step will terminate in finite time since we only have the procedure checking each expression finitely long (although that time for each expression increases as the step number increases). Thus, we avoid an infinite loop if the effective procedure given to us enters an infinite loop when $\boldsymbol{\varepsilon} \notin \Sigma$. Thus, the effective procedure given above will list each element of Σ (in fact each element of Σ will be listed infinitely many times, but the algorithm could be adjusted to fix this fact if it is deemed to be undesirable), and therefore, Σ is effectively enumerable. The proposition holds. ∎

This proposition highlights the difference between decidability and effective enumerability. With decidability, our effective procedure reports "Yes" when the expression $\boldsymbol{\varepsilon} \in \Sigma$ and also "No" when $\boldsymbol{\varepsilon} \notin \Sigma$. We are able to decide exactly what is and what is not in Σ. But with effective enumerability, our procedure only definitely reports "Yes" if $\boldsymbol{\varepsilon} \in \Sigma$, but not necessarily "No" if $\boldsymbol{\varepsilon} \notin \Sigma$. With an effectively enumerable set, and a particular $\boldsymbol{\varepsilon}$ under consideration, if our effective procedure has taken 1,000,000 years and has not given us an answer of "Yes", we still cannot say that $\boldsymbol{\varepsilon}$ is not in our set, for perhaps in the next iteration the procedure will answer "Yes." So, in this sense effective enumerability is half of decidability. It is in this sense that our goal is "half" attainable. We use our result to address our question of provability within the sentential model.

Proclamation 3.17.4 *If Σ is a decidable set of wffs, the set of tautological consequences of Σ is effectively enumerable.*

Proof: We have an effective procedure to list all of the elements of Σ as $\boldsymbol{\sigma_1}, \boldsymbol{\sigma_2}, \boldsymbol{\sigma_3}, \ldots$ Let $\boldsymbol{\tau}$ be an arbitrarily given wff.

Step 1. Confirm or deny $\emptyset \vDash \boldsymbol{\tau}$ using an appropriate truth table. [This is an effective procedure since $\emptyset$ is a finite set.] If the statement is true report "Yes."

Step 2. Confirm or deny $\boldsymbol{\sigma_1} \vDash \boldsymbol{\tau}$ using an appropriate truth table. If the statement is true, report "Yes."

⋮

Step i. Confirm or deny $\{\boldsymbol{\sigma_1}, \boldsymbol{\sigma_2}, \ldots, \boldsymbol{\sigma_i}\} \vDash \boldsymbol{\tau}$ using an appropriate truth table. If the statement is true, report "Yes."[This is an effective procedure since $\{\boldsymbol{\sigma_1}, \boldsymbol{\sigma_2}, \ldots, \boldsymbol{\sigma_i}\}$ is a finite set.]

Eventually this procedure, which is effective, containing finite instructions and involving no guesswork, must report "Yes" if and only if $\boldsymbol{\tau}$ is a tautological consequence of Σ by the corollary to the Compactness Theorem. For if $\Sigma \vDash \boldsymbol{\tau}$, then there exists finite $\Sigma_f \subseteq \Sigma$ such that $\Sigma_f \vDash \boldsymbol{\tau}$, and all members of Σ_f must eventually be represented in one of the steps above. By Proposition 3.17.5, the set of tautological consequences of Σ must be effectively enumerable. ∎

Such a result encourages our endeavor in mathematics to find new results for the above proposition indicates that all true facts following from a set of premises must be effectively enumerable. So, what this result indicates is that if a proof is to be found from a set of assumptions, it can be found. If a statement does not follow from our premises or axioms, then we may be able to show that it does not follow or we may not.

For instance, Goldbach's Conjecture in number theory states that every even integer greater than 2 is the sum of two primes. This conjecture, although it has great concrete number evidence to support it, remains unproven. Suppose Σ is the set of all statements true in number theory. Now, if Goldbach's Conjecture is a true statement of number theory, then our results above indicate that there must be a deduction which shows it to be true i.e. we can say that $\Sigma_f \vDash \boldsymbol{\tau}$ where Σ_f is a finite subset of Σ and $\boldsymbol{\tau}$ is Goldbach's Conjecture. We can think of Σ_f as a set of axioms we might be assuming. As long as we are assuming the right axioms Σ_f, then we ought to be answer "yes" to Goldbach's Conjecture being a consequence of Σ_f if Goldbach's Conjecture is true of number theory. However, if Goldbach's Conjecture is not provable from a given Σ_f, we may not be able to say so i.e. we may not be able to answer "No" to its being a consequence of our current set of axioms that we are operating under. We have no idea of whether we should stop searching for a proof for Goldbach's Conjecture given our current set of axioms. It could be (as with another famous number theoretic result that was unproven for hundreds of years, Fermat's Last Theorem) that a couple more years of research will yield exactly the sort of proof that we seek. So, we may soldier on with the mathematical project encouraged to think that we can prove all true mathematical results that are true given our current set of axioms. We just may be unable to know if some statement is not a consequence of our current set of axioms.

3.4 Shortcomings of Sentential Logic

Having examined some of the nice results and properties that the sentential model of logic has and that we would like to see for a model of humanity's deductive thought processes, we now consider where our model falls short.

A severe limitation of our model was hinted at above when we discussed one of the axioms of Peano arithmetic: "If $n \in \mathbb{N}$, then $S(n) \in \mathbb{N}$." We saw that we could translate this axiom into infinitely many statements in the sentential language. Such a translation is awkward to work with however. On the other hand, we saw that if we attempt to translate the statement as a single sentence symbol, we lose the deductive power that the axiom is intended to express. Of course, we could rephrase the axiom as, “For all $n \in \mathbb{N}$, $S(n) \in \mathbb{N}$”. The idea of the axiom is that if we *range over the set* that is the natural numbers, the successor of any one number will also be a natural number. It is the inability to express this idea of ranging over a set that leads us to our dilemma in expressing the Peano axiom in the sentential language and is a significant shortfall in our language. We do not have sufficient structure in our language to compactly describe what we feel to be an intuitive notion. The sentence symbols used as our “core” or “atomic” symbols of expressing statements are clunky when we attempt to express connections between them when those connections consist of properties of sets. Our sentential model has a significant shortcoming. However, Sentential Logic is just that: a mathematical model for humanity’s deductive thought processes. With it we have proved some interesting and useful results that indicate limits to our deductive endeavors. It has served as an excellent foundation and prototype for our next and more refined model, First-Order Logic.

Chapter 4

First-Order Languages and Structures

In the last two chapters, we developed the structure of sentential logic and examined some of its nice properties as a model for human deduction. We also saw its limitations when we attempted to describe properties of sets. In this chapter, we begin the development of a new and richer model of logic for which sentential logic will serve as the prototype–first-order logic.

Since we are attempting to present a mathematical model of deduction, we will develop first-order logic in a rigorously mathematical fashion. However, many of the results and processes developed in sentential logic carry over to the development of first-order logic, so some discussion of similar results will be suppressed in the first-order setting.

Since Gödel's Incompleteness Theorem is stated in terms of first-order logical statements, we are moving one step closer to the main result of this thesis. In this chapter, we first develop first-order languages and then discuss what truth and falsity mean within our new system.

4.1 First-Order Languages

For a first-order language, we assume that we have an infinite, but countable, alphabet in which we have two types of symbols: logical symbols and parameters. We catagorize these in the chart below.

Symbol	**Type**
(	Logical Grouping
)	Logical Grouping
$\neg$	Logical Connective
$\rightarrow$	Logical Connective
$\boldsymbol{v_n}$ for every $n \in \mathbb{N}$	Logical Variable Symbols
$\approx$	Logical Equality (optional)
$\forall$	Parameter; Quantifier
$\boldsymbol{P_i^n}$ for at least one n and one i in $\mathbb{Z}^+$	n-place Predicate Parameter(s)
$\boldsymbol{a_n}$ for $n \in \mathbb{Z}^+$	Constant Parameter(s) (optional)
$\boldsymbol{f_i^n}$ for $n, i \in \mathbb{Z}^+$	n-place Function Parameter(s) (optional)

Note: even though we have termed the equality symbol as a logical symbol, we will also think of it as a 2-place predicate symbol.

To specify a particular language, we say whether we have equality and what parameters are present. Our language will have an intended interpretation, but our language by itself is just a formal collection of symbols without any inherent meaning. The idea behind a first-order language is that it can *support* a particular mathematical structure.

Why we are including and catagorizing the specific symbols that we have in the table bears some comment. The parentheses, "$\neg$," and "$\rightarrow$" symbols are familiar from sentential logic. We do not include any other logical connectives because the set $\{\neg, \rightarrow\}$ is a complete set as mentioned in the last chapter.

We saw that the major downfall of sentential logic was its inability to express properties which range across whole sets. This inability came from the clunkiness of the sentence symbols. To remedy this in first-order logic we essentially decompose the sentence symbols to be able to express a richer structure. To this end we introduce a quantifier, variables, constants, predicate symbols and function symbols.

Variable symbols will be stand-ins for the elements in the set that our particular language fundamentally deals with. Constants will also come from this set, but including function symbols allows us to minimize how many constants we use in our language.

For instance, if we are working with the language for number theory, we will specify the constant $\mathbf{0}$ as being in our language. Of course, the intended translation of $\mathbf{0}$ is 0. To avoid further constant symbols, we will use the 1-place function symbol $\boldsymbol{S}$ (where the intended translation is the successor function), and write $\boldsymbol{S}(\mathbf{0})$ to translate for 1.

The quantifier $\forall$ is intended to translate "for every member" in the set that our language fundamentally deals with. This symbol will essentially encode the idea of ranging over a whole set. We will also want to talk about the existence of certain elements in a set. For instance, we might want to say, "There is a least natural number." We can actually encode this idea using $\neg\forall\neg$. "There is a least natural number" is equivalent to "It is not the case that for every natural number, the natural number is greater than some other natural number." In some of our discussion and examples, we replace the $\neg\forall\neg$

pattern with $\exists$ as an abbreviation for "there is a member."

The n-place predicate symbols together with the variables in our language essentially take the place of our sentence symbols in sentential logic. The intended translation of each predicate symbol is some property that elements of the foundational set may or may not have. For instance in the language of number theory the 2-place predicate symbol $\boldsymbol{<}$ is intended to translate the less than property. Thus, $\boldsymbol{S(0) < S(S(0))}$ will translate the proposition "One is less than two." For a unary predicate symbol $\boldsymbol{P}$, $\exists \boldsymbol{xPx}$ could be used to support the meaning that there is a member of a set that has property P (or equivalently, we could write $\boldsymbol{(\neg \forall x(\neg Px))}$).

Since predicate symbols give the translation of properties, the requirement that each particular language should have at least one predicate symbol makes sense. Without a predicate symbol we cannot translate any properties and hence no propositions can be translated since each proposition will include one or more properties that either are or are not fulfilled, determining the truth value of the proposition.

All of the features of our expanded first-order system interplay to give us a wider range of expressiveness.

Since we have mentioned the Peano Axioms several times now and will use them in a few more examples, we list them for the reader's reference (this list is adapted from page 1 of [5]).

Peano Axioms of Arithmetic:

N1 $0 \in \mathbb{N}$.

N2 If $n \in \mathbb{N}$, then its successor is in $\mathbb{N}$.

N3 0 is not the successor of any element of $\mathbb{N}$.

N4 If n and m in $\mathbb{N}$ have the same successor, then $n = m$.

N5 A subset of $\mathbb{N}$ which contains 0, and which contains $n + 1$ whenever contains n, must equal $\mathbb{N}$.

Example 4.1 ***Language of Number Theory****:*

Equality: Yes

Predicate Symbols: One, 2-place predicate symbol $<$

Constant Symbols: $\mathbf{0}$

1-place Function Symbols: $\boldsymbol{S}$ *(intended to translate the successor function*

2-place Function Symbols: $+$ *(for addition),* $\cdot$ *(for multiplication), and* $\boldsymbol{E}$ *(for exponentiation).*

Note in the above list that all of these parameters are just symbols. There is no inherent meaning behind them, but we can think of them as being sufficient to support a particular meaning. We will make this idea more rigorous in the discussion to come.

We can use the formal expression

$$\forall \boldsymbol{v_1}(\neg \boldsymbol{S}\boldsymbol{v_1} \approx \mathbf{0})$$

to support the logical structure of the Peano Axiom, "0 is not the successor of any number."

Example 4.2 *Using the same language for number theory, we can translate the Peano Axiom "The successor of each natural number is itself a natural number," with ease into the first-order language for number theory*

$$\forall \boldsymbol{v_1}(\exists \boldsymbol{v_2}(\boldsymbol{S}(\boldsymbol{v_1}) = \boldsymbol{v_2})).$$

In English, "For each natural number, there is another natural number equal to the successor of the first natural number." In sentential logic, to translate *this same axiom, we declared that* $\boldsymbol{A_i}$ *was intended to translate "i* $\in \mathbb{N}$*," so that* $\boldsymbol{A_0} \sim$ "$0 \in \mathbb{N}$", $\boldsymbol{A_1} \sim$ "$1 \in \mathbb{N}$", *etc. Similarly, we declared that* $\boldsymbol{A_{i+1}}$ *was intended to translate* $S(i) \in \mathbb{N}$. *So,* $\boldsymbol{A_1} \sim$ "$S(0) \in \mathbb{N}$", $\boldsymbol{A_2} \sim$ "$S(1) \in \mathbb{N}$", *etc. Thus, we expressed our axiom of Peano arithmetic as the wffs* $\boldsymbol{(A_i \to A_{i+1})}$ *for each* $i \in \mathbb{N}$. *So,* $\boldsymbol{(A_0 \to A_1)} \sim$ "$0 \in \mathbb{N}$ *implies* $S(0) \in \mathbb{N}$".

Example 4.2 demonstrates the strength of first-order logic in expressing properties that hold for entire sets. Despite the enhanced power of expressiveness in our first-order languages, we will not, in general, be able to express every proposition about a particular theory in the first-order language for that theory.

Example 4.3 *Using the language of number theory, we cannot express the proposition, "Every non-empty set of natural numbers has a least element." The reason for this is that this statement describes a property which ranges over sets. In our language for number theory, our quantifier* $\forall$ *ranges over the set of natural numbers, not the power set of the natural numbers. When we have the ability to range over members of a set and also over members of the power set of that set, we are working with a different mathematical model of logic called "second-order logic."*

Another example of a second-order logical statement is the Completeness Axiom of the real numbers: "Every bounded above set has a least upper bound." Notice that we are ranging over sets, not elements of sets.

We could create a language for which our fundamental *class* is all sets (the "set" of all sets is too large to be a set and hence we have to think of this object as a class, but we may think of our quantifier $\forall$ as ranging over a class instead of a set) as follows.

Example 4.4 ***Language of Set Theory***

Equality: Yes

Predicate Parameters: One 2-place predicate symbol $\in$ *(intended translation: "is a member of")*

Constant Symbol: $\emptyset$ *(intended translation: the set that contains no elements)*

Function Symbols: None

With this language, we can write such statements of set theory as

$$(\neg\exists \boldsymbol{v_1} \forall \boldsymbol{v_2} \boldsymbol{v_2} \in \boldsymbol{v_1})$$

In English, this says that there is no set of which every set is a member.

In this language for set theory (we may need to add some more predicate symbols) we could express our proposition that every set of natural numbers has a least element, but we cannot express this statement *about* number theory in the first-order language of number theory.

We give another example of a first-order language for a familiar mathematical structure to give the reader a little more intuitive feel for first-order languages before defining them rigorously.

Example 4.5 ***First-Order Language for an Arbitrary Group (Language for Group Theory)***

Equality: Yes

Predicate Parameters: One 1-place predicate symbol $\in$ *(intended translation: "is a member of the group")*

Constant Symbols: $\boldsymbol{e}$ *(intended translation: the identity element of the group)*

Function Symbols: One 2-place function symbole $*$ *(intended translation: the* *binary group operation)*

We can translate the group axioms into our language (we assume that our group is non-empty).

$$\forall \boldsymbol{v_1} \forall \boldsymbol{v_2}((\in \boldsymbol{v_1} \wedge \in \boldsymbol{v_2}) \rightarrow \in \boldsymbol{v_1} * \boldsymbol{v_2})$$

$$\forall \boldsymbol{v_1} \forall \boldsymbol{v_2} \forall \boldsymbol{v_3}(\boldsymbol{v_1} * (\boldsymbol{v_2} * \boldsymbol{v_3}) \approx (\boldsymbol{v_1} * \boldsymbol{v_2}) * \boldsymbol{v_3})$$

$$\forall \boldsymbol{v_1}(\boldsymbol{v_1} * \boldsymbol{e} \approx \boldsymbol{v_1} \wedge \boldsymbol{e} * \boldsymbol{v_1} \approx \boldsymbol{v_1})$$

$$\forall \boldsymbol{v_1} \exists \boldsymbol{v_2}(\boldsymbol{v_1} * \boldsymbol{v_2} \approx \boldsymbol{e} \wedge \boldsymbol{v_2} * \boldsymbol{v_1} * \boldsymbol{e})$$

These statements' English equivalents are as follows:

"If v_1 and v_2 are members of the group, then $v_1 * v_2$ is a member of the group" (the group is closed under the binary operation).

"For all elements v_1, v_2, and v_3 in the group, $v_1 * (v_2 * v_3) = (v_1 * v_2) * v_3$" (the binary operation is associative).

"For every element v_1 in the group, $v_1 * e = v_1 = e * v_2$" (the group has an identity element with respect to the group operation).

"For every element in the group v_1, there is an element v_2 such that

$v_1 * v_2 = e = v_2 * v_1$" (every element in the group has an inverse).

4.2 Rigorous Development of First-Order Languages

In our discussion of first order languages above, we were playing fast and loose with the symbols at our disposal to give the reader some initial feel for the differences between first-order languages and the language for sentential logic. We now develop things more rigorously.

First, an expression in a first-order language will be a finite sequence of symbols from the first-order language alphabet. For each first order language and for each n-place function symbol $\boldsymbol{f}$, we will define a formula-building operation F_f whose domain is the set of all variables and constant symbols in the language, and where $\mathcal{F}_f(\boldsymbol{\varepsilon_1}, \boldsymbol{\varepsilon_2}, \ldots, \boldsymbol{\varepsilon_n}) = \boldsymbol{f\varepsilon_1\varepsilon_2 \cdots \varepsilon_n}$.

Definition 4.6 *The set of* **terms** *in a first-order language is the set generated (in the sense of generation discussed in Chapter 2) by the class of formula building operations* $\mathfrak{F} = \{\mathcal{F}_f : \boldsymbol{f}$ *is a* $n-$*place function symbol in the language*$\}$ *applied to the variables and constant symbols of the language.*

Example 4.7 *In the language of number-theory discussed above,* $\mathbf{0}$*,* $\boldsymbol{S0}$*, and* $\boldsymbol{+v_2S0}$ *are terms in the language.*

Note that the intended meaning of $\boldsymbol{S0}$ is precisely the same as that for $\boldsymbol{S(0)}$, and similarly for $\boldsymbol{+v_2S0}$ and $\boldsymbol{v_2 + S(0)}$. For the purposes of establishing rigorous results for first-order languages, we have defined terms without parentheses in a non-intuitive way (so that $\boldsymbol{+v_2S0}$ means the same thing as $\boldsymbol{v_2 + S(0)}$). This way of representing mathematical expressions is actually used in practice as an alternative mode of entry into calculators. It is often referred to as "Polish notation."

Example 4.8 *In the language of set theory discussed above, only the variables and the constant symbol* $\boldsymbol{\emptyset}$ *are terms since there are no* n*-place function symbols.*

In the language for an arbitrary group, $\boldsymbol{*v_1v_2}$ *and* $\boldsymbol{*e * v_1v_2}$ *are terms.*

Notice that terms assert nothing whatsoever but are merely the objects about which assertions can be made. In this sense, the terms are the nouns and pronouns for our particular first-order language.

Definition 4.9 *An* **atomic formula** *is an expression of the form* $\boldsymbol{P t_1 t_2 \cdots t_n}$ *where* $\boldsymbol{P}$ *is an n-place predicate symbol and where* $\boldsymbol{t_i}$ *for* $1 \leq i \leq n$ *is a term.*

Note that terms are not atomic formulas, but atomic formulas are built from a finite sequence of terms and a predicate symbol. The name "atomic formulas" is suggestive of their function. Just as the sentence symbols were the core wffs in the sentential language, the atomic formulas are the core wffs in each sentential language. Each n-place predicate symbol is intended to translate a property that the n terms have or do not have. Thus, $\boldsymbol{P t_1 t_2 \cdots t_n}$ asserts that the n terms have property $\boldsymbol{P}$.

Example 4.10 $\boldsymbol{< 0S0}$ *translates the assertion that 0 is less than 1 in the language for number theory.*

$\boldsymbol{\approx v_1 v_2}$ *translates the assertion that one variable is equal to another in any first-order language under consideration. This expression is an atomic formula since we consider* $\boldsymbol{\approx}$ *to be a 2-place predicate symbol as well as a logical symbol.*

In the language for an arbitrary group, $\boldsymbol{\in * v_1 v_2}$ *is an atomic formula. It asserts that the group operation between* v_1 *and* v_2 *is a member of the group.*

Having defined the atomic formulas and made the correlation between these expressions and the sentence symbols in sentential logic, we are now in a position to define the wffs for any particular first-order language.

First we define the formula-building operations on the set of expressions for a first order language as follows:

$$\mathcal{F}_{\neg}(\boldsymbol{\alpha}) = \boldsymbol{(\neg\alpha)}$$

$$\mathcal{F}_{\rightarrow}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{(\alpha \rightarrow \beta)}$$

$$\mathcal{F}_{\forall,i}(\boldsymbol{\alpha}) = \forall \boldsymbol{v_i \alpha} \text{ for } i = 1, 2, \ldots$$

The first two formula building operations are familiar from sentential logic. The third is new with the introduction of a quantifier symbol into first-order languages.

Definition 4.11 *The set of* **wffs** *for a first-order language is the set generated from the set of atomic formulas by formula-building operations* $\mathcal{F}_{\neg}$, $\mathcal{F}_{\rightarrow}$, *and* $\mathcal{F}_{\forall,i}$ *for each* $i = 1, 2, \ldots$.

Example 4.12 *In the language for number theory,* $\boldsymbol{< 0v_1}$ *and* $\boldsymbol{\approx 0v_1}$ *are atomic formulas and hence wffs in that language. Hence*

$$\mathcal{F}_{\neg}(\boldsymbol{< 0v_1}) = \boldsymbol{(\neg < 0v_1)},$$

$$\mathcal{F}_{\rightarrow}(\boldsymbol{(\neg < 0v_1)}, \boldsymbol{\approx 0v_1}) = \boldsymbol{((\neg < 0v_1) \rightarrow \approx 0v_1)},$$

and

$$\mathcal{F}_{\forall,1}(\boldsymbol{((\neg < 0v_1) \rightarrow \approx 0v_1)}) = \forall \boldsymbol{v_1((\neg < 0v_1) \rightarrow \approx 0v_1)}$$

are each wffs in that language as well. Note the different use of equality above. Remember that $\boldsymbol{\approx}$ *is a formal symbol in the language whereas we use "=" above to say what the output from a formula building operation is. The last wff is the translation of the Peano axiom "Zero is not the successor of any natural number." We translated this axiom earlier in our intuitive initial discussion, but this is the form in the strict use of our formal language.*

Example 4.13 *In the language of set theory, the wff*

$$\forall \boldsymbol{v_1}(\neg\forall \boldsymbol{v_2} \in \boldsymbol{v_2 v_1})$$

is the wff that translates the axiom that there is no set which contains every set. Compare this with the more informal Example 4.4*. Essentially we have replaced* $\boldsymbol{es}$ *with* $\neg\forall\neg$ *and simplified double negations.*

In the language of an arbitrary group (Group Theory), the wff

$$\forall \boldsymbol{v_1}(\approx *\boldsymbol{v_1 e v_1} \wedge \approx *\boldsymbol{e v_1 v_1})$$

This wff translates the group axiom that asserts the existence of inverses.

Like with sentential logic, there is a Unique Readability Theorem for the wffs in any first-order language. Although the methods are slightly different for the proof for first-order languages, we skip the development trusting that the reader will readily accept the fact of unique readability given the rigorous development for the sentential language.

4.2.1 Free Variables

Unlike sentential logic, not every wff will be able to translate an English (or other natural language) sentence. For instance, in the language of number theory, $< \boldsymbol{v_2 v_1}$ is an atomic formula, hence a wff. However, since $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ are each intended to be translated as simple place holders for any natural numbers, it cannot be said that there is an English *statement* that can be translated into this wff. It asserts nothing, but merely provides the structure necessary to say that one natural number is less than another. Unless specific natural numbers are specified or the natural numbers which might fill the place holders are delimited , we have essentially "___< ___".

On the other hand the wff $\forall \boldsymbol{v_1} \forall \boldsymbol{v_2} < \boldsymbol{v_2 v_1}$ has an English equivalent that is a proposition: "Every natural number is less than every other natural number." Now, this proposition is false, but that is beside the point. The point is that we have delimited both of the variables $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ so that we are actually making a claim. Of course such wffs are the most interesting to us, and so we define these notions more rigorously. We first want to design a recursively defined function that will mark by the number 1 the occurrence of a non-delimited (free) variable.

Let $\mathcal{A}$ denote the set of all atomic formulas, we define for any first-order language $h_v : \mathcal{A} \longrightarrow \{0,1\}$ by

$$h(\boldsymbol{\alpha}) = \begin{cases} 1 & \text{if } \boldsymbol{v} \text{ occurs in } \boldsymbol{\alpha} \\ 0 & \text{otherwise} \end{cases}$$

The Unique Readability Theorem (which we assume holds for all first-order languages) and the Recursion Theorem will guarantee the existence of a unique extension $\overline{h_v}$ of h_v where

$$\overline{h_v}(\boldsymbol{\alpha}) = h_v(\boldsymbol{\alpha}) \text{ for atomic formula } \boldsymbol{\alpha},$$

$$\overline{h_v}((\neg \boldsymbol{\alpha})) = \overline{h_v}(\boldsymbol{\alpha}),$$

$$\overline{h_v}((\boldsymbol{\alpha} \rightarrow \boldsymbol{\beta})) = \max\{\overline{h_v}(\boldsymbol{\alpha}), \overline{h_v}(\boldsymbol{\beta})\}, \text{ and}$$

$$\overline{h_v}(\forall \boldsymbol{v_i} \boldsymbol{\alpha}) = \begin{cases} \overline{h_v}(\boldsymbol{\alpha}) & \text{if } \boldsymbol{v} \neq \boldsymbol{v_i} \\ 0 & \text{if } \boldsymbol{v} = \boldsymbol{v_i} \end{cases}.$$

Definition 4.14 *In a first-order language a variable* $\boldsymbol{v}$ **occurs free** *in a wff* $\boldsymbol{\alpha}$ *if* $\overline{h_v}(\boldsymbol{\alpha}) = 1$.

Example 4.15 *In the language for number theory, for the wff* $< \boldsymbol{v_2 v_1}$*,*

$$\overline{h_{v_1}}(< \boldsymbol{v_2 v_1}) = 1$$

since $< \boldsymbol{v_2 v_1}$ *is atomic and* $\boldsymbol{v_1}$ *occurs in* $< \boldsymbol{v_2 v_1}$. *Similarly for* $\boldsymbol{v_2}$.

In the language of number theory, no variable occurs free in

$$\forall \boldsymbol{v_1}(\neg \forall \boldsymbol{v_2} \in \boldsymbol{v_2 v_1})$$

since $\overline{h_{v_i}}(\forall \boldsymbol{v_1}(\neg \forall \boldsymbol{v_2} \in \boldsymbol{v_2 v_1})) = 0$ *for each* i *since each variable* $\boldsymbol{v_i}$ *follows the quantifier it appeared in.*

Example 4.16 *Consider a first-order language with two 1-place predicate symbols* $\boldsymbol{A}$ *and* $\boldsymbol{C}$, *and one 2-place predicate symbol* $\heartsuit$. *In the wff,*

$$\forall \boldsymbol{v_1} \forall \boldsymbol{v_2}((\neg(\boldsymbol{A v_1} \rightarrow (\neg \boldsymbol{C v_2}))) \rightarrow \heartsuit \boldsymbol{v_1 v_2}),$$

there are no free variables as in the last example. However, in the wff

$$(\forall \boldsymbol{v_1}(\neg(\boldsymbol{A v_1} \rightarrow (\neg \boldsymbol{C v_2}))) \rightarrow \forall \boldsymbol{v_2} \heartsuit \boldsymbol{v_1 v_2}),$$

the variable $\boldsymbol{v_2}$ *is free since*

$$\begin{aligned}
&\overline{h_{v_2}}((\forall \boldsymbol{v_1}(\neg(\boldsymbol{A v_1} \rightarrow (\neg \boldsymbol{C v_2}))) \rightarrow \forall \boldsymbol{v_2} \heartsuit \boldsymbol{v_1 v_2})) \\
&= \max\{\overline{h_{v_2}}((\neg(\boldsymbol{A v_1} \rightarrow (\neg \boldsymbol{C v_2}))), \overline{h_{v_2}}(\forall \boldsymbol{v_2} \heartsuit \boldsymbol{v_1 v_2})\} \\
&= \max\{\overline{h_{v_2}}((\boldsymbol{A v_1} \rightarrow (\neg \boldsymbol{C v_2})), 0\} \\
&= \overline{h_{v_2}}((\boldsymbol{A v_1} \rightarrow (\neg \boldsymbol{C v_2})) \\
&= \max\{\overline{h_{v_2}}(\boldsymbol{A v_1}), \overline{h_{v_2}}((\neg \boldsymbol{C v_2}))\} \\
&= \max\{0, \overline{h_{v_2}}(\boldsymbol{C v_2})\} \\
&= 1
\end{aligned}$$

Note that in the first wff, every incidence of $\boldsymbol{v_2}$ *appeared after that variable was quantified whereas in the second case there is an incidence of* $\boldsymbol{v_2}$ *before it gets quantified. The variable* $\boldsymbol{v_1}$ *is also seen to be free applying* $\overline{h_{v_1}}$.

Of course, the wffs that we are most interested in are the wffs with no free variables because they provide the structure to be able to translate English propositions into our formal first-order language.

Definition 4.17 *A* **sentence** *is a wff in which there are no free variables.*

In Example 4.20, $\forall \boldsymbol{v_1}(\neg \forall \boldsymbol{v_2} \in \boldsymbol{v_2 v_1})$ is a sentence since it has no free variables, and in Example 4.21 $\forall \boldsymbol{v_1} \forall \boldsymbol{v_2}((\neg(\boldsymbol{A v_1} \to (\neg \boldsymbol{C v_2}))) \to \heartsuit \boldsymbol{v_1 v_2})$ is a sentence since it has no free variables.

4.2.2 Abbreviations

Having now discussed the creation of wffs in any first-order language in a rigorous fashion, we will adopt some abbreviating conventions to aid in the readability of our wffs in the further discussion of first-order logic. It is important to note that these abbreviations will not change our previous definitions or results dealing with first-order wffs, but will merely cut down on excess symbolism. The abbreviations which we make will also make the discussion of first-order languages closer to that which the reader has already seen in his or her mathematics foundations course. We abbreviate as follows.

$(\boldsymbol{\alpha} \vee \boldsymbol{\beta})$ abbreviates $((\neg \boldsymbol{\alpha}) \to \boldsymbol{\beta})$

$(\boldsymbol{\alpha} \wedge \boldsymbol{\beta})$ abbreviates $(\neg(\boldsymbol{\alpha} \to (\neg \boldsymbol{\beta})))$

$(\boldsymbol{\alpha} \leftrightarrow \boldsymbol{\beta})$ abbreviates $(\neg((\boldsymbol{\alpha} \to \boldsymbol{\beta}) \to (\neg(\boldsymbol{\beta} \to \boldsymbol{\alpha}))))$

i.e. $((\boldsymbol{\alpha} \to \boldsymbol{\beta}) \wedge (\boldsymbol{\beta} \to \boldsymbol{\alpha}))$ (we used this abbreviation above in our first informal discussion of first-order formulas)

$\exists \boldsymbol{v \alpha}$ abbreviates $(\neg \forall \boldsymbol{v}(\neg \boldsymbol{\alpha}))$ (we have already discussed this abbreviation)

$\boldsymbol{u} \approx \boldsymbol{t}$ abbreviates $\approx \boldsymbol{ut}$ for terms $\boldsymbol{u}$ and $\boldsymbol{t}$

(similarly for other predicate symbols)

$\boldsymbol{u} \not\approx \boldsymbol{t}$ abbreviates $(\neg \approx \boldsymbol{ut})$ and similarly for other predicate symbols

We also allow other parentheses such as [,], **{**, and **}**. Using these conventions, many of our unwieldy translations become easier to comprehend.

At this point, we have developed what a first-order formal language will look like. It is essential to note that even though we may have intended meanings for symbols in a particular first-order language, the symbols have no inherent meaning behind them. The important thing is that the language will be able to *support* a particular meaning.

Example 4.18 *Consider the language that we have titled "The Language of Set Theory." Since the symbols have no inherent meaning, we could think of the constant symbol* $\boldsymbol{\emptyset}$ *as having the meaning of the number zero, and we could think of the predicate symbol* $\in$ *as having the meaning of the less than relation. Our underlying set could be the natural numbers, the integers, the rational numbers, or the real numbers. Thus the wff*

$$\exists \boldsymbol{v_1} \boldsymbol{\emptyset} \in \boldsymbol{v_1}$$

could translate the assertion that there is a natural (or rational or real) number that zero is less than.

We need to make this notion of the ability of a formal language to support a particular meaning more precise. The ability of a particular language to support specific meaning will also play a direct role in what truth and falsity mean for first-order languages.

4.3 Structures, Truth, and Models

In the sentential language, truth assignments gave us our way of talking about our model for deductions, tautological implication. We wish to develop the analogous model in first-order languages and thus need something akin to truth assignments. In the sentential language, truth assignments began by assigning truth values to the sentence symbols as the core wffs in the language. In first order languages, atomic formulas ($\boldsymbol{P t_1 t_2 \cdots t_n}$) are our most fundamental wffs. Intuitively, they are true if property $\boldsymbol{P}$ is true of the n-terms $\boldsymbol{t_1}$ through $\boldsymbol{t_n}$. Now, each term is either a variable, a constant or some composition of functions applied to variables and constants. So, if we specify a specific set that our variables and constants are to come from and also specify what actual functions the function symbols refer to and what relations the predicate symbols refer to, we should be able to determine whether each atomic formula is true or false, and this will be determined by whether the n-tuple $(\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_n})$ (where the $\boldsymbol{t_i}$'s will be assigned specific numbers) is in the relation that the predicate symbol $\boldsymbol{P}$ refers to. We define things more formally.

Definition 4.19 *A* **structure** *for a first-order language is a function* $\mathfrak{S}$ *on the set of parameters of the language where*

(i) $\mathfrak{S}(\forall) = \mathbb{U}$ *where* $\mathbb{U}$ *is a non-empty set called the universe of* $\mathfrak{S}$*.*

(ii) $\mathfrak{S}(\boldsymbol{P}) = P^{\mathfrak{S}} \subseteq \mathbb{U}^n$ *for each n-place predicate symbol* $\boldsymbol{P}$ *i.e.* $P^{\mathfrak{S}}$ *is an n-ary relation.*

- $\mathfrak{S}$ *maps* $\boldsymbol{\approx}$ *to* $=$ *if* $\boldsymbol{\approx}$ *is a symbol in the language since* $\boldsymbol{\approx}$ *is both a logical symbol and a 2-place predicate symbol if it is in the formal language.*

(iii) $\mathfrak{S}(\boldsymbol{c}) = c^{\mathfrak{S}} \in \mathbb{U}$ *for constant symbol* $\boldsymbol{c}$*.*

(iv) $\mathfrak{S}(\boldsymbol{f}) = f^{\mathfrak{S}}$ *where* $f^{\mathfrak{S}} : \mathbb{U}^n \longrightarrow \mathbb{U}$ *for the n-place function symbol* $\boldsymbol{f}$*.*

As a technical note, if $\mathfrak{S}$ is a function, what is its codomain? We can construct an ad hoc codomain by including in our set the universal set that we want to translate our formal language into, then including in our codomain the variety of n-ary relations on the universal set that we want to translate our predicate symbols to, etc.

Structures are the first-order equivalent to truth assignments in the sentential language. The idea behind assigning $\forall$ to a non-empty set is that now $\forall \boldsymbol{v}$ will mean "for every element in our specific universe". The assignments for the other symbols will give us concrete relations, constants, and functions to work with in our universe. Truth and falsity will be determined by satisfaction of our relations. Until a structure is specified, our formal language is just symbology with specific rules for how to combine that symbology.

Example 4.20 *We look at structures for the first-order language that we termed "The Language of Number Theory." Define a structure* $\mathfrak{N}$ *by* $\mathfrak{N}(\forall) = \mathbb{N}$, $\mathfrak{N}(<) = < \subseteq \mathbb{N}^2$ *(given its usual interpretation),* $\mathfrak{N}(\mathbf{0}) = 0$, $\mathfrak{N}(\boldsymbol{S}) = S$ *where* $S(x) = x+1$, $\mathfrak{N}(+) = +$ *(addition),* $\mathfrak{N}(\cdot) = \cdot$ *(multiplication), and* $\mathfrak{N}(\boldsymbol{E}) = E$ *(exponentiation). This is of course the actual structure of number theory.*

Define another structure $\mathfrak{B}$ *by* $\mathfrak{B}(\forall) = \mathbb{R}$, $\mathfrak{B}(<) = < \subseteq \mathbb{R}^2$ *(again, given its normal meaning),* $\mathfrak{B}(\mathbf{0}) = 0$, $\mathfrak{B}(\boldsymbol{S}) = S$ *where* $S(x) = x+1$, $\mathfrak{B}(+) = +$, $\mathfrak{B}(\cdot) = \cdot$, *and* $\mathfrak{B}(\boldsymbol{E}) = E$. *We have merely used a larger universe in this case than in the previous case.*

Define yet another structure $\mathfrak{C}$ *by* $\mathfrak{C}(\forall) = \mathbb{Z}$, $\mathfrak{C}(<) = D \subseteq \mathbb{Z}^2$ *where* $D = \{(m, n) : m \text{ divides } n\}$, $\mathfrak{C}(\mathbf{0}) = 1$, $\mathfrak{C}(\boldsymbol{S}) = P$ *where* $P(x) = x - 1$, $\mathfrak{C}(+) = \cdot$, $\mathfrak{C}(\cdot) = +$, *and* $\mathfrak{C}(\boldsymbol{E}) = E$.

Each of the structures above are completely valid for the formal languages we have specified, but some may be non-intuitive given our discussion above about "intended interpretations" of our formal symbols. Again, our formal languages are just a collection of symbols with rules how to combine them without any inherent meaning. We may want our formal language to be able to *support* a particular structure, but the language may be able to support many, many structures. We consider one more example of a structure at this point.

Example 4.21 *For the formal language for an arbitrary group, we can define the following structures.*

$$\mathfrak{A}(\forall) = \mathbb{Z}_2 \times \mathbb{Z}_2 = \{(0,0),(1,0),(0,1),(1,1)\},\ \mathfrak{A}(\in) = \mathbb{Z}_2 \times \mathbb{Z}_2$$

$$(\text{"is a member of } \mathbb{Z}_2 \times \mathbb{Z}_2\text{"}),\ \mathfrak{A}(\boldsymbol{e}) = (0,0),\ \mathfrak{A}(*) = + \text{ where}$$

$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$$

$$\mathfrak{B}(\forall) = Aut(\mathbb{Z}_p) \text{ where } p \text{ is a prime (the set of automorphisms on the group } \mathbb{Z}_p),$$

$$\mathfrak{B}(\in) = Aut(\mathbb{Z}_p),\ \mathfrak{A}(\boldsymbol{e}) = \iota \text{ where } \iota(x) = x \text{ (the identity map)},$$

$$\mathfrak{B}(*) = \circ \text{ (function composition)}$$

$$\mathfrak{C}(\forall) = \mathbb{R},\ \mathfrak{C}(\in) = \mathbb{R},\ \mathfrak{C}(\boldsymbol{e}) = 1,\ \mathfrak{C}(*) = \cdot \text{ (multiplication)}$$

Notice that all of the above functions are indeed structures since they fulfill the definition of a structure. However, even though we are using the formal language "for group theory," only the first two structures are bona fide groups. The third structure is not a group because it fails to fulfill the inverse axiom for groups. Now, above, in our presentation of the formal language for group theory, we claimed that we were able to represent the four group axioms using the formal language. Of course, we were doing this while implicitly thinking of our symbols referring to a concrete, but arbitrary group. We would expect then with the last structure in Example 4.28, that the wff sentence used to express the existence of inverses would be false in this particular structure (since $\mathbb{R}$ fails to be a group because 0 has no inverse), but all four wff sentences intended to express the four group axioms would be true in first two structures of Example 4.21 since they are indeed groups. We thus need to express formally what it will mean for a wff sentence to be true under a particular structure.

First, given a structure $\mathfrak{S}$, we begin with a function that assigns to all of our variable symbols in our formal language to concrete points in our specific universe $\mathbb{U}$. So let $s : \mathcal{V} \longrightarrow \mathbb{U}$ where $\mathcal{V}$ is the set of variable symbols in the formal language. Each such function s is like a truth assignment v in sentential logic. Having specified our variables, we wish to extend to a function $\overline{s}$ on all terms. By free generation (unique readability) of the terms from the variables and constant symbols in our language and by the Recursion Theorem, there will exist a unique function $\overline{s} : \mathcal{T} \longrightarrow \mathbb{U}$ where $\mathcal{T}$ is the set of all terms in our first-order language such that

$$\overline{s}(\boldsymbol{v}) = s(\boldsymbol{v}) \text{ for variable symbol } \boldsymbol{v},$$

$$\overline{s}(\boldsymbol{c}) = c^{\mathfrak{S}} \text{ for a constant symbol } \boldsymbol{c}, \text{ and}$$

$$\text{for terms } \boldsymbol{t_1},\ \boldsymbol{t_2}, \ldots,\ \boldsymbol{t_n} \text{ and } n\text{-place function symbol } \boldsymbol{f}$$

$$\overline{s}(\boldsymbol{f t_1 t_2 \cdots t_n}) = f^{\mathfrak{S}}(\overline{s}(\boldsymbol{t_1}), \overline{s}(\boldsymbol{t_2}), \ldots, \overline{s}(\boldsymbol{t_n})).$$

Thus, $\overline{s}$ assigns concrete points in the universe $\mathbb{U}$ to the terms in our formal language.

We are now in a position to define what it means for a wff $\boldsymbol{\varphi}$ to be satisfied in a structure.

Definition 4.22 *A wff* $\boldsymbol{\varphi}$ *is* **satisfied in a structure** $\mathfrak{S}$ **by** $s : \mathcal{V} \longrightarrow \mathbb{U}$ *and we denote this with* $\models_{\mathfrak{S}} \boldsymbol{\varphi}[s]$ *subject to one of the cases for the form of* $\boldsymbol{\varphi}$ *described below.*

(i) $\models_{\mathfrak{S}} \boldsymbol{\approx t_1 t_2}$ *if and only if* $\overline{s}(\boldsymbol{t_1}) = \overline{s}(\boldsymbol{t_2})$

(ii) $\models_{\mathfrak{S}} \boldsymbol{P t_1 t_2 \cdots t_n}$ *if and only if* $(\overline{s}(\boldsymbol{t_1}), \overline{s}(\boldsymbol{t_2}), \ldots, \overline{s}(\boldsymbol{t_n})) \in P^{\mathfrak{S}}$

(iii) $\models_{\mathfrak{S}} (\boldsymbol{\neg\psi})[s]$ *if and only if* $\not\models_{\mathfrak{S}} \boldsymbol{\psi}[s]$

(iv) $\models_{\mathfrak{S}} (\boldsymbol{\psi \to \chi})[s]$ *if and only if* $\not\models_{\mathfrak{S}} \boldsymbol{\psi}[s]$ *or* $\models_{\mathfrak{S}} \boldsymbol{\chi}[s]$

(v) $\models_{\mathfrak{S}} \forall \boldsymbol{x\psi}[s]$ *if and only if for all* $u \in \mathbb{U}$, $\models_{\mathfrak{S}} \boldsymbol{\psi}[s_{\boldsymbol{x}|u}]$ *where*

$$s_{\boldsymbol{x}|u}(\boldsymbol{y}) = \begin{cases} u & \text{for } \boldsymbol{y} = \boldsymbol{x} \\ s(\boldsymbol{y}) & \text{for } \boldsymbol{y} \neq \boldsymbol{x}. \end{cases}$$

Item (v) deserves some consideration. Informally when we say "for all numbers in a set, $\boldsymbol{\varphi}$ is true," we mean that if $\boldsymbol{\varphi}$ was said about any specific element in the universal set, the statement would be true. The idea with the piecewise function is that whenever the quantified variable symbol is encountered, it will be replaced with the element u from the universe. If we do this for every $u \in \mathbb{U}$ and $\boldsymbol{\varphi}$ is true in every case, then we should say that $\forall \boldsymbol{x\varphi}$ is satisfied with the function s (the function s is given before referring to $s_{\boldsymbol{x}|u}$). The function s still matters because the $\boldsymbol{\varphi}$ could have free variables (ones distinct from $\boldsymbol{x}$).

Since the wffs are freely generated by the atomic formulas in a particular language, the definition is valid by the Recursion Theorem. Although most of our following examples use strict formal languages without the use of the abbreviations $\wedge$, $\vee$, $\leftrightarrow$, and $\exists$, it is a simple matter to prove the following useful facts (we omit the proof).

Theorem 4.23 *The following hold for any structure $\mathfrak{S}$ and any $s : \mathcal{V} \longrightarrow \mathbb{U}$.*

(i) $\models_{\mathfrak{S}} (\boldsymbol{\varphi} \wedge \boldsymbol{\psi})[s]$ *if and only if* $\models_{\mathfrak{S}} \boldsymbol{\varphi}[s]$ *and* $\models_{\mathfrak{S}} \boldsymbol{\psi}[s]$.

(ii) $\models_{\mathfrak{S}} (\boldsymbol{\varphi} \vee \boldsymbol{\psi})[s]$ *if and only if* $\models_{\mathfrak{S}} \boldsymbol{\varphi}[s]$ *or* $\models_{\mathfrak{S}} \boldsymbol{\psi}[s]$.

(iii) $\models_{\mathfrak{S}} (\boldsymbol{\varphi} \leftrightarrow \boldsymbol{\psi})[s]$ *if and only if* $\models_{\mathfrak{S}} \boldsymbol{\varphi}[s]$ *and* $\models_{\mathfrak{S}} \boldsymbol{\psi}[s]$ *or* $\not\models_{\mathfrak{S}} \boldsymbol{\varphi}[s]$ *and* $\not\models_{\mathfrak{S}} \boldsymbol{\psi}[s]$.

(iv) $\models_{\mathfrak{S}} \exists \boldsymbol{x} \boldsymbol{\varphi}[s]$ *if and only if there is some $u \in \mathbb{U}$ such that* $\models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{x|u}]$.

A couple of examples of satisfaction (and non-satisfaction) of a wff by are in order.

Example 4.24 *Consider the first-order language that we developed for number theory and the wff in the first-order language,*

$$\forall \boldsymbol{v_1}((\neg < \mathbf{0}\boldsymbol{v_1}) \rightarrow \approx \mathbf{0}\boldsymbol{v_1})$$

Let $\mathfrak{A}$ be the actual structure for number theory as defined above. Let $s : \mathcal{V} \longrightarrow \mathbb{N}$ where $s(v_1) = 1$ and $s(v_i) = 0$ where $i \neq 1$. By the definition of satisfaction in the structure $\mathfrak{A}$ by the s we have that

$$\models_{\mathfrak{A}} \forall \boldsymbol{v_1}((\neg < \mathbf{0}\boldsymbol{v_1}) \rightarrow \approx \mathbf{0}\boldsymbol{v_1})[s] \textit{ iff}$$

$$\models_{\mathfrak{A}} ((\neg < \mathbf{0}\boldsymbol{v_1}) \rightarrow \approx \mathbf{0}\boldsymbol{v_1})[s_{\boldsymbol{v_1}|n}] \textit{ for all } n \in \mathbb{N} \textit{ iff}$$

$$\not\models_{\mathfrak{A}} (\neg < \mathbf{0}\boldsymbol{v_1})[s_{\boldsymbol{v_1}|n}] \textit{ or } \models_{\mathfrak{A}} \approx \mathbf{0}\boldsymbol{v_1} \; [s_{\boldsymbol{v_1}|n}] \textit{ for all } n \in \mathbb{N} \textit{ iff}$$

$$\models_{\mathfrak{A}} < \mathbf{0}\boldsymbol{v_1} \; [s_{\boldsymbol{v_1}|n}] \textit{ or } \overline{s_{\boldsymbol{v_1}|n}}(\mathbf{0}) = \overline{s_{\boldsymbol{v_1}|n}}(\boldsymbol{v_1}) \textit{ for all } n \in \mathbb{N} \textit{ iff}$$

$$(\overline{s_{\boldsymbol{v_1}|n}}(\mathbf{0}), \overline{s_{\boldsymbol{v_1}|n}}(\boldsymbol{v_1})) \in < \textit{ or } 0 = s_{\boldsymbol{v_1}|n}(\boldsymbol{v_1}) \textit{ for all } n \in \mathbb{N} \textit{ iff}$$

$$0 < s_{\boldsymbol{v_1}|n}(\boldsymbol{v_1}) \textit{ or } 0 = n \textit{ for all } n \in \mathbb{N} \textit{ iff}$$

$$0 < n \textit{ or } 0 = n \textit{ for all } n \in \mathbb{N}$$

Since this last statement is true for the natural numbers, we may say that

$$\models_{\mathfrak{A}} \forall \boldsymbol{v_1}((\neg < \mathbf{0}\boldsymbol{v_1}) \rightarrow \approx \mathbf{0}\boldsymbol{v_1})[s]$$

is true, that is, the wff

$$\forall \boldsymbol{v_1}((\neg < \mathbf{0}\boldsymbol{v_1}) \rightarrow \approx \mathbf{0}\boldsymbol{v_1})$$

is satisfied in the structure $\mathfrak{A}$ *by* s. *Notice also that nowhere was the fact that* $s(v_1) = 1$ *and* $s(v_i) = 0$ *for* $i \neq 1$ *used. Thus, our statements must hold for any arbitrary* $s : \mathcal{V} \longrightarrow \mathbb{N}$, *that is, any assignment of the variable symbols to the natural numbers (recall that* $\mathcal{V}$ *is the set of variable symbols for a fixed language under consideration). So,*

$$\models_{\mathfrak{A}} \forall \boldsymbol{v_1}((\neg < \mathbf{0}\boldsymbol{v_1}) \rightarrow \approx \mathbf{0}\boldsymbol{v_1})[s]$$

is true for any arbitrary s. *Or equivalently,*

$$\forall \boldsymbol{v_1}((\neg < \mathbf{0}\boldsymbol{v_1}) \rightarrow \approx \mathbf{0}\boldsymbol{v_1})$$

is satisfied in the structure $\mathfrak{A}$ *for any* s. *This fact should make sense since in this structure our wff essentially asserts that zero is the least natural number, which is one of the Peano axioms that defines the natural numbers. So, this wff being satisfied in this structure for every* s, *indicates the truth of the wff no matter how the variable* $\boldsymbol{v_1}$ *is assigned to a natural number. In other words, the wff is true in this particular structure no matter how the variable involved is assigned.*

Example 4.25 *Consider the language that we developed for an arbitrary group and the wff* $\in \boldsymbol{*v_1v_2}$. *Consider the structure* $\mathfrak{A}$ *of the Klein group as presented in Example* 4.21. *Let* $s : \mathcal{V} \longrightarrow \mathbb{Z}_2 \times \mathbb{Z}_2$ *be defined by* $s(\boldsymbol{v_1}) = (0, 0)$ *and*

$s(\boldsymbol{v_2}) = (1,1)$ *(note that it is immaterial where we map all the infinitely many other variables since only the variables* $\boldsymbol{v_1}$ *and* $\boldsymbol{v_2}$ *are involved in the wff at hand). Then,*

$$\models_{\mathfrak{A}} \in *\boldsymbol{v_1}\boldsymbol{v_2}\ [s] \ \textit{iff}$$

$$\overline{s}(*\boldsymbol{v_1}\boldsymbol{v_2}) \textit{ is a member of the Klein group iff}$$

$$\overline{s}(\boldsymbol{v_1}) + \overline{s}(\boldsymbol{v_2}) \textit{ is a member of the Klein group iff}$$

$$s(\boldsymbol{v_1}) + s(\boldsymbol{v_2}) \textit{ is a member of the Klein group iff}$$

$$(0,0) + (1,1) \textit{ is a member of the Klein group iff}$$

$$(1,1) \textit{ is a member of the Klein group.}$$

Since this is true, $\mathfrak{A}$ *satisfies* $\in *\boldsymbol{v_1}\boldsymbol{v_2}$ *with* s*. It is clear because of the closure of the Klein group that this will be true for every* $s : \mathcal{V} \longrightarrow \mathbb{Z}_2 \times \mathbb{Z}_2$*, but notice unlike the last example the specific assignments for* $\boldsymbol{v_1}$ *and* $\boldsymbol{v_2}$ *are appealed to in our string of "iff"s. This is so since the variables* $\boldsymbol{v_1}$ *and* $\boldsymbol{v_2}$ *appear free in the wff in question. This fact suggests the next theorem, but first we give one more example.*

Example 4.26 *Again, we use the language developed for arbitrary groups and we consider the structure* $\mathfrak{C}$ *defined in Example 4.28. Consider the wff*

$$(\neg\forall\boldsymbol{v_2}(\neg \approx *\boldsymbol{v_1}\boldsymbol{v_2}\boldsymbol{e})).$$

Define $s : \mathcal{V} \longrightarrow \mathbb{R}$ *by* $s(v_1) = 0$ *and* $s(v_2) = 1$ *(It is immaterial where we map* $\boldsymbol{v_2}$ *since it does not occur free in the wff). Now,*

$$\models_{\mathfrak{C}} (\neg\forall\boldsymbol{v_2}(\neg \approx *\boldsymbol{v_1}\boldsymbol{v_2}\boldsymbol{e}))[s] \ \textit{iff}$$

$$\not\models_{\mathfrak{C}} \forall\boldsymbol{v_2}(\neg \approx *\boldsymbol{v_1}\boldsymbol{v_2}\boldsymbol{e})[s] \ \textit{iff}$$

$$\not\models_{\mathfrak{C}} (\neg \approx *\boldsymbol{v_1 v_2 e})[s_{v_2|r}] \textit{ for some } r \in \mathbb{R}$$

$$\textit{(note that this is so since} \models_{\mathfrak{S}} \forall \boldsymbol{x\varphi}[s] \Leftrightarrow \models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{\boldsymbol{x}|u}] \textit{ for every } u \in \mathbb{U}$$

$$\textit{which is equivalent to } \not\models_{\mathfrak{S}} \forall \boldsymbol{x\varphi}[s] \Leftrightarrow \not\models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{x|u}] \textit{ for some } u \in \mathbb{U}\textit{) iff}$$

$$\models_{\mathfrak{C}} \approx *\boldsymbol{v_1 v_2 e}\ [s_{v_2|r}] \textit{ for some } r \in \mathbb{R} \textit{ iff}$$

$$\overline{s_{v_2|r}}(*\boldsymbol{v_1 v_2}) = \overline{s_{v_2|r}}(\boldsymbol{e}) \textit{ for some } r \in \mathbb{R} \textit{ iff}$$

$$\overline{s_{v_2|r}}(\boldsymbol{v_1}) \cdot \overline{s_{v_2|r}}(\boldsymbol{v_2}) = 1 \textit{ for some } r \in \mathbb{R} \textit{ iff}$$

$$s_{v_2|r}(\boldsymbol{v_1}) \cdot s_{v_2|r}(\boldsymbol{v_2}) = 1 \textit{ for some } r \in \mathbb{R} \textit{ iff}$$

$$s(\boldsymbol{v_1}) \cdot r = 1 \textit{ for some } r \in \mathbb{R} \textit{ iff}$$

$$0 \cdot r = 1 \textit{ for some } r \in \mathbb{R}.$$

This last statement is false since $0 \cdot r = 0$ *for every* $r \in \mathbb{R}$*, so we see that*

$$\not\models_{\mathfrak{C}} (\neg \forall \boldsymbol{v_2}(\neg \approx *\boldsymbol{v_1 v_2 e}))[s].$$

That is, this wff is not satisfied in the structure $\mathfrak{C}$ *with (the meaning determined by)* s*. Of course, if*
$s(v_1) = a \neq 0$*, the wff would have been satisfied.*

We have noted a couple of things in the last several examples. First, we have noted that the function $s : \mathcal{V} \longrightarrow \mathbb{U}$ plays a role in the question of satisfaction of a wff only if there are free variables involved in the wff in question. Second, we have seen that it is immaterial to the question of satisfaction of a particular wff how s maps any variable that is not involved in the wff. These facts suggest the following theorem (note that in the theorem $s_1\big|_{\mathcal{F}_\varphi}$ represents the restriction of s_1 to the set $\mathcal{F}_\varphi$).

Theorem 4.27 *Let $\mathcal{F}_\varphi$ denote the set of all free variables involved in the wff $\boldsymbol{\varphi}$. For structure $\mathfrak{S}$ and any $s_1, s_2 : \mathcal{V} \longrightarrow \mathbb{U}$, if $s_1\big|_{\mathcal{F}_\varphi} = s_2\big|_{\mathcal{F}_\varphi}$, then*

$$\models_{\mathfrak{S}} \boldsymbol{\varphi}[s_1] \text{ if and only if } \models_{\mathfrak{S}} \boldsymbol{\varphi}[s_2].$$

Proof: The proof is another induction argument on the set of wffs

$$I = \{\boldsymbol{\varphi} : \text{Such that the theorem holds true for } \boldsymbol{\varphi}\}.$$

The first step is to show that all atomic formulas are included in this set and then show closure under negation, implication, and universal quantification. We show everything except the case for negation.

Let $\boldsymbol{\varphi}$ be an atomic formula. Then $\boldsymbol{\varphi} = \boldsymbol{P t_1 t_2 \cdots t_n}$ where $\boldsymbol{P}$ is a n-ary predicate symbol and each $\boldsymbol{t_i}$ is a term. Take an arbitrary pair of functions s_1 and s_2 and suppose that they agree on all of the free variables involved in $\boldsymbol{\varphi}$. Since $\boldsymbol{\varphi}$ is atomic, every variable involved in $\boldsymbol{\varphi}$ is free. It is clear then that $\overline{s_1}(\boldsymbol{t_i}) = \overline{s_2}(\boldsymbol{t_i})$. Now,

$$\models_{\mathfrak{S}} \boldsymbol{\varphi}[s_1] \text{ iff } (\overline{s_1}(\boldsymbol{t_1}), \overline{s_1}(\boldsymbol{t_2}), \ldots, \overline{s_1}(\boldsymbol{t_n})) \in P^{\mathfrak{S}} \text{ iff}$$

$$(\overline{s_2}(\boldsymbol{t_1}), \overline{s_2}(\boldsymbol{t_2}), \ldots, \overline{s_2}(\boldsymbol{t_n})) \in P^{\mathfrak{S}} \text{ iff } \models_{\mathfrak{S}} \boldsymbol{\varphi}[s_2].$$

Since our choice for $\boldsymbol{\varphi}$ as an atomic formula was arbitrary, every atomic formula must fulfil the theorem and hence is in the set I.

Now suppose that $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$ are in I. Again, take an arbitrary pair of functions s_1 and s_2 and suppose that

$$s_1\big|_{\mathcal{F}_{(\varphi\to\psi)}} = s_2\big|_{\mathcal{F}_{(\varphi\to\psi)}}.$$

Note that $\mathcal{F}_{\boldsymbol{\varphi}}, \mathcal{F}_{\boldsymbol{\psi}} \subseteq \mathcal{F}_{(\boldsymbol{\varphi}\to\boldsymbol{\psi})}$. Thus,

$$s_1\big|_{\mathcal{F}_\varphi} = s_2\big|_{\mathcal{F}_\varphi} \text{ and } s_1\big|_{\mathcal{F}_\psi} = s_2\big|_{\mathcal{F}_\psi}.$$

By assumption of $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$ being in I, we must have

$$\models_{\mathfrak{G}} \boldsymbol{\varphi}[s_1] \text{ iff } \models_{\mathfrak{G}} \boldsymbol{\varphi}[s_2] \text{ and } \models_{\mathfrak{G}} \boldsymbol{\psi}[s_1] \text{ iff } \models_{\mathfrak{G}} \boldsymbol{\psi}[s_2].$$

Equivalently, we must have

$$\models_{\mathfrak{G}} \boldsymbol{\varphi}[s_1] \text{ and } \models_{\mathfrak{G}} \boldsymbol{\psi}[s_1] \text{ iff } \models_{\mathfrak{G}} \boldsymbol{\varphi}[s_2] \text{ and } \models_{\mathfrak{G}} \boldsymbol{\psi}[s_2].$$

By Definition 4.22, these are sufficient conditions to guarantee that

$$\models_{\mathfrak{G}} \boldsymbol{(\varphi \to \psi)}[s_1] \text{ iff } \models_{\mathfrak{G}} \boldsymbol{(\varphi \to \psi)}[s_2].$$

Thus, we must have $\boldsymbol{(\varphi \to \psi)} \in I$.

Finally, we demonstrate closure under universal quantification. Suppose for variable symbol $\boldsymbol{v_i}$ and an arbitrary pair of variable assignments s_1 and s_2 that

$$s_1\big|_{\mathcal{F}_{\forall \boldsymbol{v_i} \boldsymbol{\varphi}}} = s_2\big|_{\mathcal{F}_{\forall \boldsymbol{v_i} \boldsymbol{\varphi}}}.$$

Note that

$$\mathcal{F}_{\forall \boldsymbol{v_i} \boldsymbol{\varphi}} \subseteq \mathcal{F}_{\varphi} \subseteq \mathcal{F}_{\forall \boldsymbol{v_i} \boldsymbol{\varphi}} \cup \{\boldsymbol{v_i}\}.$$

Also, we have

$$(s_1)_{\boldsymbol{v_i}|u}(\boldsymbol{y}) = \begin{cases} u & \text{for } \boldsymbol{y} = \boldsymbol{v_i} \\ s_1(\boldsymbol{y}) & \text{for } \boldsymbol{y} \neq \boldsymbol{v_i}. \end{cases} \quad \text{and}$$

$$(s_2)_{\boldsymbol{v_i}|u}(\boldsymbol{y}) = \begin{cases} u & \text{for } \boldsymbol{y} = \boldsymbol{v_i} \\ s_2(\boldsymbol{y}) & \text{for } \boldsymbol{y} \neq \boldsymbol{v_i}. \end{cases}$$

These functions are equivalent to the following

$$(s_1)_{\boldsymbol{v_i}|u}(\boldsymbol{y}) = \begin{cases} u & \text{for } \boldsymbol{y} = \boldsymbol{v_i} \\ s_1(\boldsymbol{y}) & \text{for } \boldsymbol{y} \in \mathcal{F}_{\forall \boldsymbol{v_i} \boldsymbol{\varphi}} \\ s_1(\boldsymbol{y}) & \text{otherwise} \end{cases} \quad \text{and}$$

$$(s_2)_{\boldsymbol{v_i}|u}(\boldsymbol{y}) = \begin{cases} u & \text{for } \boldsymbol{y} = \boldsymbol{v_i} \\ s_2(\boldsymbol{y}) & \text{for } \boldsymbol{y} \in \mathcal{F}_{\forall \boldsymbol{v_i}\boldsymbol{\varphi}} \\ s_2(\boldsymbol{y}) & \text{otherwise} \end{cases}$$

Since we have assumed that

$$s_1\big|_{\mathcal{F}_{\forall \boldsymbol{v_i}\boldsymbol{\varphi}}} = s_2\big|_{\mathcal{F}_{\forall \boldsymbol{v_i}\boldsymbol{\varphi}}}$$

and $(s_1)_{\boldsymbol{v_i}|u}$ and $(s_2)_{\boldsymbol{v_i}|u}$ agree on all variables in $\mathcal{F}_{\forall \boldsymbol{v_i}\boldsymbol{\varphi}} \cup \{\boldsymbol{v_i}\} \supseteq \mathcal{F}_{\boldsymbol{\varphi}}$, we have

$$(s_1)_{\boldsymbol{v_i}|u}\big|_{\mathcal{F}_{\boldsymbol{\varphi}}} = (s_2)_{\boldsymbol{v_i}|u}\big|_{\mathcal{F}_{\boldsymbol{\varphi}}}.$$

Since we have assumed that $\boldsymbol{\varphi} \in I$ and thus fulfils the theorem, we have that

$$\models_{\mathfrak{S}} \boldsymbol{\varphi}[(s_1)_{\boldsymbol{v_i}|u}] \text{ iff } \models_{\mathfrak{S}} \boldsymbol{\varphi}[(s_2)_{\boldsymbol{v_i}|u}].$$

Hence,

$$\text{for all } u \in \mathbb{U} \models_{\mathfrak{S}} \boldsymbol{\varphi}[(s_1)_{\boldsymbol{v_i}|u}] \text{ iff for all } u \in \mathbb{U} \models_{\mathfrak{S}} \boldsymbol{\varphi}[(s_2)_{\boldsymbol{v_i}|u}].$$

By Definition 4.22, this last statement is so if and only if

$$\models_{\mathfrak{S}} \forall \boldsymbol{v_i}\boldsymbol{\varphi}[s_1] \text{ iff } \models_{\mathfrak{S}} \forall \boldsymbol{v_i}\boldsymbol{\varphi}[s_2].$$

Therefore, $\forall \boldsymbol{v_i}\boldsymbol{\varphi} \in I$ for every variable symbol $\boldsymbol{v_i}$. Closure under negation follows similarly. Hence I is exactly the set of first-order wffs for a given fixed language. ∎

Note that satisfaction of a wff depends on both the structure $\mathfrak{S}$ and $s : \mathcal{V} \longrightarrow \mathbb{U}$. So, we may state an analogous result to the last theorem. Recall that a structure is a function on the parameters of a specific language. Thus, we may write $\mathfrak{S}\big|_{\mathcal{P}}$ to restrict the structure to the parameters present in the set $\mathcal{P}$.

Theorem 4.28 *Let* $\mathcal{P}_\varphi$ *denote the set of all parameters in the wff* $\boldsymbol{\varphi}$*. For two structures* $\mathfrak{A}$ *and* $\mathfrak{B}$ *if* $\mathfrak{A}\big|_{\mathcal{P}_\varphi} = \mathfrak{B}\big|_{\mathcal{P}_\varphi}$*, then*

$$\models_{\mathfrak{A}} \boldsymbol{\varphi}[s] \text{ if and only if } \models_{\mathfrak{B}} \boldsymbol{\varphi}[s].$$

Proof: Clear. ∎

Having developed satisfaction in a structure as our analogy to truth assignments in sentential logic, we are in perfect position to define our model for implication.

Definition 4.29 *Let* Σ *be a set of wffs in a first-order language, and* $\boldsymbol{\varphi}$ *a wff.*

(i) Σ **logically implies** $\boldsymbol{\varphi}$*, and we write* $\Sigma \vDash \boldsymbol{\varphi}$*, if for every structure* $\mathfrak{S}$ *for the language and every function* $s : \mathcal{V} \longrightarrow \mathbb{U}$ *where* $\mathfrak{S}$ *satisfies every member of* Σ *with* s*,* $\mathfrak{S}$ *satisfies* $\boldsymbol{\varphi}$ *with* s*.*

(ii) Two wffs $\boldsymbol{\varphi}$ *and* $\boldsymbol{\psi}$ *are said to be* **logically equivalent** *(*$\boldsymbol{\varphi} \models\!\!\!\dashv \boldsymbol{\psi}$*) if* $\boldsymbol{\varphi} \vDash \boldsymbol{\psi}$ *and* $\boldsymbol{\psi} \vDash \boldsymbol{\varphi}$*.*

(iii) A wff $\boldsymbol{\varphi}$ *is said to be* **valid** *if* $\emptyset \vDash \boldsymbol{\varphi}$ *(abbreviated by* $\vDash \boldsymbol{\varphi}$*. (Note that a wff will be valid by definition if and only if every structure* $\mathfrak{S}$ *and every* $s : \mathcal{V} \longrightarrow \mathbb{U}$ *satisfies* $\boldsymbol{\varphi}$ *since* $\models_{\mathfrak{S}} \boldsymbol{\chi}[s]$ *holds for all* $\boldsymbol{\chi} \in \emptyset$*.)*

We give some examples of logical implication.

Example 4.30 *Consider a formal language with a 1-place predicate symbol* $\boldsymbol{Q}$*. We claim that* $\forall \boldsymbol{v_1} \boldsymbol{Q} \boldsymbol{v_1} \vDash \boldsymbol{Q} \boldsymbol{v_2}$*. Let* $\mathfrak{S}$ *be a structure for the language and let* $s : \mathcal{V} \longrightarrow \mathbb{U}$ *be any function such that* $\models_{\mathfrak{S}} \forall \boldsymbol{v_1} \boldsymbol{Q} \boldsymbol{v_1}[s]$*. We must show that* $\models_{\mathfrak{S}} \boldsymbol{Q}\boldsymbol{v_2}[s]$*. Note that this is so if and only if* $\overline{s}(\boldsymbol{v_2}) \in Q^{\mathfrak{S}}$ *which is true if and only if* $s(\boldsymbol{v_2}) \in Q^{\mathfrak{S}}$*. Now,* $\overline{s_{v_1|s(v_2)}}(\boldsymbol{v_1}) = s_{v_1|s(v_2)}(\boldsymbol{v_1}) = s(\boldsymbol{v_2})$*. So,* $s(\boldsymbol{v_2}) \in Q^{\mathfrak{S}}$ *if and only if* $\overline{s_{v_1|s(v_2)}}(\boldsymbol{v_1}) \in Q^{\mathfrak{S}}$ *which is so if and only if* $\models_{\mathfrak{S}} \boldsymbol{Q}\boldsymbol{v_1}[s_{v_1|s(v_2)}]$*.*

This holds since $\models_{\mathfrak{S}} \forall \boldsymbol{v_1} \boldsymbol{Q} \boldsymbol{v_1}[s]$ *if and only if* $\models_{\mathfrak{S}} \boldsymbol{Q}\boldsymbol{v_1}[s_{v_1|u}]$ *for all* $u \in \mathbb{U}$, *and* $s(\boldsymbol{v_2}) \in \mathbb{U}$. *So,* $\mathfrak{S}$ *satisfies* $\boldsymbol{Q}\boldsymbol{v_2}$ *with* s. *Since our choices for* $\mathfrak{S}$ *and* $s : \mathcal{V} \longrightarrow \mathbb{U}$ *were arbitrary, we may say that* $\forall \boldsymbol{v_1} \boldsymbol{Q} \boldsymbol{v_1} \vDash \boldsymbol{Q}\boldsymbol{v_2}$ *by definition. Intuitively, this logical implication holds since if something holds for the entire universe, it must hold for each specific element in the universe.*

Example 4.31 *Consider any formal first-order language with a 1-place predicate symbol* $\boldsymbol{Q}$. *We claim that* $\boldsymbol{Q}\boldsymbol{v_1} \nvDash \forall \boldsymbol{v_1} \boldsymbol{Q} \boldsymbol{v_1}$. *Take a structure* $\mathfrak{A}$ *such that* $\mathfrak{A}(\forall) = \mathbb{N}$ *and* $\mathfrak{A}(\boldsymbol{Q}) = P = \{n : n \text{ is prime}\}$. *Let* $s : \mathcal{V} \longrightarrow \mathbb{N}$ *where* $s(\boldsymbol{v_1}) = 23$. *Then* $\overline{s}(\boldsymbol{v_1}) = s(\boldsymbol{v_1}) \in P$. *So,* $\models_{\mathfrak{A}} \boldsymbol{Q}\boldsymbol{v_1}[s]$ *by definition. Now,* $\models_{\mathfrak{A}} \forall \boldsymbol{v_1} \boldsymbol{Q} \boldsymbol{v_1}[s]$ *if and only if* $\models_{\mathfrak{A}} \boldsymbol{Q}\boldsymbol{v_1}[s_{v_1|n}]$ *for all* $n \in \mathbb{N}$. *This statement holds if and only if* $\overline{s_{v_1|n}}(\boldsymbol{v_1}) = s_{v_1|n}(\boldsymbol{v_1}) = n \in Q^{\mathfrak{A}} = P$ *for all* $n \in \mathbb{N}$. *Take* $n = 4$ *and this statement is false. Thus,* $\not\models_{\mathfrak{A}} \forall \boldsymbol{v_1} \boldsymbol{Q} \boldsymbol{v_1}[s]$, *and hence* $\boldsymbol{Q}\boldsymbol{v_1} \nvDash \forall \boldsymbol{v_1} \boldsymbol{Q} \boldsymbol{v_1}$ *by definition. Intuitively, if a property holds for one element of the universe, then this fact does not imply that same property holds for every element in the universe.*

Notice that our last couple of examples involved wffs in which variables appeared free. As mentioned earlier in the chapter, the wffs that will be of the most interest to us are the sentence wffs (wffs in which no variables appear free). The following corollary to Theorem 4.27 will aid in proving results about logical implications involving sentences.

Corollary 4.31.1 (to Theorem 4.27**)** *For a sentence* $\boldsymbol{\sigma}$ *and a structure* $\mathfrak{S}$, *either*

(a) $\models_{\mathfrak{S}} \boldsymbol{\sigma}[s]$ *for every* $s : \mathcal{V} \longrightarrow \mathbb{U}$, *or*

(b) $\not\models_{\mathfrak{S}} \boldsymbol{\sigma}[s]$ *for every* $s : \mathcal{V} \longrightarrow \mathbb{U}$.

Proof: Since no variables occur free in $\boldsymbol{\sigma}$, $s\big|_{\mathcal{F}_\sigma} = s'\big|_{\mathcal{F}_\sigma}$ for all $s, s' : \mathcal{V} \longrightarrow \mathbb{U}$. Hence, by Theorem 4.35,

$$\models_{\mathfrak{S}} \boldsymbol{\sigma}[s] \text{ if and only if } \models_{\mathfrak{S}} \boldsymbol{\sigma}[s'].$$

So, if $\models_{\mathfrak{S}} \boldsymbol{\sigma}[s]$ or $\not\models_{\mathfrak{S}} \boldsymbol{\sigma}[s]$ for one particular function s, then this will be true for every function s. ∎

Since this corollary shows that the statement $\models_{\mathfrak{S}} \boldsymbol{\sigma}[s]$ is independent of our choice for s for sentence $\boldsymbol{\sigma}$, we write either $\models_{\mathfrak{S}} \boldsymbol{\sigma}$ or $\not\models_{\mathfrak{S}} \boldsymbol{\sigma}$. At long last, given this corollary we can make sense of a definition of truth and falsity in first-order languages.

Definition 4.32 *We say that a sentence* $\boldsymbol{\sigma}$ *is* **true** *in* $\mathfrak{S}$ *if* $\models_{\mathfrak{S}} \boldsymbol{\sigma}$*, and we say that* $\boldsymbol{\sigma}$ *is* **false** *in* $\mathfrak{S}$ *if* $\not\models_{\mathfrak{S}} \boldsymbol{\sigma}$*. In the case that* $\boldsymbol{\sigma}$ *is true in* $\mathfrak{S}$ *we say that* $\mathfrak{S}$ **models** $\boldsymbol{\sigma}$*.* $\mathfrak{S}$ *is said to model a set of sentences* Σ *if* $\mathfrak{S}$ *models every sentence in* Σ*.*

Example 4.33 *Corollary* 4.31.1 *and Example* 4.24 *demonstrate that the structure for number theory models the sentence*

$$\forall \boldsymbol{v_1}((\neg < \mathbf{0}\boldsymbol{v_1}) \to \approx \mathbf{0}\boldsymbol{v_1})$$

since there is no successor to zero (zero is the least natural number).

Corollary 4.33.1 (to Theorem 4.27**)** *For a set* $\Sigma \cup \{\boldsymbol{\tau}\}$ *of sentences,* $\Sigma \vDash \boldsymbol{\tau}$ *if and only if every model of* Σ *is also a model of* τ*.*

Example 4.34 *Again, consider a first-order language with at least a 1-place predicate symbol* $\boldsymbol{Q}$*. Then* $\forall \boldsymbol{v_1 Q v_1} \vDash \exists \boldsymbol{v_2 Q v_2}$*. Both wffs involved are sentences. Let* $\mathfrak{S}$ *be a structure which models* $\forall \boldsymbol{v_1 Q v_1}$*. Thus, there is a function*

$s : \mathcal{V} \longrightarrow \mathbb{U}$ such that $\models_{\mathfrak{S}} \forall \boldsymbol{v_1} \boldsymbol{Q} \boldsymbol{v_1}[s]$. This statement holds if and only if $\models_{\mathfrak{S}} \boldsymbol{Q}\boldsymbol{v_1}[s_{v_1|u}]$ for every $u \in \mathbb{U}$. This holds if and only if

$$\overline{s_{v_1|u}}(\boldsymbol{v_1}) = s_{v_1|u}(\boldsymbol{v_1}) = u \in Q^{\mathfrak{S}}$$

for all $u \in \mathbb{U}$. Since $\mathbb{U}$ is a non-empty set (by definition), there is $u' \in \mathbb{U}$. So,

$$\overline{s_{v_2|u'}}(\boldsymbol{v_2}) = s_{v_2|u'}(\boldsymbol{v_2}) = u' \in Q^{\mathfrak{S}}.$$

Thus, there is $u' \in \mathbb{U}$ such that $\models_{\mathfrak{S}} \boldsymbol{Q}\boldsymbol{v_2}[s_{v_2|u'}]$, but this is true if and only if $\models_{\mathfrak{S}} \exists \boldsymbol{v_2}\boldsymbol{Q}\boldsymbol{v_2}[s]$. Since $\exists \boldsymbol{v_2}\boldsymbol{Q}\boldsymbol{v_2}$ is a sentence satisfied by one $s : \mathcal{V} \longrightarrow \mathbb{U}$, $\mathfrak{S}$ must model $\exists \boldsymbol{v_2}\boldsymbol{Q}\boldsymbol{v_2}$. Since our choice for $\mathfrak{S}$ was arbitrary, every model of $\forall \boldsymbol{v_1}\boldsymbol{Q}\boldsymbol{v_1}$ is also a model of $\exists \boldsymbol{v_2}\boldsymbol{Q}\boldsymbol{v_2}$, and $\forall \boldsymbol{v_1}\boldsymbol{Q}\boldsymbol{v_1} \vDash \exists \boldsymbol{v_2}\boldsymbol{Q}\boldsymbol{v_2}$. Intuitively, if a property holds for every element in a non-empty set, that property holds for at least one element in the set. This statement, very easy to express in a first-order language, was a statement that gave our sentential language a great deal of trouble.

Example 4.35 *Once again, consider a first-order language with at least a 1-place predicate symbol $\boldsymbol{Q}$. Then, $\vDash \exists \boldsymbol{x}(\boldsymbol{Q}\boldsymbol{x} \rightarrow \forall \boldsymbol{x}\boldsymbol{Q}\boldsymbol{x})$. In other words, $\exists \boldsymbol{x}(\boldsymbol{Q}\boldsymbol{x} \rightarrow \forall \boldsymbol{x}\boldsymbol{Q}\boldsymbol{x})$ is a valid sentence. Let $\mathfrak{S}$ be any structure. We must show that this arbitrary structure models the sentence. Now, either $r \in Q^{\mathfrak{S}}$ for all $r \in \mathbb{U}$ or there is some $u \in \mathbb{U}$ for which $u \notin Q^{\mathfrak{S}}$ (this is a tautology at the meta-level). Since $\mathbb{U}$ is not empty, there is at least one function $s : \mathcal{V} \longrightarrow \mathbb{U}$. For this function, the above tautology holds, if and only if*

$$\overline{s_{x|r}}(\boldsymbol{x}) = s_{x|r}(\boldsymbol{x}) \in Q^{\mathfrak{S}} \textit{ for all } r \in \mathbb{U} \textit{ or}$$

$$\overline{s_{x|u}}(\boldsymbol{x}) = s_{x|u}(\boldsymbol{x}) \notin Q^{\mathfrak{S}} \textit{ for some } u \in \mathbb{U} \textit{ iff}$$

$$\models_{\mathfrak{S}} \boldsymbol{Q}\boldsymbol{x}[s_{x|r}] \textit{ for all } r \in \mathbb{U} \textit{ or } \not\models_{\mathfrak{S}} \boldsymbol{Q}\boldsymbol{x}[s_{x|u}] \textit{ for some } u \in \mathbb{U} \textit{ iff}$$

$$\models_{\mathfrak{S}} \forall \boldsymbol{x}\boldsymbol{Q}\boldsymbol{x}[s] \text{ or } \not\models_{\mathfrak{S}} \boldsymbol{Q}\boldsymbol{x}[s_{x|u}] \text{ for some } u \in \mathbb{U}$$

By Corollary 4.31.1, $\forall \boldsymbol{x}\boldsymbol{Q}\boldsymbol{x}$ *is a sentence satisfied in the structure* $\mathfrak{S}$ *by the function* s, *if and only if* $\forall \boldsymbol{x}\boldsymbol{Q}\boldsymbol{x}$ *is satisfied in* $\mathfrak{S}$ *by the function* $s_{x|u}$, *and so*

$$\models_{\mathfrak{S}} \forall \boldsymbol{x}\boldsymbol{Q}\boldsymbol{x}[s] \text{ or } \not\models_{\mathfrak{S}} \boldsymbol{Q}\boldsymbol{x}[s_{x|u}] \text{ for some } u \in \mathbb{U} \text{ iff}$$

$$\models_{\mathfrak{S}} \forall \boldsymbol{x}\boldsymbol{Q}\boldsymbol{x}[s_{x|u}] \text{ or } \not\models_{\mathfrak{S}} \boldsymbol{Q}\boldsymbol{x}[s_{x|u}] \text{ for some } u \in \mathbb{U} \text{ iff}$$

$$\models_{\mathfrak{S}} (\boldsymbol{Q}\boldsymbol{x} \rightarrow \forall \boldsymbol{x}\boldsymbol{Q}\boldsymbol{x})[s_{x|u}] \text{ for some } u \in \mathbb{U} \text{ iff}$$

$$\models_{\mathfrak{S}} \exists \boldsymbol{x}(\boldsymbol{Q}\boldsymbol{x} \rightarrow \forall \boldsymbol{x}\boldsymbol{Q}\boldsymbol{x})[s].$$

Hence $\mathfrak{S}$ *models* $\exists \boldsymbol{x}(\boldsymbol{Q}\boldsymbol{x} \rightarrow \forall \boldsymbol{x}\boldsymbol{Q}\boldsymbol{x})$, *and since our choice for the structure was arbitrary,* $\models \exists \boldsymbol{x}(\boldsymbol{Q}\boldsymbol{x} \rightarrow \forall \boldsymbol{x}\boldsymbol{Q}\boldsymbol{x})$.

This last example serves to illustrate the relationship between our meta-logic (the real world) and the mathematical model of logic we have constructed. Notice that we have used a meta-tautology and the meta-theorems we have derived concerning the formal mathematical structures of first-order languages to be able to say that $\exists \boldsymbol{x}(\boldsymbol{Q}\boldsymbol{x} \rightarrow \forall \boldsymbol{x}\boldsymbol{Q}\boldsymbol{x})$ is a valid sentence (the formal equivalent of a tautology). What we are doing is creating our formal mathematical system of logic, embedded within the real world, and we can reason at the meta-level (in the real world) about this system with meta-logic. Our goal is that by reasoning about formal logic as a model of our meta (real world) logic, our model might shed some light about how reasoning in the real world works and what properties it has. This project is very similar to modeling a projectile's path with mathematics. I can make some measurements in the real world and come up with a formal mathematical model for a projectile using a quadratic function. I can even reason at the formal level with this model to able to find nice features of the formal model (the vertex, x-intercepts, etc.).

The formal features of the model then shed some light on what is going on in the real world i.e. how high the rocket goes and when it is on the ground. The quadratic model for the rocket will of course be imperfect since some variables were unaccounted for, but even as a simplistic model it sheds light on the real world.

At this point we can shed only minimal light on the meta-world, but will be able to say more as the we develop more tools in the following chapters. Before leaving this chapter, however there are two particular topics that we wish to visit which will be key in the succeeding development of our tools of Soundness, Completeness, and Compactness and also our ultimate goal of Gödel's Incompleteness Theorem.

4.4 Definability of Relations Within Structures

We wish to discuss just how much structure our first order languages can express. Towards that end we develop a new term of measurement. Consider a fixed structure $\mathfrak{S}$ and a wff $\boldsymbol{\varphi}$ with free variables among $\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_k}$ and $u_1, u_2, \ldots, u_k \in \mathbb{U}$. We will say,

$$\models_{\mathfrak{S}} \boldsymbol{\varphi}[[u_1, u_2, \ldots, u_k]]$$

if and only if there is a function $s : \mathcal{V} \longrightarrow \mathbb{U}$ such that $s(\boldsymbol{v_i}) = u_i$ for each $1 \leq i \leq k$ and $\mathfrak{S}$ satisfies $\boldsymbol{\varphi}$ with this function. By Theorem 4.27, this statement holds if and only if $\models_{\mathfrak{S}} \boldsymbol{\varphi}[s]$ for *every* such function s. In the universe $\mathbb{U}$ determined by $\mathfrak{S}$ we may thus define the k-ary relation,

$$\{(u_1, u_2, \ldots, u_k) : \models_{\mathfrak{S}} \boldsymbol{\varphi}[[u_1, u_2, \ldots, u_k]]\}.$$

Given the structure $\mathfrak{S}$, this relation is completely determined by the the wff $\boldsymbol{\varphi}$.

Definition 4.36 *Given the structure* $\mathfrak{S}$ *and the the wff* $\boldsymbol{\varphi}$, *whose free variables are among* $\boldsymbol{v_1}$, $\boldsymbol{v_2}, \ldots,$ $\boldsymbol{v_k}$,

$$\{(u_1, u_2, \ldots, u_k) : \models_{\mathfrak{S}} \boldsymbol{\varphi}[[u_1, u_2, \ldots, u_k]]\}$$

is the **relation that** $\boldsymbol{\varphi}$ **defines** *in* $\mathfrak{S}$. *Given a* k*-ary relation* R *in the universe* $\mathbb{U}$ *determined by* $\mathfrak{S}$, *if there is a* $\boldsymbol{\varphi}$ *such that*

$$R = \{(u_1, u_2, \ldots, u_k) : \models_{\mathfrak{S}} \boldsymbol{\varphi}[[u_1, u_2, \ldots, u_k]]\},$$

then R *is said to be* **definable** *in* $\mathfrak{S}$.

Example 4.37 *Take the formal language that we have used to support the language of number theory and also the actual structure,* $\mathfrak{A}$, *of number theory. Consider the set* $\{n \in \mathbb{N} : \ n \textit{ is odd } \}$. *We claim that*

$$\boldsymbol{\varphi} = \exists \boldsymbol{v_2}(\boldsymbol{S}(\boldsymbol{S}(\mathbf{0})) \cdot \boldsymbol{v_2} + \boldsymbol{S}(\mathbf{0}) \approx \boldsymbol{v_1})$$

defines the set of odd numbers in the structure of number theory. To see this, notice that

$$\models_{\mathfrak{A}} \boldsymbol{\varphi}[[n]] \textit{ iff}$$

$$\textit{There is } s : \mathcal{V} \longrightarrow \mathbb{N} \textit{ with } s(\boldsymbol{v_1}) = n \textit{ such that } \models_{\mathfrak{A}} \boldsymbol{\varphi}[s] \textit{ iff}$$

$$\textit{There is } m \in \mathbb{N} \textit{ such that } \models_{\mathfrak{A}} \boldsymbol{S}(\boldsymbol{S}(\mathbf{0})) \cdot \boldsymbol{v_2} + \boldsymbol{S}(\mathbf{0}) \approx \boldsymbol{v_1}[s_{v_2|m}] \textit{ iff}$$

$$\textit{There is } m \in \mathbb{N} \textit{ such that } \overline{s_{v_2|m}}(\boldsymbol{S}(\boldsymbol{S}(\mathbf{0})) \cdot \boldsymbol{v_2} + \boldsymbol{S}(\mathbf{0})) = \overline{s_{v_2|m}}(\boldsymbol{v_1}) \textit{ iff}$$

$$\textit{There is } m \in \mathbb{N} \textit{ such that } \overline{s_{v_2|m}}(\boldsymbol{S}(\boldsymbol{S}(\mathbf{0})) \cdot \boldsymbol{v_2}) + \overline{s_{v_2|m}}(\boldsymbol{S}(\mathbf{0})) = s_{v_2|m}(\boldsymbol{v_1}) \textit{ iff}$$

$$\textit{There is } m \in \mathbb{N} \textit{ such that } \overline{s_{v_2|m}}(\boldsymbol{S}(\boldsymbol{S}(\mathbf{0}))) \cdot \overline{s_{v_2|m}}(\boldsymbol{v_2}) + S(\overline{s_{v_2|m}}(\mathbf{0})) = s(\boldsymbol{v_1}) \textit{ iff}$$

$$\textit{There is } m \in \mathbb{N} \textit{ such that } S(S(\overline{s_{v_2|m}}(\mathbf{0}))) \cdot s_{v_2|m}(\boldsymbol{v_2}) + S(0) = n \textit{ iff}$$

$$\textit{There is } m \in \mathbb{N} \textit{ such that } S(S(0)) \cdot m + 1 = n \textit{ iff}$$

There is $m \in \mathbb{N}$ *such that* $2 \cdot m + 1 = n$ *iff*

n *is odd.*

So, $\{n \in \mathbb{N} : \ n \text{ is odd } \} = \{n \in \mathbb{N} : \models_{\mathfrak{A}} \boldsymbol{\varphi}[[n]]\}$*, and* $\boldsymbol{\varphi}$ *defines the set of odd numbers in* $\mathbb{N}$*.*

With this example, we can see how definability gives a measure of expressibility of a formal language, for, we can always start with some relation in our concrete structure (number theory for instance) and then "check" to see whether in our formal language there is a formula that defines that relation. If a relation is definable in the formal language, then our formal language is adequate to support that particular meta-structure. If no formula of our formal language defines a relation, then our language falls short in its expressibility of the meta-structure that actually exists in a bona fide relation (or we may say perhaps that our language is "incomplete"...). The question becomes whether there actually are any such relations that exist and yet are undefinable by a formal language. This question is the heart of the incompleteness of Gödel's Incompleteness Theorem. In a general sense, we may say that there must always be relations that are undefinable for a formal language with a countable alphabet, for, there are only countably many ($\aleph_0$) possible formulas in such a language, but uncountably many relations on even a countable set. The point of Gödel's Incompleteness Theorem is to demonstrate one such relation for a language that can support number theory.

Definability will clearly be a key tool in what follows. Another necessary tool we discuss in the next section.

4.5 Elementary Equivalence and Homomorphisms

Having defined structures, we wish to see when two structures are similar or the same. One measure of sameness of structure is *elementary equivalence.*

Definition 4.38 *For a first-order language, two structures of the language,* $\mathfrak{A}$ *and* $\mathfrak{B}$*, are* **elementary equivalent** *if for any sentence* $\boldsymbol{\sigma}$ *in the language*

$$\models_{\mathfrak{A}} \boldsymbol{\sigma} \text{ if and only if } \models_{\mathfrak{B}} \boldsymbol{\sigma}$$

and we say $\mathfrak{A} \equiv \mathfrak{B}$*.*

Intuitively, this definition says that every formal sentence has the same truth value in both structures. Every property that can be expressed by first-order sentences is shared between the two structures. It is difficult at this stage with our limited tools to be able to show elementary equivalence, so we give one example with some hand-wavy justification.

Example 4.39 *Consider a first order language with only one 2-place predicate symbol* $<$ *and equality. Let the structure* $\mathfrak{A}$ *be such that* $\mathfrak{A}(\forall) = \mathbb{Q}$ *(the rational numbers) and* $\mathfrak{A}(<) = \{(a, b) : a < b\}$*. Let the structure* $\mathfrak{B}$ *be such that* $\mathfrak{B}(\forall) = \mathbb{R}$ *(the real numbers) and* $\mathfrak{B}(<) = \{(a, b) : a < b\}$*. Then* $\mathfrak{A} \equiv \mathfrak{B}$*. We don't give a fully rigorous justification here, but, roughly, every ordering statement that can be stated using first-order sentences about the rational numbers can also be stated about the real numbers and vice-versa. So, for instance, take the sentence*

$$\forall \boldsymbol{v_1} \forall \boldsymbol{v_2} \exists \boldsymbol{v_3} (\boldsymbol{v_1} < \boldsymbol{v_2} \rightarrow (\boldsymbol{v_1} < \boldsymbol{v_3} \wedge \boldsymbol{v_3} < \boldsymbol{v_2})).$$

Each structure is a model of this sentence since between each two rational numbers there is another rational number, and the same holds for the real numbers.

That each first-order sentence of the language should have the same truth value in both structures makes sense since the reals are constructed from the rationals by the Completeness Axiom which states that every subset of the real numbers that has an upper bound has a least upper bound. This statement is false for the rational numbers. For instance, take the set $\{3, 3.1, 3.14, 3.141, 3.1415, \ldots\}$*. This is a set of rational approximations to* π*. This set is bounded above and below (by 3 and 4), but there is no least upper bound for the set since for any proposed upper bound (say, 3.15), a smaller rational upper bound may be found (3.142). The Completeness Axiom ensures that there is a least upper bound for this set in the real numbers, and* π *is this least upper bound (of course* π *is not a rational number). So, the Completeness Axiom is the essential difference (in terms of ordering) between the rational and real numbers and is a statement false in the rational numbers but true in the real numbers.*

However, the Completeness Axiom is not a first-order logical statement within the formal language that we have at our disposal. Notice that the variable in the statement of the Completeness Axiom are sets whereas for our structures $\mathfrak{A}$ *and* $\mathfrak{B}$*, our variables are mapped to either rational or real numbers. There is nothing that we have developed so far that would allow us to work in the same universe and yet talk about both ranging over variables and ranging over sets containing elements within that universe as well. This need is fulfilled in second-order logic (not treated in this thesis). So, it makes sense that these two structures should be elementary equivalent since their main difference* *comes from a statement that cannot be expressed in a first-order language.*

The above example demonstrates that elementary equivalence gives us a measure of sameness of structure at the first-order level, but there may be properties, even significant properties, of universes that cannot be captured by first-order sentences, and hence our measure of elementary equivalence fails us at this point.

A more comprehensive and general measure comes with structure preserving maps, or, homomorphisms. Homomorphisms show up during the study of mid to upper division mathematics in a variety of contexts such as linear algebra, abstract algebra, ring theory, and field theory. Here, we may state a more overarching and generic definition of what a homomorphism is.

Definition 4.40 *Given a first-order language and structures $\mathfrak{A}$ and $\mathfrak{B}$ for the language, a* **homomorphism** *of $\mathfrak{A}$ into $\mathfrak{B}$ is a map from $\mathbb{A}$ into $\mathbb{B}$ (the universes for $\mathfrak{A}$ and $\mathfrak{B}$ respectively) fulfilling the following requirements:*

(i) For each n-place predicate symbol P in the formal language,

$$(a_1, a_2, \ldots, a_n) \in P^{\mathfrak{A}} \text{ if and only if } (h(a_1), h(a_2), \ldots, h(a_n)) \in P^{\mathfrak{B}}$$

(ii) For each n-place function symbol f,

$$h(f^{\mathfrak{A}}(a_1, a_2, \ldots, a_n)) = f^{\mathfrak{B}}(h(a_1), h(a_2), \ldots, h(a_n))$$

(iii) For each constant symbol c,

$$h(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$$

If h is one-to-one and onto, h is said to be an **isomorphism***, and $\mathfrak{A}$ and $\mathfrak{B}$ are said to be* **isomorphic** *($\mathfrak{A} \simeq \mathfrak{B}$). An* **automorphism** *is a one-to-one and onto map of the universe of a structure into itself.*

Example 4.41 *Consider a first order language with equality, one 2-place predicate symbol,* $\boldsymbol{P}$*, one-two place function symbol* $\boldsymbol{f}$*, and one constant symbol* $\boldsymbol{e}$*. Consider the structure* $\mathfrak{A}$ *where* $\forall \longmapsto \mathbb{R}$, $\boldsymbol{P} \longmapsto <$, $\boldsymbol{f} \longmapsto +$, *and* $\boldsymbol{e} \longmapsto 0$ *and the structure* $\mathfrak{B}$ *where* $\forall \longmapsto \mathbb{R}^+$, $\boldsymbol{P} \longmapsto <$, $\boldsymbol{f} \longmapsto \cdot$, *and* $\boldsymbol{e} \longmapsto 1$. *Let* $exp : \mathbb{R} \longrightarrow \mathbb{R}^+$ *be the standard exponential function. The mapping is one-to-one and onto since* $ln : \mathbb{R}^+ \longrightarrow \mathbb{R}$ *such that* $exp \circ ln = I = ln \circ exp$ *where* I *is the identity map. Notice that since exp is an increasing function if* $a, b \in \mathbb{R}$ *with* $a < b$, $exp(a) < exp(b)$ *fulfilling property (i) of a homomorphism. Note that* $exp(a + b) = exp(a) \cdot exp(b)$*, fulfilling property (ii) of a homomorphism, and also note that* $exp(0) = 1$ *fulfilling property (iii) of a homomorphism. So, exp is a one-to-one and onto homomorphism, hence an isomorphism for* $\mathfrak{A}$ *and* $\mathfrak{B}$*, and hence the structure* $\mathfrak{A}$ *is isomorphic to the structure* $\mathfrak{B}$*. For all intents and purposes, except for the exterior "skin" of symbology, there is absolutely no difference between structures* $\mathfrak{A}$ *and* $\mathfrak{B}$*. Their universes have the same cardinality (since exp is one-to-one and onto) and functions and relations imposed by* $\mathfrak{A}$ *and* $\mathfrak{B}$ *are preserved.*

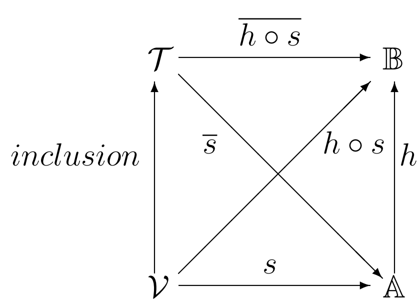
This example is of course a classic example of a group homomorphism between the groups $\langle \mathbb{R}, + \rangle$ *and* $\langle \mathbb{R}^+, \cdot \rangle$*. However, note that our language is not the language of groups per se since there is not the one-place membership predicate as we defined the language for groups to include and there is instead a 2-place predicate symbol. So, the notion of homomorphism can be extended to the preservation of very generic structures.*

Example 4.42 *Consider the structures* $\mathfrak{A}$ *and* $\mathfrak{B}$ *as defined in Example* 4.41. *Take the function* $I : \mathbb{Q} \longrightarrow \mathbb{R}$ *where* $r \longmapsto r$ *(this would be the identity function apart the mismatch between the domain and the codomain). The function is of course one-to-one but not onto because the domain is countable* *whereas the codomain is uncountable. Clearly, the ordering between the two sets will be preserved, and so I is a homomorphism for the two structures.*

The last example serves to illustrate a couple of things. The first is that whereas elementary equivalence could not measure a significant difference between these two structures, the above homomorphism can via the concept of onto. So, a homomorphism is a more powerful measure than elementary equivalence. The second is that since the map is one-to-one and preserves the ordering and since $\mathbb{Q} \subseteq \mathbb{R}$ we may think of $\mathfrak{A}$ as a substructure of $\mathfrak{B}$, a useful concept to play with. To relate the two measures, we have the following theorem.

Theorem 4.43 (The Homomorphism Theorem) *Let h be a homomorphism of $\mathfrak{A}$ into $\mathfrak{B}$, and let s map the set of variables of the given first-order language into $\mathfrak{A}$. Then:*

(i) For any term t, $h(\overline{s}(t)) = \overline{h \circ s}(t)$ (Note that $\overline{s}(t)$ is computed in $\mathbb{A}$ whereas $\overline{h \circ s}(t)$ is computed in $\mathbb{B}$ as shown in the following diagram)



(ii) For any quantifier-free formula $\boldsymbol{\varphi}$ not containing the equality symbol,

$$\models_{\mathfrak{A}} \boldsymbol{\varphi}[s] \text{ if and only if } \models_{\mathfrak{B}} \boldsymbol{\varphi}[h \circ s]$$

(iii) If h is one-to-one, then part (ii) holds even if the equality symbol is present in $\boldsymbol{\varphi}$.

(iv) If h is onto, then part (ii) holds even if $\boldsymbol{\varphi}$ has a quantifier symbol.

Proof: We use an inductive argument with the set

$$S = \{\boldsymbol{t} \in \mathcal{T} : h \circ \overline{s}(\boldsymbol{t}) = \overline{h \circ s}(\boldsymbol{t})\}.$$

For a variable symbol $\boldsymbol{v}$ in the language $\overline{s}(\boldsymbol{v}) = s(\boldsymbol{v})$, and $\overline{h \circ s}(\boldsymbol{v}) = h \circ s(\boldsymbol{v})$, so that all variables must be in S. For a constant symbol c in the language, $\overline{s}(\boldsymbol{c}) = c^{\mathfrak{A}}$, and $\overline{h \circ s}(\boldsymbol{c}) = c^{\mathfrak{B}}$ since $h \circ s : \mathcal{V} \longrightarrow \mathbb{B}$. Since $h(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$, all constant symbols in the given first order language will be in the set.

For our induction hypothesis, let us be given an n-place function symbol $\boldsymbol{f}$, and let us assume that $\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_n} \in S$. Then, $\overline{s}(\boldsymbol{f t_1 t_2 \cdots t_n}) = f^{\mathfrak{A}}(\overline{s}(\boldsymbol{t_1}), \overline{s}(\boldsymbol{t_2}), \ldots, \overline{s}(\boldsymbol{t_n}))$ So,

$$\begin{aligned} h(\overline{s}(\boldsymbol{f t_1 t_2 \cdots t_n})) &= f^{\mathfrak{B}}(h(\overline{s}(\boldsymbol{t_1})), h(\overline{s}(\boldsymbol{t_2})), \ldots, h(\overline{s}(\boldsymbol{t_n}))) \\ &= f^{\mathfrak{B}}(\overline{h \circ s}(\boldsymbol{t_1}), \overline{h \circ s}(\boldsymbol{t_2}), \ldots, \overline{h \circ s}(\boldsymbol{t_n})) \\ &= \overline{h \circ s}(\boldsymbol{f t_1 t_2 \cdots t_n}) \end{aligned}$$

since h is a homomorphism, $\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_n} \in S$, and by results previously established about extensions of variable assignments. Hence $\boldsymbol{f t_1 t_2 \cdots t_n} \in S$ and $S = \mathcal{T}$ by the Induction Principle. This proves part (i).

For part (ii) if $\boldsymbol{\varphi} = \boldsymbol{P t_1 t_2 \cdots t_n}$, then

$$\models_{\mathfrak{A}} \boldsymbol{\varphi}[s] \text{ if and only if } (\overline{s}(\boldsymbol{t_1}), \overline{s}(\boldsymbol{t_2}), \ldots, \overline{s}(\boldsymbol{t_n})) \in P^{\mathfrak{A}} \text{ (by definition) iff}$$

$$(h(\overline{s}(\boldsymbol{t_1})), h(\overline{s}(\boldsymbol{t_2})), \ldots, h(\overline{s}(\boldsymbol{t_n}))) \in P^{\mathfrak{B}} \text{ since } h \text{ is a homomorphism, iff}$$

$$(\overline{h \circ s}(\boldsymbol{t_1}), \overline{h \circ s}(\boldsymbol{t_2}), \ldots, \overline{h \circ s}(\boldsymbol{t_n})) \in P^{\mathfrak{B}} \text{ (by part (i)) iff}$$

$$\models_{\mathfrak{B}} \boldsymbol{\varphi}[h \circ s].$$

A simple induction argument suffices to show that the same statement holds if $\boldsymbol{\varphi}$ involves the logical connectives $\neg$ or $\rightarrow$.

Now, we may say that

$$\models_{\mathfrak{A}} \approx \boldsymbol{t_1 t_2}\ [s] \text{ if and only if } \overline{s}(\boldsymbol{t_1}) = \overline{s}(\boldsymbol{t_2})$$

by definition. Of course this statement implies that $h(\overline{s}(\boldsymbol{t_1})) = h(\overline{s}(\boldsymbol{t_2}))$, and the statements are equivalent as long as h is one-to-one. By part (i),

$$h(\overline{s}(\boldsymbol{t_1})) = h(\overline{s}(\boldsymbol{t_2})) \text{ if and only if } \overline{h \circ s}(\boldsymbol{t_1}) = \overline{h \circ s}(\boldsymbol{t_2}) \text{ iff}$$

$$\models_{\mathfrak{B}} \approx \boldsymbol{t_1 t_2}\ [h \circ s].$$

For part (iv), we assume that h is onto. Suppose

$$\models_{\mathfrak{B}} \boldsymbol{\varphi}[h \circ s] \text{ if and only if } \models_{\mathfrak{A}} \boldsymbol{\varphi}[s]$$

for every $s : \mathcal{V} \longrightarrow \mathbb{A}$. We wish to demonstrate that

$$\models_{\mathfrak{B}} \forall \boldsymbol{x}\boldsymbol{\varphi}[h \circ s] \text{ if and only if } \models_{\mathfrak{A}} \forall \boldsymbol{x}\boldsymbol{\varphi}[s]$$

. Now

$$\models_{\mathfrak{B}} \forall \boldsymbol{x}\boldsymbol{\varphi}[h \circ s] \text{ iff}$$

$$\text{For all } b \in \mathbb{B},\ \models_{\mathfrak{B}} \boldsymbol{\varphi}[(h \circ s)_{x|b}] \text{ iff}$$

$$\text{For all } a \in \mathbb{A},\ \models_{\mathfrak{B}} \boldsymbol{\varphi}[(h \circ s)_{x|h(a)}] \text{ since } h \text{ is an onto mapping.}$$

Now,

$$(h \circ s)_{x|h(a)}(\boldsymbol{y}) = \begin{cases} h(a) & \text{if } \boldsymbol{x} = \boldsymbol{y} \\ h(s(\boldsymbol{y})) & \text{if } \boldsymbol{x} \neq \boldsymbol{y} \end{cases}$$

Since

$$s_{x|a}(\boldsymbol{y}) = \begin{cases} a & \text{if } \boldsymbol{x} = \boldsymbol{y} \\ s(\boldsymbol{y}) & \text{if } \boldsymbol{x} \neq \boldsymbol{y} \end{cases}$$

it is clear that $(h \circ s)_{x|h(a)} = h \circ (s_{x|a})$ Thus,

$$\text{For all } a \in \mathbb{A},\ \models_{\mathfrak{B}} \boldsymbol{\varphi}[(h \circ s)_{x|h(a)}] \text{iff}$$

$$\text{For all } a \in \mathbb{A},\ \models_{\mathfrak{B}} \boldsymbol{\varphi}[h \circ (s_{x|a})] \text{ iff}$$

$$\text{For all } a \in \mathbb{A},\ \models_{\mathfrak{A}} \boldsymbol{\varphi}[s_{x|a}] \text{ (by hypothesis) iff}$$

$$\models_{\mathfrak{A}} \forall \boldsymbol{x}\boldsymbol{\varphi}[s].$$

Hence, all four parts of the theorem are proved. ∎

This theorem gives the relationship between our two measures of sameness of structure. Note in particular that if h is an isomorphism, that is, $\mathfrak{A}$ and $\mathfrak{B}$ are isomorphic, that sentences have the same truth value in both structures. That is, isomorphic structures are elementary equivalent.

Also, isomorphisms preserve definable relations. Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures and let R be an n-ary relation in $\mathbb{A}$. Let h be a homomorphism of $\mathfrak{A}$ into $\mathfrak{B}$. Then denote $h(R) = \{(h(a_1), h(a_2), \ldots, h(a_n)) : (a_1, a_2, \ldots, a_n) \in R\}$.

Corollary 4.43.1 *If h is an isomorphism between $\mathfrak{A}$ and $\mathfrak{B}$, and R an n-ary definable relation in $\mathbb{A}$, then $h(R)$ is a definable relation in $\mathbb{B}$.*

Proof: Since R is definable, there is a wff $\boldsymbol{\varphi}$ such that

$$R = \{(a_1, a_2, \ldots, a_n) : \models_{\mathfrak{A}} \boldsymbol{\varphi}[[a_1, a_2, \ldots, a_n]]\}.$$

We will demonstrate that $h(R)$ is definable by the same formula $\boldsymbol{\varphi}$. Note that $\models_{\mathfrak{A}} \boldsymbol{\varphi}[[a_1, a_2, \ldots, a_n]]$ is true if and only if there is $s : \mathcal{V} \longrightarrow \mathbb{A}$ where $s(\boldsymbol{v_i}) = a_i$ for $1 \leq i \leq n$ (recall that in this scenario $\boldsymbol{v_i}$ for $1 \leq i \leq n$ are assumed to account for all of the free variables in $\boldsymbol{\varphi}$) where $\models_{\mathfrak{A}} \boldsymbol{\varphi}[s]$. By the Homomorphism Theorem, this is so if and only if $\models_{\mathfrak{B}} \boldsymbol{\varphi}[h \circ s]$ where $h \circ s : \mathcal{V} \longrightarrow \mathbb{B}$ such that $(h \circ s)(\boldsymbol{v_i}) = h(a_i)$ for $1 \leq i \leq n$. By definition, this is so if and only if $\models_{\mathfrak{B}} \boldsymbol{\varphi}[[h(a_1), h(a_2), \ldots, h(a_n)]]$. Thus,

$$\begin{aligned} h(R) &= \{(h(a_1), h(a_2), \ldots, h(a_n)) : (a_1, a_2, \ldots, a_n) \in R\} \\ &= \{(h(a_1), h(a_2), \ldots, h(a_n)) : \models_{\mathfrak{B}} \boldsymbol{\varphi}[[h(a_1), h(a_2), \ldots, h(a_n)]]\} \end{aligned}.$$

Thus, $h(R)$ is definable by $\boldsymbol{\varphi}$ by definition. ∎

Corollary 4.43.2 *If h is an automorphism of $\mathfrak{A}$ and R is a definable relation, then $h(R) = R$.*

Proof: By the last corollary,

$$h(R) = \{(h(a_1), h(a_2), \ldots, h(a_n)) : \models_{\mathfrak{A}} \boldsymbol{\varphi}[[h(a_1), h(a_2), \ldots, h(a_n)]]\}.$$

The last set is a subset of R by definition, and since h is one-to-one and onto, we see at once that $h(R) = R$. ∎

Example 4.44 *Take the first-order language used in Example 4.41. Let $h : \mathbb{R} \longrightarrow \mathbb{R}$ with $h(x) = x^3$. This mapping is one-to-one and onto and preserves the standard ordering of the real numbers and is hence an automorphism for the ordering structure under consideration. If $\mathbb{Q}$ is definable in this ordering structure, then $h(\mathbb{Q}) = \mathbb{Q}$. Of course $h(\mathbb{Q}) \subseteq \mathbb{Q}$. Now, take $3 \in \mathbb{Q}$. If $h(\mathbb{Q}) = \mathbb{Q}$, then there is a rational number q such that $h(q) = 3$. However, note that $h(3^{1/3}) = 3$ and since h is one-to-one and onto $\mathbb{R}$, $3^{1/3}$ is the unique pre-image of 3, and $3^{1/3}$ is not rational. In fact then $h(\mathbb{Q}) \subset \mathbb{Q}$, and by the last corollary $\mathbb{Q}$ is not definable in the ordering structure for $\mathbb{R}$.*

As mentioned above, our results in this section will be key tools for what follows in the succeeding chapters.

We have in this chapter developed a new model for humanity's deductive thought processes. In the next chapters, we discuss the first-order model for proofs, we begin analyzing what nice properties our new model has, and we prepare for the statement and proof of Gödel's Incompleteness Theorem.

Chapter 5

First-Order Deductions

As with our discussion of Sentential Logic, we want to examine what proofs, or deductions, look like in our model. After all, this is what our aim has been. We are mathematizing real-world first-order logic so that we have all of the rigor and machinery of mathematics to be able to shed light on the our real-world deductions.

Now, our intuition is that a logical deduction occurs when we start from statements we already know to be true and then use the rules of logic finitely many times (being finite creatures)to make new statements, that is, new facts and theorems. So, we will mathematize our intuition, and then prove (at the meta-level) facts about our model of provability within our formal first-order system. To avoid confusion, we will call our formal language "proofs", deductions, and we will continue to call our meta-level "proofs", proofs.

5.1 Formalizing our Intuition

To model deducibility, we will have three components. First we will have a set Λ of formulas (discussed below), the same for all first-order languages. These axioms reflect the logical properties that we assume to be true when we reason in the real world. Next, we will have a set of formulas Γ where these formulas are specific to the particular first-order language under consideration. This set mimics our set of statements that we have already shown to be true or are assuming to be true and is the set of formulas from which we hope to obtain our new statements (our theorems). Finally, we will have a rule of inference which will allow us to formally generate our theorems from our collection of logical axioms and assumed statements $\Gamma \cup \Lambda$. Let $\mathcal{W}$ be the set of wffs in our particular language under consideration. Define $\mathcal{I} : \mathcal{W} \times \mathcal{W} \longrightarrow \mathcal{W}$ as follows:

$$\mathcal{I}(\boldsymbol{\alpha}, \boldsymbol{\varphi}) = \begin{cases} \boldsymbol{\beta} & \text{if } \boldsymbol{\varphi} = (\boldsymbol{\alpha} \to \boldsymbol{\beta}) \\ \boldsymbol{\varphi} & \text{otherwise} \end{cases}$$

This function will be our rule of inference and reflects our real-world inference that if it is true that statement A implies statement B, and if A is true, then we know that statement B must be true. This inference rule is known as *modus ponens*. Notice that if our function is restricted to $\Gamma \cup \Lambda$, we will only potentially generate a new formula that we do not already have if we have a modus ponens pair.

We now have all the basic elements to begin talking about theorems and deductions. In this context, we will call a set of wffs Δ inductive (the same "inductive" as in Chapter 2) if and only if $\Gamma \cup \Lambda \subseteq \Delta$ and Δ is closed under $\mathcal{I}$.

Definition 5.1 *The set of* **theorems** *of* Γ *is the smallest inductive set* Δ *(i.e. the intersection of all inductive sets). A* **theorem** *of* Γ *is an element* $\boldsymbol{\tau} \in \Delta$*, and we write* $\Gamma \vdash \boldsymbol{\tau}$

Emphasising the constructive nature of proving a theorem, we have the following definition for deduction.

Definition 5.2 *A* **deduction** *of $\boldsymbol{\varphi}$ from Γ is a sequence $\langle \boldsymbol{\alpha_0}, \ldots, \boldsymbol{\alpha_n} \rangle$ of formulas such that $\boldsymbol{\alpha_n} = \boldsymbol{\varphi}$ and for each $i \leq n$ either*

(i) $\boldsymbol{\alpha_i} \in \Gamma \cup \Lambda$ *or*

(ii) For some $j < k < i$, $\boldsymbol{\alpha_i} = \mathcal{I}(\boldsymbol{\alpha_j}, \boldsymbol{\alpha_k})$ (i.e. $\boldsymbol{\alpha_k} = (\boldsymbol{\alpha_j} \rightarrow \boldsymbol{\alpha_i})$)

A deduction and a theorem are actually equivalent concepts given the discussion in Chapter 2 about induction and the sets C_* and C^*.

Theorem 5.3 $\Gamma \vdash \boldsymbol{\tau}$ *if and only if there is a deduction of $\boldsymbol{\tau}$ from Γ.*

Proof: This is a specific case of Theorem 2.16. ∎

Example 5.4 *Suppose (for the time being) that $\Lambda = \{\boldsymbol{\alpha_1}, \boldsymbol{\alpha_2}, \ldots\}$ and that $\Gamma = \{\boldsymbol{\varphi}, (\boldsymbol{\varphi} \rightarrow \forall \boldsymbol{x}(\neg \boldsymbol{\psi})), (\forall \boldsymbol{x}(\neg \boldsymbol{\psi}) \rightarrow \boldsymbol{\chi})\}$. Then,*

$$\langle \boldsymbol{\varphi} \rangle, \langle (\boldsymbol{\varphi} \rightarrow \forall \boldsymbol{x}(\neg \boldsymbol{\psi})), \boldsymbol{\varphi}, \forall \boldsymbol{x}(\neg \boldsymbol{\psi}) \rangle, \text{ and}$$

$$\langle (\boldsymbol{\varphi} \rightarrow \forall \boldsymbol{x}(\neg \boldsymbol{\psi})), \boldsymbol{\varphi}, \forall \boldsymbol{x}(\neg \boldsymbol{\psi}), (\forall \boldsymbol{x}(\neg \boldsymbol{\psi}) \rightarrow \boldsymbol{\chi}), \boldsymbol{\chi} \rangle$$

are all deductions for $\boldsymbol{\varphi}$, $\forall \boldsymbol{x}(\neg \boldsymbol{\psi})$, and $\boldsymbol{\chi}$ respectively. So, $\boldsymbol{\varphi}$, $\forall \boldsymbol{x}(\neg \boldsymbol{\psi})$, and $\boldsymbol{\chi}$ are all theorems of Γ.

So, now we have a basic description of the formal model for deductions in first-order logic. However, since the set Λ is a set of logical axioms that is supposed to be used in deductions for *all* first-order languages, we must specify exactly what this set contains. We catagorize the members of this set into 6 main groups.

5.1.1 Tautologies

For this group, which we will sometimes refer to as (logical) axiom group 1, we will essentially be replacing the sentence symbols in sentential tautologies with first-order wffs. First, call atomic formulas and formulas of the form $\forall \boldsymbol{x}\boldsymbol{\alpha}$ prime formulas. All other formulas are of the form $(\boldsymbol{\alpha} \to \boldsymbol{\beta})$ or $(\neg\boldsymbol{\alpha})$. All first-order formulas are then generated from the prime formulas by the formula building operations $\mathcal{F}_\neg$ and $\mathcal{F}_\to$. So, returning to the ideas of sentential logic, if we let our sentence symbols be the prime formulas, the entire development of sentential logic is the same just with prime formulas treated like indecomposable chunks (as far as the sentential development is concerned). We will call these tautologies in this sentential rehash, first-order tautologies.

Example 5.5 *Consider the formula* $((\boldsymbol{Px} \to \boldsymbol{Qy}) \to ((\neg\boldsymbol{Qy}) \to (\neg\boldsymbol{Px})))$. *This is a tautology in the first-order sense, for*

$\boldsymbol{Px}$	$\boldsymbol{Qy}$	$(\neg\boldsymbol{Px})$	$(\neg\boldsymbol{Qy})$	$(\boldsymbol{Px} \to \boldsymbol{Qy})$	$((\neg\boldsymbol{Qy}) \to (\neg\boldsymbol{Px}))$	$((\boldsymbol{Px} \to \boldsymbol{Qy}) \to ((\neg\boldsymbol{Qy}) \to (\neg\boldsymbol{Px})))$
T	T	F	F	T	T	T
T	F	F	T	F	F	T
F	T	T	F	T	T	T
F	F	T	T	T	T	T

This connection of sentential logic with first-order logic leads to a useful theorem relating our sentential model of deduction and our first-order model of deduction.

Theorem 5.6 $\Gamma \vdash \boldsymbol{\tau}$ *if and only if* $\Gamma \cup \Lambda$ *tautologically implies* $\boldsymbol{\tau}$.

Proof: In any case, it is clear that $\{\boldsymbol{\alpha}, (\boldsymbol{\alpha} \to \boldsymbol{\beta})\}$ tautologically implies $\boldsymbol{\beta}$. Suppose that $\Gamma \vdash \boldsymbol{\tau}$ and let v be a truth assignment satisfying all of the formulas of $\Gamma \cup \Lambda$ where satisfaction is the satisfaction of sentential logic (illustrated in the last example). Since $\Gamma \vdash \boldsymbol{\tau}$, there is a deduction of $\boldsymbol{\tau}$ from Γ, and by induction, all components of the deduction will be satisfied by v given the tautological implication that we noted at the beginning of the proof. Hence, $\boldsymbol{\tau}$ will be satisfied by v, and $\Gamma \cup \Lambda$ tautologically implies $\boldsymbol{\tau}$ by definition.

Now suppose that $\Gamma \cup \Lambda$ tautologically implies $\boldsymbol{\tau}$. By Corollary 3.2.1 (a corollary to the Compactness Theorem for sentential logic), there exists a finite subset of $\Gamma \cup \Lambda$ such that this finite subset, say, $\{\boldsymbol{\gamma_1}, \boldsymbol{\gamma_2}, \ldots, \boldsymbol{\gamma_m}, \boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}, \ldots, \boldsymbol{\lambda_n}\}$ tautologically implies $\boldsymbol{\tau}$. By repeated application of Corollary 2.46.1 (and suppressing the use of some parentheses), we have that

$$\boldsymbol{\gamma_1} \rightarrow \boldsymbol{\gamma_2} \rightarrow \cdots \rightarrow \boldsymbol{\gamma_m} \rightarrow \boldsymbol{\lambda_1} \rightarrow \boldsymbol{\lambda_2} \rightarrow \cdots \rightarrow \boldsymbol{\lambda_n} \rightarrow \boldsymbol{\tau}$$

is a tautology and is thus by definition in Λ. So, applying our rule of inference, I, $m + n$ times to the appropriate elements in $\Gamma \cup \Lambda$, we see that there must be a deduction of $\boldsymbol{\tau}$ from Γ. Hence, the theorem holds. ∎

This theorem gives us a powerful tool in our arsenal for showing when a formula is a theorem of a particular set.

5.1.2 Substitution

We wish to add a new type of rule to our set of logical axioms Λ, and we will call this group axiom group 2. First, we will say that $\boldsymbol{\varphi_{x|t}}$ is the formula $\boldsymbol{\varphi}$ where the variable $\boldsymbol{x}$ is replaced with the term $\boldsymbol{t}$ wherever $\boldsymbol{x}$ occurs in $\boldsymbol{\varphi}$. The rules for each type of formula are as follows

(i) For atomic $\boldsymbol{\varphi}$, $\boldsymbol{\varphi_{x|t}}$ is the formula obtained by replacing $\boldsymbol{x}$ with the term $\boldsymbol{t}$.

(ii) $(\neg\boldsymbol{\varphi})_{\boldsymbol{x}|\boldsymbol{t}} = (\neg(\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}}))$

(iii) $(\boldsymbol{\varphi} \to \boldsymbol{\psi})_{\boldsymbol{x}|\boldsymbol{t}} = (\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}} \to \boldsymbol{\psi}_{\boldsymbol{x}|\boldsymbol{t}})$

(iv) $(\forall \boldsymbol{y}\boldsymbol{\varphi})_{\boldsymbol{x}|\boldsymbol{t}} = \begin{cases} \forall \boldsymbol{y}\boldsymbol{\varphi} & \text{if } \boldsymbol{x} = \boldsymbol{y} \\ \forall \boldsymbol{y}(\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}}) & \text{if } \boldsymbol{x} \neq \boldsymbol{y} \end{cases}$
(the first rule in the piecewise function is the case when $\boldsymbol{x}$ does not occur free in $\forall \boldsymbol{y}\boldsymbol{\varphi}$).

This definition is valid by recursion. The type of formula we wish to add to Λ is one of the form $(\forall \boldsymbol{x}\boldsymbol{\varphi} \to \boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}})$. Each term is either a variable symbol, constant symbol, or function symbol applied to variable and constant symbols. In a structure, the variables and constants will be mapped to elements in the universe (or stand-ins for elements in the universe) determined by the structure, and the function symbols will be mapped to functions on the universe. So, every term will be mapped to some element in the universe. Essentially then, this axiom says that if a statement holds for every element in the universe, then it holds for any particular element in the universe. But, as it stands, we need to make some adjustment.

Consider the wff $(\neg\forall \boldsymbol{y}(\boldsymbol{x} \approx \boldsymbol{y}))$. Then the wff

$$(\forall \boldsymbol{x}(\neg\forall \boldsymbol{y}(\boldsymbol{x} \approx \boldsymbol{y})) \to (\neg\forall \boldsymbol{y}(\boldsymbol{x} \approx \boldsymbol{y}))_{\boldsymbol{x}|\boldsymbol{y}})$$

(which is of the form $(\forall \boldsymbol{x}\boldsymbol{\varphi} \to \boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}})$ where $\boldsymbol{\varphi} = (\neg\forall \boldsymbol{y}(\boldsymbol{x} \approx \boldsymbol{y}))$) is the formula

$$(\forall \boldsymbol{x}(\neg\forall \boldsymbol{y}(\boldsymbol{x} \approx \boldsymbol{y})) \to (\neg\forall \boldsymbol{y}(\boldsymbol{y} \approx \boldsymbol{y}))).$$

The antecedent of this wff will be true in any structure with more than one element, but the consequent is always false, so that this formula will be false in almost every structure. It would be very ill-advised to add a formula that could be false to our set of logical axioms for axioms by their nature are statements that are assumed to be true. We must therefore restrict the terms that can actually replace free variables in a formula so that the above disaster does not happen.

Definition 5.7 *A term is* **substitutable** *for a variable $\boldsymbol{x}$ occurring in a formula subject to the following conditions.*

(i) For an atomic formula $\boldsymbol{\varphi}$, any term $\boldsymbol{t}$ is substitutable for any variable $\boldsymbol{x}$ that occurs in $\boldsymbol{\varphi}$.

(ii) For the formula $(\neg\boldsymbol{\varphi})$, term $\boldsymbol{t}$ is substitutable for the variable $\boldsymbol{x}$ if and only if it is substitutable for $\boldsymbol{x}$ in $\boldsymbol{\varphi}$.

(iii) For the formula $(\boldsymbol{\varphi} \rightarrow \boldsymbol{\psi})$, term $\boldsymbol{t}$ is substitutable for variable $\boldsymbol{x}$ if and only if it is substitutable for $\boldsymbol{x}$ in both $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$.

(iv) For the formula $\forall \boldsymbol{y}\boldsymbol{\varphi}$, term $\boldsymbol{t}$ is substitutable for variable $\boldsymbol{x}$ if and only if $\boldsymbol{y}$ does not occur in $\boldsymbol{t}$ and $\boldsymbol{t}$ is substitutable for $\boldsymbol{x}$ in in $\boldsymbol{\varphi}$.

The definition is valid by the recursion. If term $\boldsymbol{t}$ is substitutable for variable $\boldsymbol{x}$ in $\boldsymbol{\varphi}$, then there will be no "disasters" in $\boldsymbol{\varphi}_{x|t}$ like in the example above with the wff $(\forall \boldsymbol{x}\boldsymbol{\varphi} \rightarrow \boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}})$. The variable symbol $\boldsymbol{y}$ was not substitutable for the variable $\boldsymbol{x}$ in the example preceding the definition because the quanifier "captured" the variable $\boldsymbol{y}$ once $\boldsymbol{x}$ was replaced with $\boldsymbol{y}$. So, we add to Λ formulas of the form $(\forall \boldsymbol{x}\boldsymbol{\varphi} \rightarrow \boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}})$ where $\boldsymbol{t}$ is substitutable for $\boldsymbol{x}$ in $\boldsymbol{\varphi}$.

Example 5.8 $(\forall \boldsymbol{v_1}\forall \boldsymbol{v_2}(\boldsymbol{P}\boldsymbol{v_1}\boldsymbol{v_2}) \rightarrow \forall \boldsymbol{v_2}(\boldsymbol{P}\boldsymbol{c}\boldsymbol{v_2}))$ *is in* Λ *for constant symbol c since $\boldsymbol{v_2}$ is distinct from $\boldsymbol{c}$ and $\boldsymbol{c}$ is substitutable for $\boldsymbol{v_1}$ in $\boldsymbol{P}\boldsymbol{v_1}\boldsymbol{v_2}$. However,* $(\forall \boldsymbol{v_1}\forall \boldsymbol{v_2}(\boldsymbol{P}\boldsymbol{v_1}\boldsymbol{v_2}) \rightarrow \forall \boldsymbol{v_2}(\boldsymbol{P}(\boldsymbol{f}\boldsymbol{v_1}\boldsymbol{v_2})\boldsymbol{v_2}))$ *is not in* Λ *for function symbol $\boldsymbol{f}$ since $\boldsymbol{v_2}$ occurs in $\boldsymbol{f}(\boldsymbol{v_1}\boldsymbol{v_2})$.*

5.1.3 Other Formulas

We also wish to include a couple other types of formulas. First, the formula

$$(\forall \boldsymbol{x}(\varphi \to \psi) \to (\forall \boldsymbol{x}\varphi \to \forall \boldsymbol{x}\psi))$$

ought to be included in the logical axioms Λ. This group of axioms will be called group 3. Intuitively, if an implication holds true for the entire universe, then if the antecedent is true for the entire universe then the consequent ought to be true for the entire universe.

Next we include formulas in Λ of the form $(\varphi \to \forall \boldsymbol{x}\varphi)$ where $\boldsymbol{x}$ does not occur free in $\boldsymbol{\varphi}$. This group is axiom group 4. Intuitively, if a variable doesn't exist in a formula or is already bound by a quantifier, then quantifying that variable will add no expressiveness (no stronger claim when interpreted into by a structure).

We will include the following two formulas that involve equality even if our language under consideration does not include equality (we will choose to ignore them when making deductions in our language if the language does not include equality). Of course we should include as a logical axiom the formula $\boldsymbol{x} \approx \boldsymbol{x}$ (axiom group 5). Also, the formula $((\boldsymbol{x} \approx \boldsymbol{y}) \to (\varphi \to \varphi'))$ where φ' is $\boldsymbol{\varphi}$ with $\boldsymbol{x}$ replaced with $\boldsymbol{y}$ in no, some, or all occurrences of $\boldsymbol{x}$ in $\boldsymbol{\varphi}$, ought to be included in Λ (axiom group 6). Intuitively, if two variables represent the same value and if a statement holds true for the first variable, then the statement will also hold true for the second variable. As is clear, the inclusion of these types of formulas requires no special explanation.

5.1.4 Generalizations

Although all of the types of formulas discussed in the last three subsections will be included in Λ, we will actually wish to include more formulas that are related to these basic types.

Definition 5.9 *Formula* $\boldsymbol{\psi}$ *is said to be a* **generalization** *of formula* $\boldsymbol{\varphi}$ *if for some* $n \geq 0$, $\boldsymbol{\psi} = \forall \boldsymbol{x_1} \forall \boldsymbol{x_2} \cdots \forall \boldsymbol{x_n} \boldsymbol{\varphi}$ *(any formula will be a generalization of itself* $(n = 0)$*).*

For Λ, we will include all generalizations of the formulas discussed in the last three subsections. The idea behind a generalization is that it will express the structure of the formula for more elements in the universe. So, having agreed on the base type of formulas in the last three subsections that should be logical axioms, we wish to express that same logical structure for more elements in the universe. (Note: there is a potential problem with substitutability, but such a problem is addressed by alphabetic variants (discussed later).)

Example 5.10 *In Example 5.5 we had the tautology*

$$((\boldsymbol{Px} \to \boldsymbol{Qy}) \to ((\neg \boldsymbol{Qy}) \to (\neg \boldsymbol{Px})))$$

. Two generalizations of this formula are

$$\forall \boldsymbol{x}((\boldsymbol{Px} \to \boldsymbol{Qy}) \to ((\neg \boldsymbol{Qy}) \to (\neg \boldsymbol{Px}))) \text{ and}$$

$$\forall \boldsymbol{y} \forall \boldsymbol{x}((\boldsymbol{Px} \to \boldsymbol{Qy}) \to ((\neg \boldsymbol{Qy}) \to (\neg \boldsymbol{Px})))$$

and hence all three of these formulas are in Λ*.*

5.2 Theorems About Deductions

Having specified exactly what Λ contains, we have completely formalized by what we mean by a deduction (proof) in a first-order setting (refer back to Definition 5.2). By Theorem 5.3, there is a deduction of a formula if and only if that formula is a theorem (a formal theorem, not the numbered meta-theorems such as "Theorem 5.3") of our set of formulas which represents a set of assumptions, or givens, Γ. So, to know that a formula is a theorem of a set of formulas it suffices to know that there is a deduction of the formulas from the set of assumptions. It is not even necessary to demonstrate what the deduction actually is; it just suffices to know that a deduction exists. As such it will be useful to prove several metatheorems about deducibility. These rules will be useful to us as tools in the ensuing development.

Theorem 5.11 (Generalization Theorem) *If* $\Gamma \vdash \boldsymbol{\varphi}$ *and* $\boldsymbol{x}$ *does not occur free in any formula in* Γ*, then* $\Gamma \vdash \forall \boldsymbol{x}\boldsymbol{\varphi}$*.*

Proof: We show that $\forall \boldsymbol{x}\boldsymbol{\varphi}$ is a theorem of Γ by an induction argument since the set of theorems of Γ are generated by the rule of inference $\mathcal{I}$ from the set $\Gamma \cup \Lambda$. We start with the set of theorems of Γ,

$$S = \{\boldsymbol{\varphi} : \ \Gamma \vdash \forall \boldsymbol{x}\boldsymbol{\varphi}\}.$$

If $\boldsymbol{\varphi} \in \Lambda$, then $\forall \boldsymbol{x}\boldsymbol{\varphi} \in \Lambda$ since this latter formula is a generalization of the former. If $\boldsymbol{\varphi} \in \Gamma$, then x does not occur free in $\boldsymbol{\varphi}$ by assumption. So, the formula $(\boldsymbol{\varphi} \rightarrow \forall \boldsymbol{x}\boldsymbol{\varphi})$ is a logical axiom, and since $\mathcal{I}(\boldsymbol{\varphi}, (\boldsymbol{\varphi} \rightarrow \forall \boldsymbol{x}\boldsymbol{\varphi})) = \forall \boldsymbol{x}\boldsymbol{\varphi}$, $\Gamma \vdash \forall \boldsymbol{x}\boldsymbol{\varphi}$. Thus, $\Gamma \cup \Lambda \subseteq S$. All that remains is to show that S is closed under the rule of inference. Let $\boldsymbol{\varphi}, \boldsymbol{\chi} \in S$. Either $\boldsymbol{\chi} = (\boldsymbol{\varphi} \rightarrow \boldsymbol{\psi})$ for some formula $\boldsymbol{\psi}$ or this is not the case. If it is not the case, then $\mathcal{I}(\boldsymbol{\varphi}, \boldsymbol{\chi}) = \boldsymbol{\chi} \in S$.

If $\boldsymbol{\chi} = (\boldsymbol{\varphi} \to \boldsymbol{\psi})$ for some formula $\boldsymbol{\psi}$, then, $\mathcal{I}(\boldsymbol{\varphi}, \boldsymbol{\chi}) = \boldsymbol{\psi}$. So, $\boldsymbol{\psi}$ is a theorem of Γ since $\boldsymbol{\varphi}$ and $\boldsymbol{\chi}$ are. Now,

$$\boldsymbol{\tau} = (\forall \boldsymbol{x}(\boldsymbol{\varphi} \to \boldsymbol{\psi}) \to (\forall \boldsymbol{x}\boldsymbol{\varphi} \to \forall \boldsymbol{x}\boldsymbol{\psi}))$$

is a logical axiom as discussed above. Since $\boldsymbol{\chi} = (\boldsymbol{\varphi} \to \boldsymbol{\psi})$, and $\boldsymbol{\chi} \in S$, then $\Gamma \vdash \forall \boldsymbol{x}(\boldsymbol{\varphi} \to \boldsymbol{\psi})$. Now, $\mathcal{I}(\forall \boldsymbol{x}(\boldsymbol{\varphi} \to \boldsymbol{\psi}), \boldsymbol{\tau}) = (\forall \boldsymbol{x}\boldsymbol{\varphi} \to \forall \boldsymbol{x}\boldsymbol{\psi})$. Since $\boldsymbol{\varphi} \in S$ and hence $\Gamma \vdash \forall \boldsymbol{x}\boldsymbol{\varphi}$, we have that

$$\mathcal{I}(\forall \boldsymbol{x}\boldsymbol{\varphi}, (\forall \boldsymbol{x}\boldsymbol{\varphi} \to \forall \boldsymbol{x}\boldsymbol{\psi})) = \forall \boldsymbol{x}\boldsymbol{\psi}.$$

Therefore, $\Gamma \vdash \forall \boldsymbol{x}\boldsymbol{\psi}$, and $\boldsymbol{\psi} \in S$. Thus, $\mathcal{I}(\boldsymbol{\varphi}, \boldsymbol{\chi}) \in S$ in any case, and S is closed under the rule of inference. Thus, S is in fact the entire set of theorems of Γ, and our meta-theorem holds. ∎

Notice that the variable $\boldsymbol{x}$ can occur free in $\boldsymbol{\varphi}$ as long as $\boldsymbol{\varphi}$ is not an element of Γ. The idea behind the theorem is that if we can show that a statement $\boldsymbol{\varphi}$ holds for an arbitrary element, $\boldsymbol{x}$, of the universe, then the statement should hold for every element in the universe.

Theorem 5.12 *[Rule T] If* $\Gamma \vdash \boldsymbol{\gamma_1}, \Gamma \vdash \boldsymbol{\gamma_2}, \ldots, \Gamma \vdash \boldsymbol{\gamma_n}$ *and* $\{\boldsymbol{\gamma_1}, \ldots, \boldsymbol{\gamma_n}\}$ *tautologically implies* $\boldsymbol{\tau}$*, then* $\Gamma \vdash \boldsymbol{\tau}$*.*

Proof: Since $\{\boldsymbol{\gamma_1}, \ldots, \boldsymbol{\gamma_n}\}$ tautologically implies $\boldsymbol{\tau}$,

$$(\boldsymbol{\gamma_1} \to \boldsymbol{\gamma_2} \to \cdots \to \boldsymbol{\gamma_n} \to \boldsymbol{\tau})$$

is a tautology in the sentential sense by Corollary 2.46.1. Hence, this formula is a logical axiom, and since $\Gamma \vdash \boldsymbol{\gamma_1}, \ldots, \Gamma \vdash \boldsymbol{\gamma_n}$, we may apply the rule of inference $\mathcal{I}$ n-times to obtain $\Gamma \vdash \boldsymbol{\tau}$. ∎

Theorem 5.13 (The Deduction Theorem) $\Gamma \cup \{\boldsymbol{\gamma}\} \vdash \boldsymbol{\varphi}$ *if and only if* $\Gamma \vdash (\boldsymbol{\gamma} \to \boldsymbol{\varphi})$

Proof: Suppose $\Gamma \cup \{\gamma\} \vdash \varphi$. Then by Theorem 5.6, $(\Gamma \cup \{\gamma\}) \cup \Lambda$ tautologically implies φ. By Theorem 2.44, $\Gamma \cup \Lambda$ tautologically implies $(\gamma \to \varphi)$. By Theorem 5.6, $\Gamma \vdash (\gamma \to \varphi)$

Conversely, suppose $\Gamma \vdash (\gamma \to \varphi)$. Since $\mathcal{I}(\gamma, (\gamma \to \varphi)) = \varphi$, it is clear that $\Gamma \cup \{\gamma\} \vdash \varphi$. ∎

This theorem reflects the fact that if we assume that a statement γ is true and are able to deduce φ from our assumption, this will be equivalent to showing that the implication "If γ, then φ" is true.

Theorem 5.14 (Contraposition) $\Gamma \vdash (\varphi \to \psi)$ *if and only if* $\Gamma \vdash ((\neg\psi) \to (\neg\varphi))$.

Proof: The formula $((\varphi \to \psi) \to ((\neg\psi) \to (\neg\varphi)))$ is a tautology. Hence it must be in Λ. Assuming $\Gamma \vdash (\varphi \to \psi)$ and since

$$\mathcal{I}((\varphi \to \psi), ((\varphi \to \psi) \to ((\neg\psi) \to (\neg\varphi)))) = ((\neg\psi) \to (\neg\varphi)),$$

we can see that $\Gamma \vdash ((\neg\psi) \to (\neg\varphi))$. So if $\Gamma \vdash (\varphi \to \psi)$, then $\Gamma \vdash ((\neg\psi) \to (\neg\varphi))$. By symmetry, the converse will be true. ∎

These last two theorems indicate that rules of deduction that we know to be true in the real world (such as contraposition) hold true in our formal model for logic. What about proof by contradiction?

Definition 5.15 *A set of formulas* Γ *is* **inconsistent** *if for some formula* φ *both* $\Gamma \vdash \varphi$ *and* $\Gamma \vdash (\neg\varphi)$.

Theorem 5.16 (Proof by Contradiction) *If the set of formulas* $\Gamma \cup \{\varphi\}$ *is inconsistent, then* $\Gamma \vdash (\neg\varphi)$.

Proof: Since $\Gamma \cup \{\varphi\}$ is inconsistent, there is a formula β such that $\Gamma \cup \{\varphi\} \vdash \beta$ and $\Gamma \cup \{\varphi\} \vdash (\neg\beta)$. By the Deduction Theorem, we have

$\Gamma \vdash (\varphi \to \beta)$ and $\Gamma \vdash (\varphi \to (\neg\beta))$. Notice that the two formulas $(\varphi \to \beta)$ and $(\varphi \to (\neg\beta))$ can only both be true for the same truth assignment if and only if φ is false. Hence, the set $\{(\varphi \to \beta), (\varphi \to (\neg\beta))\}$ tautologically implies $(\neg\varphi)$. By Rule T, $\Gamma \vdash (\neg\varphi)$ ∎

This theorem reflects our proof by contradiction procedure. We assume the opposite of what we are trying to show, we demonstrate an inconsistency with our set of assumptions, Γ, and we may then conclude that what we wanted to show is in fact true. Another theorem dealing with inconsistent sets of formulas is of note.

Theorem 5.17 *If* Γ *is inconsistent, then for any formula* α*,* $\Gamma \vdash \alpha$*.*

Proof: The following table demonstrates that for any formulas α and φ, $(\varphi \to ((\neg\varphi) \to \alpha))$ is a tautology.

φ	α	$(\neg\varphi)$	$((\neg\varphi) \to \alpha)$	$(\varphi \to ((\neg\varphi) \to \alpha))$
T	T	F	T	T
T	F	F	T	T
F	T	T	T	T
F	F	T	F	T

Since Γ is inconsistent, by definition, there is a formula φ such that both $\Gamma \vdash \varphi$, and $\Gamma \vdash (\neg\varphi)$. Now, $\mathcal{I}(\varphi, (\varphi \to ((\neg\varphi) \to \alpha))) = ((\neg\varphi) \to \alpha))$, and $\mathcal{I}((\neg\varphi), ((\neg\varphi) \to \alpha))) = \alpha$. Hence, $\Gamma \vdash \alpha$. ∎

This theorem reflects the fact that any statement follows from a contradiction and so reasoning from a contradiction is unhelpful.

In our discussion about substitutability of terms for variables, it is clear that we may extend the substitutability idea of a variable for a constant symbol c we are thus justified in stating the following theorem.

Theorem 5.18 (Generalization on Constants) *Assume that $\Gamma \vdash \boldsymbol{\varphi}$ and that $\boldsymbol{c}$ is a constant symbol which does not occur in Γ. Then there is a variable $\boldsymbol{y}$ (which does not occur in $\boldsymbol{\varphi}$) such that $\Gamma \vdash \forall \boldsymbol{y} \boldsymbol{\varphi}_{\boldsymbol{c}|\boldsymbol{y}}$. Furthermore, there is a deduction of $\forall \boldsymbol{y} \boldsymbol{\varphi}_{\boldsymbol{c}|\boldsymbol{y}}$ from Γ in which $\boldsymbol{c}$ does not occur.*

The intuition behind this theorem is that if a statement holds for a particular constant in the universe, but our list of assumptions stated nothing about the constant, then essentially our constant behaves as an arbitrary element in the universe. We may thus generalize the statement to hold for the whole universe. Before proving the theorem, we will need a couple of lemmas.

Lemma 5.2.1 *Let $\boldsymbol{t}$ be a term substitutable for $\boldsymbol{x}$ in $\boldsymbol{\beta}$. If $\boldsymbol{x} \neq \boldsymbol{y}$, $\boldsymbol{y}$ does not occur in $\boldsymbol{\beta}$, and $\boldsymbol{c}$ does not occur in $\boldsymbol{t}$, then $\boldsymbol{t}$ is substitutable for $\boldsymbol{x}$ in $\boldsymbol{\beta}_{\boldsymbol{c}|\boldsymbol{y}}$ and*

$$(\boldsymbol{\beta}_{\boldsymbol{x}|\boldsymbol{t}})_{\boldsymbol{c}|\boldsymbol{y}} = (\boldsymbol{\beta}_{\boldsymbol{c}|\boldsymbol{y}})_{\boldsymbol{x}|\boldsymbol{t}}$$

Proof: Since $\boldsymbol{x}$ is a variable symbol and $\boldsymbol{c}$ is a constant symbol, no incidences of $\boldsymbol{x}$ in $\boldsymbol{\beta}$, including quantification of $\boldsymbol{x}$ will be replaced by $\boldsymbol{y}$ in $\boldsymbol{\beta}_{\boldsymbol{c}|\boldsymbol{y}}$. Furthermore, since $\boldsymbol{y} \neq \boldsymbol{x}$, there will be no more incidences of $\boldsymbol{x}$ in $\boldsymbol{\beta}_{\boldsymbol{c}|\boldsymbol{y}}$ than there are in $\boldsymbol{\beta}$. Clearly then, since $\boldsymbol{t}$ is substitutable for $\boldsymbol{x}$ in $\boldsymbol{\beta}$, $\boldsymbol{t}$ will be substitutable for $\boldsymbol{x}$ in $\boldsymbol{\beta}_{\boldsymbol{c}|\boldsymbol{y}}$.

Since $\boldsymbol{c}$ does not occur in $\boldsymbol{t}$, there will be no more incidences of $\boldsymbol{c}$ in $\boldsymbol{\beta}_{\boldsymbol{x}|\boldsymbol{t}}$ than in $\boldsymbol{\beta}$. In fact both of these formulas will have exactly the same incidences of $\boldsymbol{c}$ since $\boldsymbol{c} \neq \boldsymbol{x}$. These statements together with those made in the first paragraph indicate that

$$(\boldsymbol{\beta}_{\boldsymbol{x}|\boldsymbol{t}})_{\boldsymbol{c}|\boldsymbol{y}} = (\boldsymbol{\beta}_{\boldsymbol{c}|\boldsymbol{y}})_{\boldsymbol{x}|\boldsymbol{t}}.$$

∎

Let $\boldsymbol{t}_{\boldsymbol{c}|\boldsymbol{y}}$ designate the term $\boldsymbol{t}$ where every incidence of $\boldsymbol{c}$ is replaced with $\boldsymbol{y}$.

Lemma 5.2.2 *Let $\boldsymbol{t}$ be a term substitutable for $\boldsymbol{x}$ in $\boldsymbol{\beta}$. If $\boldsymbol{x} \neq \boldsymbol{y}$, $\boldsymbol{y}$ does not occur in $\boldsymbol{\beta}$, and $\boldsymbol{c}$ does occur in $\boldsymbol{t}$, then $\boldsymbol{t}_{\boldsymbol{c}|\boldsymbol{y}}$ is substitutable for $\boldsymbol{x}$ in $\boldsymbol{\beta}_{\boldsymbol{c}|\boldsymbol{y}}$ and*

$$(\boldsymbol{\beta}_{\boldsymbol{x}|\boldsymbol{t}})_{\boldsymbol{c}|\boldsymbol{y}} = (\boldsymbol{\beta}_{\boldsymbol{c}|\boldsymbol{y}})_{\boldsymbol{x}|\boldsymbol{t}_{\boldsymbol{c}|\boldsymbol{y}}}$$

Proof: We may use all of the same reasoning as in the last lemma. Substitutability will follow since $\boldsymbol{x} \neq \boldsymbol{y}$. It is seen that the equality holds after a moments reflection

The reader may wish to examine formulas $((\boldsymbol{Pxc})_{\boldsymbol{x}|\boldsymbol{c}})_{\boldsymbol{c}|\boldsymbol{y}}$ and $((\boldsymbol{Pxc})_{\boldsymbol{c}|\boldsymbol{y}})_{\boldsymbol{x}|\boldsymbol{c}}$ to see the nuances in the lemmas discussed above. We are now going to prove the theorem for Generalization on Constants.

Proof: (Generalization on Constants) Let $\langle \boldsymbol{\alpha_0}, \boldsymbol{\alpha_1}, \ldots, \boldsymbol{\alpha_n} \rangle$ be a deduction of $\boldsymbol{\varphi}$ from Γ (hence $\boldsymbol{\alpha_n} = \boldsymbol{\varphi}$). Choose i to be the smallest natural number such that $\boldsymbol{v_i}$ does not occur in any of the $\boldsymbol{\alpha_j}$'s in the deduction given for $\boldsymbol{\varphi}$, and let $\boldsymbol{y} = \boldsymbol{v_i}$. We claim that

$$\langle (\boldsymbol{\alpha_0})_{\boldsymbol{c}|\boldsymbol{y}}, (\boldsymbol{\alpha_1})_{\boldsymbol{c}|\boldsymbol{y}}, \ldots, (\boldsymbol{\alpha_n})_{\boldsymbol{c}|\boldsymbol{y}} \rangle$$

is a deduction of $\boldsymbol{\varphi}_{\boldsymbol{c}|\boldsymbol{y}}$ from Γ (note that $(\boldsymbol{\alpha_n})_{\boldsymbol{c}|\boldsymbol{y}} = \boldsymbol{\varphi}_{\boldsymbol{c}|\boldsymbol{y}}$). We examine each component in the potential deduction by cases.

Case 1: $\boldsymbol{\alpha_j} \in \Gamma$. By assumption, since $\boldsymbol{c}$ does not occur in any formula in Γ, then $\boldsymbol{c}$ does not occur in $\boldsymbol{\alpha_j}$, so then $(\boldsymbol{\alpha_j})_{\boldsymbol{c}|\boldsymbol{y}} = \boldsymbol{\alpha_j} \in \Gamma$.

Case 2: $\boldsymbol{\alpha_j} \in \Lambda$. If $\boldsymbol{\alpha_j}$ is a tautology, then $(\boldsymbol{\alpha_j})_{\boldsymbol{c}|\boldsymbol{y}}$ will be a tautology as well as is clear from the discussion in subsection 5.1.1 since tautologies do not depend on the specific variables and constants involved in first-order formulas but merely the logical connectives involved in the tautology.

If $\boldsymbol{\alpha_j} = (\forall \boldsymbol{x\beta} \rightarrow \boldsymbol{\beta}_{\boldsymbol{x}|\boldsymbol{t}})$ where $\boldsymbol{t}$ is substitutable for $\boldsymbol{x}$ in β. Then,

$$(\forall \boldsymbol{x\beta} \rightarrow \boldsymbol{\beta}_{\boldsymbol{x}|\boldsymbol{t}})_{c|y} = ((\forall \boldsymbol{x\beta})_{\boldsymbol{c}|\boldsymbol{y}} \rightarrow (\boldsymbol{\beta}_{\boldsymbol{x}|\boldsymbol{t}})_{\boldsymbol{c}|\boldsymbol{y}}).$$

Since $\boldsymbol{x} \neq \boldsymbol{y}$ ($\boldsymbol{y}$ does not occur in $\boldsymbol{\alpha_j}$), $\boldsymbol{y}$ is substitutable for $\boldsymbol{c}$ in $\forall \boldsymbol{x\beta}$, and

$$\boldsymbol{(}(\forall \boldsymbol{x\beta})_{\boldsymbol{c|y}} \rightarrow (\boldsymbol{\beta_{x|t}})_{\boldsymbol{c|y}}\boldsymbol{)} = \boldsymbol{(}\ \forall \boldsymbol{x\beta_{c|y}} \rightarrow (\boldsymbol{\beta_{x|t}})_{\boldsymbol{c|y}}\boldsymbol{)}.$$

Applying Lemmas 5.2.1 and 5.2.2, $\boldsymbol{t}$ and $\boldsymbol{t_{c|y}}$ are both substitutable for $\boldsymbol{x}$ in $\boldsymbol{\beta_{c|y}}$, and either $(\boldsymbol{\beta_{x|t}})_{\boldsymbol{c|y}} = (\boldsymbol{\beta_{c|y}})_{\boldsymbol{x|t}}$ or $(\boldsymbol{\beta_{x|t}})_{\boldsymbol{c|y}} = (\boldsymbol{\beta_{c|y}})_{\boldsymbol{x|t_{c|y}}}$ depending on whether $\boldsymbol{c}$ occurs in $\boldsymbol{t}$. So, we have that either

$$\boldsymbol{(}\ \forall \boldsymbol{x\beta_{c|y}} \rightarrow (\boldsymbol{\beta_{x|t}})_{\boldsymbol{c|y}}\boldsymbol{)} = \boldsymbol{(}\ \forall \boldsymbol{x\beta_{c|y}} \rightarrow (\boldsymbol{\beta_{c|y}})_{\boldsymbol{x|t}}\boldsymbol{)}$$

or

$$\boldsymbol{(}\ \forall \boldsymbol{x\beta_{c|y}} \rightarrow (\boldsymbol{\beta_{x|t}})_{\boldsymbol{c|y}}\boldsymbol{)} = \boldsymbol{(}\ \forall \boldsymbol{x\beta_{c|y}} \rightarrow (\boldsymbol{\beta_{c|y}})_{\boldsymbol{x|t_{c|y}}}\boldsymbol{)}.$$

In either case, these formulas are logical axioms.

Most of the rest of the cases for $\boldsymbol{\alpha_j} \in \Lambda$ follow very easily upon inspection. Only one other case is of note. Suppose $\boldsymbol{\alpha_j} = \boldsymbol{(x \approx z \rightarrow (\beta \rightarrow \beta'))}$ where $\boldsymbol{\beta'}$ is $\boldsymbol{\beta}$ with some of incidences of $\boldsymbol{x}$ replaced by $\boldsymbol{z}$. So,

$$(\boldsymbol{\alpha_j})_{\boldsymbol{c|y}} = \boldsymbol{(x \approx z \rightarrow (\beta_{c|y} \rightarrow \beta'_{c|y}))}.$$

Note that $\boldsymbol{\beta'_{c|y}} = (\boldsymbol{\beta_{c|y}})\boldsymbol{'}$ since $\boldsymbol{c} \neq \boldsymbol{x}$ and $\boldsymbol{c} \neq \boldsymbol{z}$ so that $(\boldsymbol{\alpha_j})_{\boldsymbol{c|y}}$ will be a logical axiom of the same type as $\boldsymbol{\alpha_j}$.

We have covered the cases where $\boldsymbol{\alpha_j}$ is in Γ or in Λ. Suppose now that $\boldsymbol{\alpha_j} = \mathcal{I}(\boldsymbol{\alpha_i}, \boldsymbol{(\alpha_i \rightarrow \alpha_j)})$ where $i < j$. Now,

$$\boldsymbol{(\alpha_i \rightarrow \alpha_j)}_{\boldsymbol{c|y}} = \boldsymbol{(}(\boldsymbol{\alpha_i})_{\boldsymbol{c|y}} \rightarrow (\boldsymbol{\alpha_j})_{\boldsymbol{c|y}}\boldsymbol{)}, \text{ and}$$

$$\mathcal{I}((\boldsymbol{\alpha_i})_{\boldsymbol{c|y}}, \boldsymbol{(}(\boldsymbol{\alpha_i})_{\boldsymbol{c|y}} \rightarrow (\boldsymbol{\alpha_j})_{\boldsymbol{c|y}}\boldsymbol{)}) = (\boldsymbol{\alpha_j})_{\boldsymbol{c|y}},$$

so that $(\boldsymbol{\alpha_j})_{\boldsymbol{c|y}}$ is generated from our rule of inference. As this is the last case for $\boldsymbol{\alpha_j}$, we conclude that

$$\langle (\boldsymbol{\alpha_0})_{\boldsymbol{c|y}}, (\boldsymbol{\alpha_1})_{\boldsymbol{c|y}}, \ldots, (\boldsymbol{\alpha_n})_{\boldsymbol{c|y}} \rangle$$

is a deduction of $\varphi_{c|y}$ from Γ.

Now let Φ be the finite set of formulas from Γ used in our deduction of $\varphi_{c|y}$. As mentioned above, any such formula $(\alpha_j)_{c|y}$ will be equal to α_j, and the variable y will not occur in this formula by assumption. Since $\Phi \vdash \varphi_{c|y}$, by the Generalization Theorem, $\Phi \vdash \forall y \varphi_{c|y}$. Since $\Phi \subseteq \Gamma$, we have $\Gamma \vdash \forall y \varphi_{c|y}$. Of course in our deduction we used to show $\Phi \vdash \varphi_{c|y}$, the symbol c did not occur (since it had been replaced by y for each α_j). Since the proof of the Generalization Theorem added no new symbols, we may say that there is a deduction in which c does not occur for $\Gamma \vdash \forall y \varphi_{c|y}$. ∎

This theorem and the following corollaries will be useful in our ensuing discussion.

Corollary 5.18.1 *Under the same assumptions, there is a deduction of* $\varphi_{c|y}$ *from* Γ *in which* c *does not appear.*

Proof: This fact is seen in the proof of the theorem. ∎

Corollary 5.18.2 *Assume that* $\Gamma \vdash \varphi_{x|c}$*, where the constant symbol* c *does not occur in* Γ *or in* φ*. Then* $\Gamma \vdash \forall x \varphi$*, and there is a deduction of* $\forall x \varphi$ *from* Γ *in which* c *does not occur.*

Proof: By Generalization on Constants, there is a variable y that does not occur in $\varphi_{x|c}$ such that $\Gamma \vdash \forall y (\varphi_{x|c})_{c|y}$ and there is a deduction of $\forall y (\varphi_{x|c})_{c|y}$ in which c does not occur. Note that since c does not occur in φ, the only incidences of c in $\varphi_{x|c}$ will come from where c was substituted for x. It is clear then that$(\varphi_{x|c})_{c|y} = \varphi_{x|y}$, and hence $\forall y (\varphi_{x|c})_{c|y} = \forall y \varphi_{x|y}$. We wish to show that x is substitutable for y in $\varphi_{x|y}$ and that $(\varphi_{x|y})_{y|x} = \varphi$ so that we may say that $(\forall y \varphi_{x|y} \to \varphi)$ is a logical axiom. Since y does not appear in $\varphi_{x|c}$, y occurs in φ if and only if x occurs in φ and $x = y$. If

$\boldsymbol{x} = \boldsymbol{y}$, then $\Gamma \vdash \forall \boldsymbol{y}\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{y}}$ is the same thing as $\Gamma \vdash \forall \boldsymbol{x}\boldsymbol{\varphi}$ since $\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{x}} = \boldsymbol{\varphi}$, and we are done. So, we suppose that $\boldsymbol{x} \neq \boldsymbol{y}$. Hence, since there are no incidences of $\boldsymbol{y}$ in $\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{c}}$, there will be no incidences of $\boldsymbol{y}$ in $\boldsymbol{\varphi}$. Hence, $\boldsymbol{x}$ will occur in $\boldsymbol{\varphi}$ precisely when $\boldsymbol{y}$ occurs in $\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{y}}$. So, we must have that $\boldsymbol{x}$ is substitutable for $\boldsymbol{y}$ in $\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{y}}$ and that $(\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{y}})_{\boldsymbol{y}|\boldsymbol{x}} = \boldsymbol{\varphi}$. Since $(\forall \boldsymbol{y}\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{y}} \to (\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{y}})_{\boldsymbol{y}|\boldsymbol{x}})$ is a logical axiom (since $\boldsymbol{x}$ is substitutable for $\boldsymbol{y}$ in $\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{y}}$), we have that $(\forall \boldsymbol{y}\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{y}} \to \boldsymbol{\varphi})$ is a logical axiom. Now,

$$\mathcal{I}(\forall \boldsymbol{y}\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{y}}, (\forall \boldsymbol{y}\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{y}} \to \boldsymbol{\varphi})) = \boldsymbol{\varphi}.$$

Now, $\forall \boldsymbol{y}\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{y}} \vdash \boldsymbol{\varphi}$ and $\boldsymbol{x}$ does not occur free in in $\forall \boldsymbol{y}\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{y}}$, so that by Generalization Theorem $\forall \boldsymbol{y}\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{y}} \vdash \forall \boldsymbol{x}\boldsymbol{\varphi}$. Note that $\boldsymbol{c}$ does not occur in this deduction since it does not occur in $\boldsymbol{\varphi}$. Hence, we have a deduction of $\forall \boldsymbol{x}\boldsymbol{\varphi}$ from Γ in which $\boldsymbol{c}$ does not occur. ∎

Corollary 5.18.3 *Assume that the constant symbol $\boldsymbol{c}$ does not occur in $\boldsymbol{\varphi}$, $\boldsymbol{\psi}$, or in Γ, and that $\Gamma \cup \{\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{c}}\} \vdash \boldsymbol{\psi}$, then $\Gamma \cup \{\exists \boldsymbol{x}\boldsymbol{\varphi}\} \vdash \boldsymbol{\psi}$ and there is a deduction of $\boldsymbol{\psi}$ from $\Gamma \cup \{\exists \boldsymbol{x}\boldsymbol{\varphi}\}$ in which $\boldsymbol{c}$ does not occur.*

Proof: Since $\Gamma \cup \{\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{c}}\} \vdash \boldsymbol{\psi}$, by the Deduction Theorem and Contraposition, this statement holds if and only if $\Gamma \cup \{(\neg\boldsymbol{\psi})\} \vdash (\neg\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{c}})$. Since $(\neg\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{c}}) = (\neg\boldsymbol{\varphi})_{\boldsymbol{x}|\boldsymbol{c}}$, by the proceeding corollary, we have that

$$\Gamma \cup \{(\neg\boldsymbol{\psi})\} \vdash \forall \boldsymbol{x}(\neg\boldsymbol{\varphi}).$$

Applying the Deduction Theorem and Contraposition again and using some appropriate abbreviations, the last statement is equivalent to the desired result. ∎

Before leaving this chapter, we mention two more topics.

5.2.1 Theorems of Deduction Involving Equality

Although theorems of deducibility involving the formal language equality symbol will be important in our discussion in future chapters, we state these theorems without proof, appealing to the intuitive nature of the results and trusting that the reader can derive adequate proofs.

Theorem 5.19 *(i)* $\vdash \forall \boldsymbol{x}(\boldsymbol{x} \approx \boldsymbol{x})$ *(formal reflexivity of* $\approx$*).*

(ii) $\vdash \forall \boldsymbol{x} \forall \boldsymbol{y}[(\boldsymbol{x} \approx \boldsymbol{y}) \to (\boldsymbol{y} \approx \boldsymbol{x})]$ *(formal symmetry of* $\approx$*).*

(iii) $\vdash \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{z}[((\boldsymbol{x} \approx \boldsymbol{y}) \to ((\boldsymbol{y} \approx \boldsymbol{z}) \to (\boldsymbol{x} \approx \boldsymbol{z})))]$ *(formal transitivity of* $\approx$*).*

Theorem 5.20 *For* $\boldsymbol{R}$ *a two place predicate symbol,*

$$\vdash \forall \boldsymbol{x_1} \forall \boldsymbol{x_2} \forall \boldsymbol{y_1} \forall \boldsymbol{y_2}[(\boldsymbol{x_1} \approx \boldsymbol{y_1} \to (\boldsymbol{x_2} \approx \boldsymbol{y_2} \to \boldsymbol{R x_1 x_2} \to \boldsymbol{R y_1 y_2}))].$$

Similarly for n-place predicate symbols.

Theorem 5.21 *For* $\boldsymbol{f}$ *a two place function symbol,*

$$\vdash \forall \boldsymbol{x_1} \forall \boldsymbol{x_2} \forall \boldsymbol{y_1} \forall \boldsymbol{y_2}((\boldsymbol{x_1} \approx \boldsymbol{y_1} \to (\boldsymbol{x_2} \approx \boldsymbol{y_2} \to \boldsymbol{f}(\boldsymbol{x_1 x_2}) \approx \boldsymbol{f}(\boldsymbol{y_1 y_2})))).$$

Similarly for n-place function symbols.

5.2.2 Alphabetic Variants

Suppose we desire to show that $\vdash [(\forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{z} \boldsymbol{Q x y z}) \to (\forall \boldsymbol{z} \boldsymbol{Q z z z})]$. Notice that since $\boldsymbol{z}$ is not substitutable for either $\boldsymbol{x}$ or $\boldsymbol{y}$ in $\forall \boldsymbol{z} \boldsymbol{Q x y z}$, we cannot use our substitution results very effectively. However we may demonstrate $\vdash [(\forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{w} \boldsymbol{Q x y w}) \to (\forall \boldsymbol{z} \boldsymbol{Q z z z})]$ very simply. Note that

$$(\forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{w} \boldsymbol{Q x y w} \to \forall \boldsymbol{y} \forall \boldsymbol{w} \boldsymbol{Q z y w}),\ (\forall \boldsymbol{y} \forall \boldsymbol{w} \boldsymbol{Q z y w} \to \forall \boldsymbol{w} \boldsymbol{Q z z w}),\ \text{and}$$

$$(\forall \boldsymbol{w} \boldsymbol{Q} \boldsymbol{z} \boldsymbol{z} \boldsymbol{w} \rightarrow \boldsymbol{Q} \boldsymbol{z} \boldsymbol{z} \boldsymbol{z})$$

are each substitution logical axioms since $\boldsymbol{z}$ is substitutable for each of $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{w}$ (assuming $\boldsymbol{z}$ is distinct from each of these variables). So,

$$\forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{w} \boldsymbol{Q} \boldsymbol{x} \boldsymbol{y} \boldsymbol{w} \vdash \boldsymbol{Q} \boldsymbol{z} \boldsymbol{z} \boldsymbol{z}$$

and by the Generalization Theorem, $\forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{w} \boldsymbol{Q} \boldsymbol{x} \boldsymbol{y} \boldsymbol{w} \vdash \forall \boldsymbol{z} \boldsymbol{Q} \boldsymbol{z} \boldsymbol{z} \boldsymbol{z}$. By the Deduction Theorem, $\vdash (\forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{w} \boldsymbol{Q} \boldsymbol{x} \boldsymbol{y} \boldsymbol{w} \rightarrow \forall \boldsymbol{z} \boldsymbol{Q} \boldsymbol{z} \boldsymbol{z} \boldsymbol{z})$.

Of course, the structure of

$$(\forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{z} \boldsymbol{Q} \boldsymbol{x} \boldsymbol{y} \boldsymbol{z} \rightarrow \forall \boldsymbol{z} \boldsymbol{Q} \boldsymbol{z} \boldsymbol{z} \boldsymbol{z}) \text{ and}$$

$$(\forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{w} \boldsymbol{Q} \boldsymbol{x} \boldsymbol{y} \boldsymbol{w} \rightarrow \forall \boldsymbol{z} \boldsymbol{Q} \boldsymbol{z} \boldsymbol{z} \boldsymbol{z})$$

appears to be logically identical. In fact, we can demonstrate that

$$\forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{z} \boldsymbol{Q} \boldsymbol{x} \boldsymbol{y} \boldsymbol{z} \vdash \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{w} \boldsymbol{Q} \boldsymbol{x} \boldsymbol{y} \boldsymbol{w} \text{ and}$$

$$\forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{w} \boldsymbol{Q} \boldsymbol{x} \boldsymbol{y} \boldsymbol{w} \vdash \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{z} \boldsymbol{Q} \boldsymbol{x} \boldsymbol{y} \boldsymbol{z}.$$

For $\forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{z} \boldsymbol{Q} \boldsymbol{x} \boldsymbol{y} \boldsymbol{z} \vdash \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{w} \boldsymbol{Q} \boldsymbol{x} \boldsymbol{y} \boldsymbol{w}$, we have

For $\forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{w} \boldsymbol{Q} \boldsymbol{x} \boldsymbol{y} \boldsymbol{w} \vdash \forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{z} \boldsymbol{Q} \boldsymbol{x} \boldsymbol{y} \boldsymbol{z}$, the argument is exactly symmetric. Via this method we can show our original desired result

$$\vdash (\forall \boldsymbol{x} \forall \boldsymbol{y} \forall \boldsymbol{z} \boldsymbol{Q} \boldsymbol{x} \boldsymbol{y} \boldsymbol{z} \rightarrow \forall \boldsymbol{z} \boldsymbol{Q} \boldsymbol{z} \boldsymbol{z} \boldsymbol{z}),$$

but it took a great deal of work because of unhelpful quirk due the definition of substitutability and the chosen quantified variable, neither of which are salient features of the result we wish to demonstrate. The whole process above amounted to changing the variable over which we were quantifying. The example above is illustrative of a theorem that will allow us to change the

$\vdash (\forall z Qxyz \to Qxyw)$	w substitutable for z in $Qxyz$; Logical Axiom
$\vdash (\forall z Qxyz \to \forall w Qxyw)$	Deduction Theorem; Generalization Theorem; Deduction Theorem
$\vdash \forall y(\forall z Qxyz \to \forall w Qxyw)$	Generalization Theorem
$\vdash (\forall y \forall z Qxyz \to \forall y \forall w Qxyw)$	Logical Axiom of the form $(\forall y(\alpha \to \beta) \to (\forall y \alpha \to \beta))$; Rule of Inference
$\vdash (\forall x \forall y \forall z Qxyz \to \forall x \forall y \forall w Qxyw)$	Similar to the last few steps.
$\forall x \forall y \forall z Qxyz \vdash \forall x \forall y \forall w Qxyw$	Deduction Theorem

variable over which we quantify when we wish to substitute in a term that is not substitutable under our current quantified variable. We state the theorem without proof, trusting that the reader has an intuitive grasp of what we are doing.

Theorem 5.22 (Existence of Alphabetic Variants) *Let φ be a formula, t a term, and x a variable. Then we can find a formula φ' (which differs from φ only in the choice of the quantified variables) such that*

(i) $\varphi \vdash \varphi'$ *and* $\varphi' \vdash \varphi$

(ii) t *is substitutable for* x *in* φ'.

Having developed what truth and falsity mean in a first order setting in the last chapter and having developed our formal model for deductions in this chapter, we are ready to see how the two interrelate. The interrelationship between the two will be at the heart of the statement of Gödel's Incompleteness Theorem.

Chapter 6

Soundness, Completeness, and Compactness

In a first-order setting, we have associated with a set of first-order formulas Γ two statements:

$$\Gamma \models \varphi \text{ and } \Gamma \vdash \varphi.$$

Recall that the statement on the left says that the set of formulas Γ logically implies φ (Definition 4.36). This means that every structure which satisfies every formula in Γ with $s : \mathcal{V} \longrightarrow \mathbb{U}$ will also satisfy φ with s. If $\Gamma \cup \{\varphi\}$ contains only sentences, then logical implication means that every structure that models Γ will also model φ. In other words, if the sentences of Γ are true in a structure, the statement φ must also be true in that structure.

On the other hand $\Gamma \vdash \varphi$ means that φ is a theorem of Γ, or equivalently, that there is a formal deduction (our model for a proof) of φ from Γ (i.e. φ was generated via our rule of inference from the formulas of Γ). So, $\Gamma \models \varphi$ is a statement about the *truth* of φ given our set of assumptions Γ whereas $\Gamma \vdash \varphi$ is a statement about the *deducibility* of φ from Γ. In this chapter, we will show how these two interrelate, and this interrelationship will get at many of the core ideas in Gödel's Incompleteness Theorem.

There are three key questions that we want to answer in this chapter.

(1) If we have a deduction ($\Gamma \vdash \boldsymbol{\varphi}$), does it follow that the formula deduced will be true in every structure in which the hypotheses (Γ) are satisfied? (This concept is known as *soundness*.)

(2) If we know that a formula holds true in every structure that satisfies a set of hypotheses ($\Gamma \vDash \boldsymbol{\varphi}$), is there a deduction of that formula from the set of hypotheses? (This concept is known as *completeness*.)

(3) If a statement is deducible, is it always deducible from a finite set of hypotheses? (This concept is known as *compactness*.)

We begin with soundness.

6.1 Soundness

An example of soundness is in order to more fully illustrate what our soundness theorem will say.

Example 6.1 *Using the language for number theory, we can easily demonstrate that* $\forall \boldsymbol{x}(\boldsymbol{x} < \boldsymbol{Sx}) \vdash \mathbf{0} < \boldsymbol{S}\mathbf{0}$. *This follows from our substitution logical axiom since* $\mathbf{0}$ *is substitutable for* $\boldsymbol{x}$ *in* $\boldsymbol{x} < \boldsymbol{Sx}$. *Since there is no inherent meaning behind any of the symbols, soundness says that since this deduction exists, in any structure in which* $\forall \boldsymbol{x}(\boldsymbol{x} < \boldsymbol{Sx})$ *is true* $\mathbf{0} < \boldsymbol{S}\mathbf{0}$ *must also be true. Of course in the structure of number theory it is true that the successor of every natural number is greater than that natural number. So soundness says that the successor of 0 must be greater than 0.*

However if we consider a different structure, say the structure $\mathfrak{S}$ where $\mathfrak{S}(\forall) = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \ldots\}$, $\mathfrak{S}(<) = \in$, $\mathfrak{S}(\mathbf{0}) = \emptyset$, $\mathfrak{S}(\boldsymbol{S}) = \mathcal{F}$ where $\mathcal{F}(A) = \{A\}$, $\mathfrak{S}(+) = \cap$, $\mathfrak{S}(\cdot) = \cap$, and $\mathfrak{S}(\boldsymbol{E}) = \cap$. Our deduction of course still holds, since the deduction only has to deal strictly with the formal language and not with any particular structure. Also, since $A \in \{A\} = \mathcal{F}(A)$ for every member of the universe, soundness says that it must also be the case that $\emptyset \in \mathcal{F}(\emptyset)$. Soundness says that $\mathbf{0} < \boldsymbol{S}\mathbf{0}$ must be true in every structure where $\forall \boldsymbol{x} \boldsymbol{x} < \boldsymbol{S}\boldsymbol{x}$ is true.

Now, for the formal statement of the Soundness Theorem.

Theorem 6.2 (Soundness Theorem) *If $\Gamma \vdash \boldsymbol{\varphi}$, then $\Gamma \models \boldsymbol{\varphi}$.*

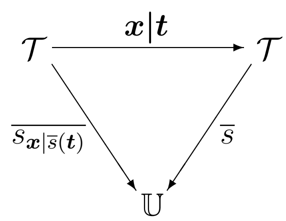
Before proving the Soundness Theorem, we will need a few lemmas.

Lemma 6.1.1 *For a structure $\mathfrak{S}$ and for $s : \mathcal{V} \longrightarrow \mathbb{U}$, for any variable $\boldsymbol{x}$, and terms $\boldsymbol{u}$ and $\boldsymbol{t}$, $\overline{s}(\boldsymbol{u}_{\boldsymbol{x}|\boldsymbol{t}}) = \overline{s_{x|\overline{s}(\boldsymbol{t})}}(\boldsymbol{u})$.*

Before proving the lemma, let us get a grasp on what it says. On the left side of the equation, we take the term $\boldsymbol{u}$ and substitute every incidence of the variable $\boldsymbol{x}$ with the term $\boldsymbol{t}$, we then evaluate $\overline{s}(\boldsymbol{u}_{\boldsymbol{x}|\boldsymbol{t}})$ which will be the image of the term $\boldsymbol{u}_{\boldsymbol{x}|\boldsymbol{t}}$ in the universe of $\mathfrak{S}$. Recall that for $a \in \mathbb{U}$,

$$s_{\boldsymbol{x}|a}(\boldsymbol{y}) = \begin{cases} a & \text{for } \boldsymbol{y} = \boldsymbol{x} \\ s(\boldsymbol{y}) & \text{for } \boldsymbol{y} \neq \boldsymbol{x}. \end{cases}$$

So, for $\overline{s_{x|\overline{s}(\boldsymbol{t})}}(\boldsymbol{u})$, we first calculate the image of the term $\boldsymbol{t}$ in the universe and then calculate the image $\overline{s_{x|\overline{s}(\boldsymbol{t})}}(\boldsymbol{u})$ in the universe where every incidence of $\boldsymbol{x}$ in $\boldsymbol{u}$ will be mapped to the image of $\boldsymbol{t}$ in the universe. Essentially, the equation states that it does not matter whether we substitute $\boldsymbol{t}$ for $\boldsymbol{x}$ and then map to the universe or whether we map to the universe and then substitute the image of $\boldsymbol{t}$ when calculating what the image of $\boldsymbol{u}$ is. The associated diagram is as follows where $\mathcal{T}$ is the set of terms for the specific first order language being used.



Proof: The argument is by induction on the form of the term $\boldsymbol{u}$. If $\boldsymbol{u}$ is a constant symbol, then $\boldsymbol{u}_{\boldsymbol{x}|\boldsymbol{t}} = \boldsymbol{u}$ so that $\overline{s}(\boldsymbol{u}_{\boldsymbol{x}|\boldsymbol{t}}) = \overline{s}(\boldsymbol{u}) = u^{\mathfrak{S}}$. But $\overline{s_{x|\overline{s}(\boldsymbol{t})}}(\boldsymbol{u}) = u^{\mathfrak{S}}$, so that our equality holds.

If $\boldsymbol{u} = \boldsymbol{x}$, then $\overline{s}(\boldsymbol{u}_{\boldsymbol{x}|\boldsymbol{t}}) = \overline{s}(\boldsymbol{t})$. On the other hand

$$\overline{s_{x|\overline{s}(\boldsymbol{t})}}(\boldsymbol{u}) = \overline{s_{x|\overline{s}(\boldsymbol{t})}}(\boldsymbol{x}) = s_{x|\overline{s}(\boldsymbol{t})}(\boldsymbol{x}) = \overline{s}(\boldsymbol{t})$$

so that our equation holds again. The induction step is hard to write, but by these last two cases and the discussion involving $\overline{s}$ on page 100 it is clear that equality will hold. ∎

This last lemma is useful for the next lemma.

Lemma 6.1.2 (Substitution Lemma) *If the term $\boldsymbol{t}$ is substitutable for the variable $\boldsymbol{x}$ in the wff $\boldsymbol{\varphi}$, then*

$$\models_{\mathfrak{S}} \boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}}[s] \text{ if and only if } \models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{\boldsymbol{x}|\overline{s}(\boldsymbol{t})}].$$

Proof: We proceed by induction on the set of wffs for which the lemma holds. If we have an atomic formula, say, $\boldsymbol{P u_1 u_2 \cdots u_n}$, then

$$\models_{\mathfrak{S}} (\boldsymbol{P u_1 u_2 \cdots u_n})_{\boldsymbol{x}|\boldsymbol{t}}[s] \text{ if and only if}$$

$$(\overline{s}((\boldsymbol{u_1})_{\boldsymbol{x}|\boldsymbol{t}}), \overline{s}((\boldsymbol{u_2})_{\boldsymbol{x}|\boldsymbol{t}}), \ldots, \overline{s}((\boldsymbol{u_n})_{\boldsymbol{x}|\boldsymbol{t}})) \in P^{\mathfrak{S}}.$$

By the preceding lemma, this last statement is true if and only if

$$(\overline{s_{\boldsymbol{x}|\overline{s}(\boldsymbol{t})}}(\boldsymbol{u_1}), \overline{s_{\boldsymbol{x}|\overline{s}(\boldsymbol{t})}}(\boldsymbol{u_2}), \ldots, \overline{s_{\boldsymbol{x}|\overline{s}(\boldsymbol{t})}}(\boldsymbol{u_n})) \in P^{\mathfrak{S}} \text{ if and only if}$$

$$\models_{\mathfrak{S}} \boldsymbol{P u_1 u_2 \cdots u_n}[s_{\boldsymbol{x}|\overline{s}(\boldsymbol{t})}].$$

This concludes the base case of induction since first-order wffs are freely generated from the set of atomic formulas.

Suppose now that $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$ are wffs that fulfill the lemma i.e., we assume that $\boldsymbol{t}$ is substitutable for $\boldsymbol{x}$ in both formulas and that

$$\models_{\mathfrak{S}} \boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}}[s] \text{ if and only if } \models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{\boldsymbol{x}|\overline{s}(\boldsymbol{t})}] \text{ and that}$$

$$\models_{\mathfrak{S}} \boldsymbol{\psi}_{\boldsymbol{x}|\boldsymbol{t}}[s] \text{ if and only if } \models_{\mathfrak{S}} \boldsymbol{\psi}[s_{\boldsymbol{x}|\overline{s}(\boldsymbol{t})}].$$

The lemma will clearly hold for the cases $\boldsymbol{(\varphi \to \psi)}$ and $\boldsymbol{(\neg\varphi)}$ by Definition 4.22.

Consider now $\forall \boldsymbol{y\varphi}$. There are two possibilities. Either $\boldsymbol{x}$ occurs free in $\boldsymbol{\varphi}$ or it does not. If it does not, then $\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}} = \boldsymbol{\varphi}$ by the discussion on pages 130-131 and by the definition of free variable. Also, $s\big|_{\mathcal{F}_\varphi} = s_{\boldsymbol{x}|\overline{s(\boldsymbol{t})}}\big|_{\mathcal{F}_\varphi}$ where $\mathcal{F}_\varphi$ is as in Theorem 4.27. By Theorem 4.27 then, we can say that

$$\models_{\mathfrak{S}} \forall \boldsymbol{y} \boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}}[s] \text{ if and only if } \models_{\mathfrak{S}} \forall \boldsymbol{y\varphi}[s_{\boldsymbol{x}|\overline{s}(\boldsymbol{t})}]$$

If $\boldsymbol{x}$ does occur free in $\boldsymbol{\varphi}$, we first note that we are assuming that $\boldsymbol{t}$ is substitutable for $\boldsymbol{x}$ in $\forall \boldsymbol{y\varphi}$ otherwise the statement of the lemma holds as a vacuously true statement. Hence, by our discussion of substitutability in the last chapter $\boldsymbol{y}$ must not occur in the term $\boldsymbol{t}$. So, $\overline{s}(\boldsymbol{t}) = \overline{s_{y|u}}(\boldsymbol{t})$ for all $u \in \mathbb{U}$. If $\boldsymbol{x} = \boldsymbol{y}$, then $(\forall \boldsymbol{y\varphi})_{\boldsymbol{x}|\boldsymbol{t}} = \forall \boldsymbol{y\varphi}$, and $\boldsymbol{x}$ does not occur free in this formula.

An application of Theorem 4.27 then guarantees the statement of the lemma. If $\boldsymbol{x} \neq \boldsymbol{y}$, then we have that $(\forall \boldsymbol{y}\boldsymbol{\varphi})_{\boldsymbol{x}|\boldsymbol{t}} = \forall \boldsymbol{y}\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}}$. Notice that

$$s_{\boldsymbol{y}|u_{\boldsymbol{x}|\overline{s}(\boldsymbol{t})}}(\boldsymbol{v}) = s_{\boldsymbol{x}|\overline{s}(\boldsymbol{t})_{\boldsymbol{y}|u}}(\boldsymbol{v}) = \begin{cases} u & \text{if } \boldsymbol{v} = \boldsymbol{y} \text{ and } \boldsymbol{v} \neq \boldsymbol{x} \\ \overline{s}(\boldsymbol{t}) & \text{if } \boldsymbol{v} \neq \boldsymbol{y} \text{ and } \boldsymbol{v} = \boldsymbol{x} \\ s(v) & \text{if } \boldsymbol{v} \neq \boldsymbol{y} \text{ and } \boldsymbol{v} \neq \boldsymbol{x} \end{cases}$$

In the above function, there will of course be no case when $\boldsymbol{y} = \boldsymbol{x}$ since these are not the same variable by assumption. Now,

$$\models_{\mathfrak{S}} (\forall \boldsymbol{y}\boldsymbol{\varphi})_{\boldsymbol{x}|\boldsymbol{t}}[s] \text{ iff } \models_{\mathfrak{S}} \forall \boldsymbol{y}\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}}[s] \text{ iff}$$

$$\models_{\mathfrak{S}} \boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}}[s_{\boldsymbol{y}|u}] \text{ for all } u \in \mathbb{U} \text{ iff}$$

$$\models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{\boldsymbol{y}|u_{\boldsymbol{x}|\overline{s_{\boldsymbol{y}|u}}(\boldsymbol{t})}}] \text{ for all } u \in \mathbb{U}.$$

Since $\overline{s}(\boldsymbol{t}) = \overline{s_{y|u}}(\boldsymbol{t})$ for all $u \in \mathbb{U}$, the last statement is equivalent to

$$\models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{\boldsymbol{y}|u_{\boldsymbol{x}|\overline{s}(\boldsymbol{t})}}] \text{ for all } u \in \mathbb{U} \text{ iff } \models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{\boldsymbol{x}|\overline{s}(\boldsymbol{t})_{\boldsymbol{y}|u}}] \text{ for all } u \in \mathbb{U} \text{ iff}$$

$$\models_{\mathfrak{S}} \forall \boldsymbol{y}\boldsymbol{\varphi}[s_{\boldsymbol{x}|\overline{s}(\boldsymbol{t})}].$$

Since then the set of wffs that fulfill the lemma includes the atomic formulas and is closed under the formula building operations, we conclude that the lemma holds for all formulas $\boldsymbol{\varphi}$. ∎

We will need the next lemma to form a link between tautological implication and logical implication.

Lemma 6.1.3 *If* Γ *tautologically implies* $\boldsymbol{\varphi}$*, then* Γ *logically implies* $\boldsymbol{\varphi}$

Proof: Assume that Γ tautologically implies $\boldsymbol{\varphi}$ (recall the discussion of tautologies in the last chapter). Let a structure $\mathfrak{S}$ and a mapping

$s : \mathcal{V} \longrightarrow \mathbb{U}$ be given. Define a truth assignment on the set of prime formulas by

$$v(\boldsymbol{\alpha}) = T \text{ if and only if } \models_{\mathfrak{S}} \boldsymbol{\alpha}[s].$$

We seek to show that for any formula $\boldsymbol{\varphi}$,

$$\overline{v}(\boldsymbol{\varphi}) = T \text{ if and only if } \models_{\mathfrak{S}} \boldsymbol{\varphi}[s].$$

Since a wff of the form $\forall \boldsymbol{x}\boldsymbol{\psi}$ is a prime formula, it remains to be shown that the statement holds for formulas of the form $(\boldsymbol{\psi} \rightarrow \boldsymbol{\chi})$ or $(\neg\boldsymbol{\psi})$. However, if the statement holds for both $\boldsymbol{\psi}$ and $\boldsymbol{\chi}$, then the statement must hold for each of $(\boldsymbol{\psi} \rightarrow \boldsymbol{\chi})$ and $(\neg\boldsymbol{\psi})$ by the recursive construction of $\overline{v}$ and by Definition 4.29. So, by induction, the statement holds for all formulas.

Thus, if we have a structure $\mathfrak{S}$ and a mapping $s : \mathcal{V} \longrightarrow \mathbb{U}$ satisfying (Definition 4.22) every member of Γ, the truth assignment v will satisfy (Definition 2.35) every member of Γ. So, since Γ tautologically implies $\boldsymbol{\varphi}$, $\overline{v}(\boldsymbol{\varphi}) = T$, and this is so if and only if $\models_{\mathfrak{S}} \boldsymbol{\varphi}[s]$. Hence, Γ logically implies $\boldsymbol{\varphi}$.

∎

Lemma 6.1.4 *The set of logical axioms* Λ *is valid.*

Proof: We prove that each group of logical axioms is valid and then prove that any generalization of a valid formula will also be valid.

Axiom Group 1 (Tautologies): By definition $\emptyset$ tautologically implies a tautology. By the preceding lemma, $\emptyset$ logically implies the tautology, and this is exactly what is meant by a valid formula by definition.

Axiom Group 2 (Substitution): Assume that $\boldsymbol{t}$ is substitutable for $\boldsymbol{x}$ in the formula $\boldsymbol{\varphi}$. Let $\mathfrak{S}$ be an arbitrary structure such that $\models_{\mathfrak{S}} \forall \boldsymbol{x}\boldsymbol{\varphi}[s]$. This is so if and only if $\models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{\boldsymbol{x}|u}]$ for all $u \in \mathbb{U}$. Now $\overline{s}(\boldsymbol{t}) \in \mathbb{U}$ so that we must have $\models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{\boldsymbol{x}|\overline{s}(\boldsymbol{t})}]$. By the Substitution Lemma, this last statement is equivalent to $\models_{\mathfrak{S}} \boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}}[s]$. By Definition 4.22, $\models_{\mathfrak{S}} (\forall \boldsymbol{x}\boldsymbol{\varphi} \rightarrow \boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}})[s]$. Since $\mathfrak{S}$ and $s : \mathcal{V} \longrightarrow \mathbb{U}$ were arbitrarily chosen, $(\forall \boldsymbol{x}\boldsymbol{\varphi} \rightarrow \boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}})$ is a valid formula.

Axiom Group 3: Note that

$$\models_{\mathfrak{S}} (\forall \boldsymbol{x}(\boldsymbol{\varphi} \rightarrow \boldsymbol{\psi}) \rightarrow (\forall \boldsymbol{x}\boldsymbol{\varphi} \rightarrow \forall \boldsymbol{x}\boldsymbol{\psi}))[s] \text{ iff}$$

$$\not\models_{\mathfrak{S}} \forall \boldsymbol{x}(\boldsymbol{\varphi} \rightarrow \boldsymbol{\psi})[s] \text{ or } \models_{\mathfrak{S}} (\forall \boldsymbol{x}\boldsymbol{\varphi} \rightarrow \forall \boldsymbol{x}\boldsymbol{\psi})[s]$$

$$\text{But } \models_{\mathfrak{S}} (\forall \boldsymbol{x}\boldsymbol{\varphi} \rightarrow \forall \boldsymbol{x}\boldsymbol{\psi})[s] \text{ iff } \not\models_{\mathfrak{S}} \forall \boldsymbol{x}\boldsymbol{\varphi}[s] \text{ or } \models_{\mathfrak{S}} \forall \boldsymbol{x}\boldsymbol{\psi}[s].$$

Now assume that $\models_{\mathfrak{S}} \forall \boldsymbol{x}(\boldsymbol{\varphi} \rightarrow \boldsymbol{\psi})[s]$ and that $\models_{\mathfrak{S}} \forall \boldsymbol{\varphi}[s]$. This statement holds if and only if

$$\models_{\mathfrak{S}} (\boldsymbol{\varphi} \rightarrow \boldsymbol{\psi})[s_{\boldsymbol{x}|u}] \text{ and } \models_{\mathfrak{S}} \varphi[s_{\boldsymbol{x}|u}] \text{ for every } u \in \mathbb{U} \text{ iff}$$

$$\not\models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{\boldsymbol{x}|u}] \text{ for some } u \in \mathbb{U} \text{ or } \models_{\mathfrak{S}} \boldsymbol{\psi}[s_{\boldsymbol{x}|u}] \text{ for all } u \in \mathbb{U}, \text{ and}$$

$$\models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{\boldsymbol{x}|u}] \text{ for all } u \in \mathbb{U}.$$

This last statement implies that $\models_{\mathfrak{S}} \boldsymbol{\psi}[s_{\boldsymbol{x}|u}]$ for all $u \in \mathbb{U}$, or equivalently $\models_{\mathfrak{S}} \forall \boldsymbol{x}\boldsymbol{\psi}[s]$. Using the equivalence discussed at the beginning of this case, we can say that

$$\models_{\mathfrak{S}} (\forall \boldsymbol{x}(\boldsymbol{\varphi} \rightarrow \boldsymbol{\psi}) \rightarrow (\forall \boldsymbol{x}\boldsymbol{\varphi} \rightarrow \forall \boldsymbol{x}\boldsymbol{\psi}))[s].$$

Thus, axioms of this form are valid.

Axiom Group 4: Suppose that $\boldsymbol{x}$ does not occur free in $\boldsymbol{\varphi}$. Furthermore, suppose that $\models_{\mathfrak{S}} \boldsymbol{\varphi}[s]$. Now, s and $s_{\boldsymbol{x}|u}$ agree on all free variables in $\boldsymbol{\varphi}$ for all $u \in \mathbb{U}$ since $\boldsymbol{x}$ does not occur free in $\boldsymbol{\varphi}$. Hence by Theorem 4.27,

$$\models_{\mathfrak{S}} \boldsymbol{\varphi}[s] \text{ iff } \models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{\boldsymbol{x}|u}],$$

and this statement holds for all $u \in \mathbb{U}$. Hence, $\models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{\boldsymbol{x}|u}]$ for all $u \in \mathbb{U}$, and this statement is equivalent to $\models_{\mathfrak{S}} \forall \boldsymbol{x}\boldsymbol{\varphi}[s]$. Since $\models_{\mathfrak{S}} (\boldsymbol{\varphi} \rightarrow \forall \boldsymbol{x}\boldsymbol{\varphi})[s]$ holds if and only if $\not\models_{\mathfrak{S}} \boldsymbol{\varphi}[s]$ or $\models_{\mathfrak{S}} \forall \boldsymbol{x\varphi}[s]$, our work shows that $\models_{\mathfrak{S}} \boldsymbol{(\varphi \rightarrow \forall \varphi)}[s]$. Hence, $\boldsymbol{(\varphi \rightarrow \forall \varphi)}$ is a valid formula.

Axiom Group 5: This is trivial since $\models_{\mathfrak{S}} \boldsymbol{x} \approx \boldsymbol{x}[s]$ if and only if $s(\boldsymbol{x}) = s(\boldsymbol{x})$, which is always true.

Axiom Group 6: Similar to axiom group 3, it suffices to show that $\models_{\mathfrak{S}} \boldsymbol{x} \approx \boldsymbol{y}[s]$ and $\models_{\mathfrak{S}} \boldsymbol{\varphi}[s]$ implies that $\models_{\mathfrak{S}} \boldsymbol{\varphi'}[s]$ where $\boldsymbol{\varphi'}$ is obtained from $\boldsymbol{\varphi}$ by replacing some, but not necessarily all instances of $\boldsymbol{x}$ in $\boldsymbol{\varphi}$ with $\boldsymbol{y}$. Now, $\models_{\mathfrak{S}} \boldsymbol{x} \approx \boldsymbol{y}[s]$ if and only if $s(\boldsymbol{x}) = s(\boldsymbol{y})$. So, it is clear that for any term $\boldsymbol{t'}$ obtained by replacing instances of $\boldsymbol{x}$ in term $\boldsymbol{t}$ with the variable $\boldsymbol{y}$, that $\overline{s}(\boldsymbol{t}) = \overline{s}(\boldsymbol{t'})$ (we could use induction to show this more rigorously). If $\boldsymbol{\varphi} = \boldsymbol{t_1} \approx \boldsymbol{t_2}$ (for terms $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$), then

$$\models_{\mathfrak{S}} \boldsymbol{\varphi}[s] \text{ iff } \overline{s}(\boldsymbol{t_1}) = \overline{s}(\boldsymbol{t_2}) \text{ iff}$$

$$\overline{s}(\boldsymbol{t'_1}) = \overline{s}(\boldsymbol{t'_2}) \text{ iff } \models_{\mathfrak{S}} \boldsymbol{t'_1} \approx \boldsymbol{t'_2}[s]$$

and our result holds. The case for $\boldsymbol{\varphi} = \boldsymbol{Pt_1t_2 \cdots t_n}$ and $\boldsymbol{\varphi'} = \boldsymbol{Pt'_1t'_2 \cdots t'_n}$ is very similar. We can complete the argument for all remaining cases for $\boldsymbol{\varphi}$ by a simple induction argument.

Having now shown that all formulas in the various axiom groups are valid, all that remains is to show that any generalization of any of these formulas must also be valid. Begin with the assumption that $\vDash \boldsymbol{\varphi}$ for some formula $\boldsymbol{\varphi}$. Assume that $\mathfrak{S}$ is a structure and $s : \mathcal{V} \longrightarrow \mathbb{U}$. Then $\models_{\mathfrak{S}} \boldsymbol{\varphi}[s_{\boldsymbol{x}|u}]$ for all $u \in \mathbb{U}$ since $\vDash \boldsymbol{\varphi}$. Thus, $\models_{\mathfrak{S}} \forall \boldsymbol{\varphi}[s]$. By definition of generalization then, and by induction, any generalization of one of the logical axioms in the six groups (which are each valid as shown above) will itself be valid. By definition of the set of logical axioms then, the entire set of logical axioms is valid. ∎

Now, our strategy for proving the Soundness Theorem will be as follows: the theorems of Γ are generated via our rule of inference $\mathcal{I}$ from the set $\Gamma \cup \Lambda$

where Λ is the set of logical axioms. We can thus proceed by induction on the set of wffs $\{\varphi|\Gamma \vdash \varphi \text{ implies } \Gamma \vDash \varphi\}$.

Proof: (Soundness Theorem for First-Order Logic) If φ is a logical axiom, then $\vDash \varphi$ as shown in the previous lemma. So, of course $\Gamma \vDash \varphi$.

If φ is in Γ, then of course $\Gamma \vDash \varphi$ by definition.

Suppose that $\varphi = \mathcal{I}(\psi, (\psi \rightarrow \varphi)$ where we assume that we have already shown that the theorem holds for ψ and for $(\psi \rightarrow \varphi)$. So, assume that $\mathfrak{S}$ is a structure and $s : \mathcal{V} \longrightarrow \mathbb{U}$ such that every member of Γ is satisfied in $\mathfrak{S}$ by s. We wish to show that $\models_{\mathfrak{S}} \varphi[s]$. Now, $\models_{\mathfrak{S}} (\psi \rightarrow \varphi)[s]$ and $\models_{\mathfrak{S}} \psi[s]$, so that as we have argued in many previous results, $\models_{\mathfrak{S}} \varphi[s]$ as desired. ∎

Recall that a set of formulas Γ is said to be consistent if there is no formula φ such that both φ and $(\neg\varphi)$ are deducible from Γ.

Definition 6.3 *The set of formulas* Γ *is said to be* **satisfiable** *if there is some structure* $\mathfrak{S}$ *and some* $s : \mathcal{V} \longrightarrow \mathbb{U}$ *such that every member of* Γ *is satisfied in* $\mathfrak{S}$ *by* s.

Corollary 6.3.1 *If* Γ *is satisfiable, then* Γ *is consistent.*

Proof: The proof is accomplished by contraposition. Suppose Γ is inconsistent. Then by definition there is a formula φ such that both $\Gamma \vdash \varphi$ and $\Gamma \vdash (\neg\varphi)$. By soundness, we have that $\Gamma \vDash \varphi$ and that $\Gamma \vDash (\neg\varphi)$. So, if Γ were satisfiable, for some structure $\mathfrak{S}$ and some $s : \mathcal{V} \longrightarrow \mathbb{U}$, we would have every member of Γ satisfied in $\mathfrak{S}$ by s. But then, we must have

$$\models_{\mathfrak{S}} \varphi[s] \text{ and } \models_{\mathfrak{S}} (\neg\varphi)[s]$$

. However, this is impossible since

$$\models_{\mathfrak{S}} (\neg\varphi)[s] \text{ iff } \not\models_{\mathfrak{S}} \varphi[s].$$

So in fact, Γ is not satisfiable, and by contraposition, the corollary holds. ∎

Corollary 6.3.2 *There is an alphabetic variant of φ , φ' such that φ and φ' are logically equivalent.*

Proof: By Theorem 5.22, there is an alphabetic variant φ' of φ such that $\varphi \vdash \varphi'$ and $\varphi' \vdash \varphi$. By soundness, $\varphi \models \varphi'$ and $\varphi' \models \varphi$. ∎

This corollary reflects are intuition that if a certain set of statements can be modeled by a real structure (in the real world), then that real structure must have no contradictions in it.

Having addressed soundness, we now examine its corollary, completeness.

6.2 Completeness

Soundness essentially says that if you have a proof of a statement, then that statement must be true. On the other hand, completeness says that any fact that is true given a set of assumptions must have a proof. These nuances of this theorem may not be readily apparent since the way we know something is true in mathematics is through a proof (soundness). Examining a statement that is not known to be true but is believed to be true will clarify what completeness means.

Example 6.4 *Goldbach's Conjecture is a well-known open question in number theory. It states that every even integer greater than 2 can be written as the sum of two prime numbers. Without going into the gory details, this statement could be expressed in our formal language that we have been using for number theory. To date, there is no proof for this statement but a great amount of* *empirical evidence indicating that it is probably true (of course one counterexample would show that it is in general, false). So, what soundness says is that if we let* Γ *be a set of axioms from which we are attempting to prove statements of number theory, expressed in the formal language, if Goldbach's Conjecture is logically implied by our set of axioms (*$\Gamma \models \boldsymbol{\varphi}$ *where* $\boldsymbol{\varphi}$ *is Goldbach's Conjecture expressed in the formal language), then completeness says that there* **must** *be a proof (formally, a deduction) of Goldbach's Conjecture from our set of axioms. Thus, since there is a great amount of empirical evidence that Goldbach's Conjecture is true, there is hope, given the Completeness Theorem we are about to show, that there is in fact a day when a formal proof for Goldbach's conjecture will be found.*

We now state two versions of the Completeness Theorem.

Theorem 6.5 (Completeness Theorem (Version 1))

$$\textit{If } \Gamma \models \boldsymbol{\varphi}, \textit{ then } \Gamma \vdash \boldsymbol{\varphi}.$$

Theorem 6.6 (Completeness Theorem (Version 2)) *Any consistent set of formulas is satisfiable.*

The method of our proof will be to first show that the two versions are equivalent and then we will give a proof of Version 2.

Theorem 6.7 *The two versions of the Completeness Theorem are equivalent.*

Proof: Assume Version 1 holds and let Γ be a consistent set of formulas. Note that the implication "If $\boldsymbol{\gamma} \in \emptyset$, then $\models_{\mathfrak{S}} \boldsymbol{\gamma}[s]$," is true for every structure $\mathfrak{S}$ and every $s : \mathcal{V} \longrightarrow \mathbb{U}$. Now, it is not the case that both

$$\models_{\mathfrak{S}} \boldsymbol{\varphi}[s] \text{ and } \models_{\mathfrak{S}} (\neg\boldsymbol{\varphi})[s]$$

for any formula φ by Definition 4.30. By the contrapositive of the soundness theorem, it is not the case that both $\vdash \varphi$ and $\vdash (\neg\varphi)$, so that $\emptyset$ is a consistent. However, by the true implication mentioned above, $\emptyset$ is satisfiable by every structure $\mathfrak{S}$ and every $s : \mathcal{V} \longrightarrow \mathbb{U}$.

Suppose Γ is a consistent set. We have just taken care of the case when $\Gamma = \emptyset$. Suppose now that $\Gamma \neq \emptyset$. So, there is $\gamma_0 \in \Gamma$. Since Γ is consistent, it is not the case that both $\Gamma \vdash \gamma_0$ and $\Gamma \vdash (\neg\gamma_0)$. By the contrapositive of our hypothesis, either $\Gamma \nvDash \gamma_0$ or $\Gamma \nvDash (\neg\gamma_0)$. The former case is not possible by soundness and the fact that $\Gamma \vdash \gamma_0$ since $\gamma_0 \in \Gamma$. So, it must be the case that $\Gamma \nvDash (\neg\gamma_0)$. By definition, there is a structure $\mathfrak{S}$ and a $s : \mathcal{V} \longrightarrow \mathbb{U}$ such that every member of Γ is satisfied by s in $\mathfrak{S}$, but such that $\not\models_{\mathfrak{S}} (\neg\gamma_0)[s]$. The important piece here is that we have found a structure and a function on the variables within that structure that satisfies Γ, and we have shown that Version 2 holds if Version 1 holds.

Assume now that Version 2 holds. Assume that $\Gamma \vDash \varphi$ for a set of formulas $\Gamma \cup \{\varphi\}$. Now either Γ is consistent or Γ is inconsistent. If Γ is inconsistent, then by Theorem 5.17, $\Gamma \vdash \varphi$. Suppose now that Γ is consistent. Note that by Theorem 5.16, if $\Gamma \cup \{(\neg\varphi)\}$ is inconsistent, then $\Gamma \vdash (\neg(\neg\varphi))$. This is a proof by contradiction at the formal level. Suppose by way of contradiction (at the meta-level) that $\Gamma \cup \{(\neg\varphi)\}$ is a consistent set of formulas. Then by our assumption that Version 2 holds, there is a structure $\mathfrak{S}$ and $s : \mathcal{V} \longrightarrow \mathbb{U}$ such that $\mathfrak{S}$ satisfies every member of $\Gamma \cup \{(\neg\varphi)\}$ with s. That is, $\mathfrak{S}$ satisfies every member of Γ with s and $\models_{\mathfrak{S}} (\neg\varphi)[s]$. However, this situation is impossible by our assumption that $\Gamma \vDash \varphi$ which would imply that $\models_{\mathfrak{S}} \varphi[s]$. Hence, in fact $\Gamma \cup \{(\neg\varphi)\}$ is an inconsistent set, and we can conclude that $\Gamma \vdash (\neg(\neg\varphi))$. Since $((\neg(\neg\varphi)) \rightarrow \varphi)$ is a tautology, we may say that $\Gamma \vdash \varphi$

by our rule of inference. Thus, Version 2 implies Version 1 and the two versions of completeness are equivalent. ∎

We now prove Version 2 of the Completeness Theorem.

Proof: (Version 2 of the Completeness Theorem) This proof is involved and so it is organized by headings indicating the purpose behind each section of the proof.

Let Γ be a consistent set of wffs in the language at hand.

Step 1: Enriching the Language with Constants

Create a new first-order language by using every symbol in the language in which Γ is consistent and then add to this language a countably infinite number of new distinct constant symbols. For instance, if $\boldsymbol{c_0}, \boldsymbol{c_1}, \ldots$ is an enumeration of the countably many constant symbols in the old language, we add the constant symbols $\boldsymbol{c_{-1}}, \boldsymbol{c_{-2}}, \ldots$ to our language recognizing them as all distinct from our original list of constants and distinct from each other. After this enrichment, it is clear that Γ is still a set of formulas in this language. We also wish to show that Γ is still a consistent set in this enriched language.

Suppose Γ is inconsistent in this language. Then there is a formula in the new language $\boldsymbol{\chi}$ such that $\Gamma \vdash \boldsymbol{\chi}$ and $\Gamma \vdash (\neg\boldsymbol{\chi})$. Only finitely many of the new constants are involved in each deduction. So, by repeated application of 5.18.1, we may replace any new constants used in the deductions from Γ of $\boldsymbol{\chi}$ and $(\neg\boldsymbol{\chi})$ with variables, and we may do this so that $\Gamma \vdash \boldsymbol{\chi'}$ and $\Gamma \vdash (\neg\boldsymbol{\chi'})$ where $\boldsymbol{\chi'}$ is χ where any instances of new constants have been replaced by variables. Since we have the exact same variables as in the original language, $\boldsymbol{\chi'}$ and $(\neg\boldsymbol{\chi'})$ are formulas in our original language. But since Γ is consistent in the old language, it is impossible that we have $\Gamma \vdash \boldsymbol{\chi'}$ and $\Gamma \vdash (\neg\boldsymbol{\chi'})$. So, what we supposed is false and Γ is consistent in the enriched language.

Step 2: Accounting for Counterexamples to All Possible Formulas

Since we are dealing with a countable language, we may enumerate all variables in the enriched language and also all formulas. Thus, we may form the pairings $(\boldsymbol{v_1}, \boldsymbol{\varphi_1})$, $(\boldsymbol{v_2}, \boldsymbol{\varphi_2})$, ... We now let

$$\boldsymbol{\theta_1} = ((\neg\forall \boldsymbol{v_1}\boldsymbol{\varphi_1}) \rightarrow (\neg\boldsymbol{\varphi_1}_{\boldsymbol{v_1}|\boldsymbol{d_1}}))$$

where $\boldsymbol{d_1}$ is the first of the new constant symbols not occurring in $\boldsymbol{\varphi_1}$. By recursion, we may define

$$\boldsymbol{\theta_n} = ((\neg\forall \boldsymbol{v_n}\boldsymbol{\varphi_n}) \rightarrow (\neg\boldsymbol{\varphi_n}_{\boldsymbol{v_n}|\boldsymbol{d_n}}))$$

where $\boldsymbol{d_n}$ is the first of the new constant symbols not occurring in $\boldsymbol{\varphi_n}$ or in $\boldsymbol{\theta_k}$ for any $k < n$. The idea behind including all of these formulas is that each formula provides the structure necessary to account for a counterexample to $\boldsymbol{\varphi_i}$ if there is one in *some* structure $\mathfrak{S}$. That is, if $\not\models_{\mathfrak{S}} \forall \boldsymbol{v_i}\boldsymbol{\varphi_i}[s]$ for some structure $\mathfrak{S}$ and some $s : \mathcal{V} \longrightarrow \mathbb{U}$, then this will be true if and only if there is $u \in \mathbb{U}$ such that $\not\models_{\mathfrak{S}} \boldsymbol{\varphi_i}[s_{\boldsymbol{v_i}|u}]$. This is so iff $u \in \mathbb{U}$ such that $\models_{\mathfrak{S}} (\neg\boldsymbol{\varphi_i})[s_{\boldsymbol{v_i}|u}]$. As long as $\overline{s}(\boldsymbol{d_i}) = d_i^{\mathfrak{S}} = u$, then by the Substitution Lemma,

$$\models_{\mathfrak{S}} (\neg\boldsymbol{\varphi_i})[s_{\boldsymbol{v_i}|u}] \text{ iff } \models_{\mathfrak{S}} (\neg\boldsymbol{\varphi_i}_{\boldsymbol{v_i}|\boldsymbol{c_i}})[s].$$

Now, let $\Theta = \{\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \ldots\}$. We want to show that $\Gamma \cup \Theta$ is consistent. Suppose not. Then because deductions are finite, for some $m \geq 0$, $\Gamma \cup \Theta_{m+1}$ is inconsistent where $\Theta_{m+1} = \{\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \ldots, \boldsymbol{\theta_m}, \boldsymbol{\theta_{m+1}}\}$. Take the least such m. By Theorem 5.16, $\Gamma \cup \Theta_m \vdash (\neg\boldsymbol{\theta_{m+1}})$ (if $m = 0$, we define $\Theta_m = \emptyset$). Note that $\boldsymbol{\theta_{m+1}} = ((\neg\forall \boldsymbol{x}\boldsymbol{\varphi}) \rightarrow (\neg\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{d}}))$ for some variable $\boldsymbol{x}$, some formula $\boldsymbol{\varphi}$ in the enriched language, and some constant $\boldsymbol{d}$ that does not occur in $\boldsymbol{\varphi}$ nor an any $\boldsymbol{\theta_k}$ for $1 \leq k \leq m$. Now, $(\neg\boldsymbol{\theta_{m+1}})$ tautologically implies the formulas

$(\neg\forall \boldsymbol{x}\boldsymbol{\varphi})$ and $\boldsymbol{\varphi_{x|d}}$ since if $(\neg\boldsymbol{\theta_{m+1}})$ is true, $\boldsymbol{\theta_{m+1}}$ is false and this is so if and only if $(\neg\forall \boldsymbol{x}\boldsymbol{\varphi})$ and $\boldsymbol{\varphi_{x|d}}$ are true. By Rule T, we have that

$$\Gamma \cup \Theta_m \vdash (\neg\forall \boldsymbol{x}\boldsymbol{\varphi}) \text{ and}$$

$$\Gamma \cup \Theta_m \vdash \boldsymbol{\varphi_{x|d}}.$$

By Corollary 5.18.2, since $\boldsymbol{d}$ does not occur in $\boldsymbol{\varphi}$ nor in any formula in $\Gamma \cup \Theta_m$, we have $\Gamma \cup \Theta_m \vdash \forall \boldsymbol{x}\boldsymbol{\varphi}$ so that we have both this deduction and $\Gamma \cup \Theta_m \vdash (\neg\forall \boldsymbol{x}\boldsymbol{\varphi})$. But then $\Gamma \cup \Theta_m$ must be an inconsistent set by definition, which contradicts the leastness of m if $m \geq 1$ or the consistency of Γ if $m = 0$ ($\Theta_m = \emptyset$ in this case). Thus, we must have that the set $\Gamma \cup \Theta$ is consistent.

Step 3: Extending $\boldsymbol{\Gamma \cup \Theta}$ to a Maximal Consistent Set

Since $\Gamma \cup \Theta$ is a consistent set, there is no formula $\boldsymbol{\chi}$ such that $\Gamma \cup \Theta \vdash \boldsymbol{\chi}$ and $\Gamma \cup \Theta \vdash (\neg\boldsymbol{\chi})$. So, by Theorem 5.6, there is no formula $\boldsymbol{\chi}$ such that $\Gamma \cup \Theta \cup \Lambda$ tautologically implies both $\boldsymbol{\chi}$ and $(\neg\boldsymbol{\chi})$. Thus, for every formula $\boldsymbol{\chi}$ in the language, there is a truth assignment v on the set of prime formulas such that v satisfies every member of $\Gamma \cup \Theta \cup \Lambda$ but such that either $\overline{v}(\boldsymbol{\chi}) = F$ or $\overline{v}((\neg\boldsymbol{\chi})) = F$. The important piece here is that there is a truth assignment v that satisfies every member of $\Gamma \cup \Theta \cup \Lambda$. Given this truth assignment, define the following set:

$$\Delta = \{\boldsymbol{\varphi} : \overline{v}(\boldsymbol{\varphi}) = T\}.$$

Clearly $\Gamma \cup \Theta \cup \Lambda \subseteq \Delta$ since v satisfies every member of $\Gamma \cup \Theta \cup \Lambda$. Now, take any formula of the language $\boldsymbol{\psi}$. Either $\overline{v}(\boldsymbol{\psi}) = T$ or $\overline{v}(\boldsymbol{\psi}) = F$ (which is equivalent to $\overline{v}((\neg\boldsymbol{\psi})) = T$). So, either $\boldsymbol{\psi} \in \Delta$ or $(\neg\boldsymbol{\psi}) \in \Delta$, but not both, for every formula $\boldsymbol{\psi}$ in our enriched language.

All that remains to be shown for this step is that Δ is a consistent set of formulas. Suppose by way of contradiction that $\Delta \vdash \boldsymbol{\chi}$ and $\Delta \vdash (\neg\boldsymbol{\chi})$ for some formula $\boldsymbol{\chi}$ of the language. Since $\Lambda \subseteq \Delta$, we may say that by Theorem 5.6 that Δ tautologically implies both $\boldsymbol{\chi}$ and $(\neg\boldsymbol{\chi})$. Since the truth assignment v satisfies every member of Δ, it must satisfy both $\boldsymbol{\chi}$ and $(\neg\boldsymbol{\chi})$, an impossibility. Thus, Δ must be a consistent set.

We can demonstrate as well that Δ is deductively closed. Suppose $\Delta \vdash \boldsymbol{\varphi}$. Then by the consistency of Δ, $\Delta \nvdash (\neg\boldsymbol{\varphi})$. By Theorem 5.6, Δ does not tautologically imply $(\neg\boldsymbol{\varphi})$. Now, the truth assignment v that defines Δ of course satisfies Δ. Thus, $\overline{v}((\neg\boldsymbol{\varphi})) = F$ since Δ does not tautologically imply $(\neg\boldsymbol{\varphi})$. Thus, $(\neg\boldsymbol{\varphi}) \notin \Delta$ by the definition of Δ. Since either $\boldsymbol{\varphi}$ or $(\neg\boldsymbol{\varphi})$ is Δ, we must have $\boldsymbol{\varphi} \in \Delta$, and hence, Δ is deductively closed.

Step 4: Constructing a Preliminary Structure for Satisfaction

Refer the languages we have been discussing as follows: let $\mathcal{L}_0$ represent the original language (in which Γ is consistent) and let $\mathcal{L}_1$ represent the enriched language after adding all of the additional constant symbols. Now, let $\mathcal{L}_1'$ be the language that is the same as $\mathcal{L}_1$ but where all instances of $\boldsymbol{\approx}$ (if there are any) are replaced by a two place predicate symbol $\boldsymbol{E}$. We now define a structure $\mathfrak{A}$ for $\mathcal{L}_1'$ as follows:

$$\mathfrak{A}(\forall) = \mathcal{T}_{\mathcal{L}_1} \text{ where this is the set of all terms for the language } \mathcal{L}_1,$$

$$\mathfrak{A}(\boldsymbol{P}) = P^{\mathfrak{A}} = \{(\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_n}) \in \mathcal{T}_{\mathcal{L}_1}^n : \boldsymbol{P t_1 t_2 \cdots t_n} \in \Delta\}$$

$$\text{where } \boldsymbol{P} \text{ is an } n\text{-place predicate symbol and } \boldsymbol{P} \neq \boldsymbol{E},$$

$$\mathfrak{A}(\boldsymbol{E}) = E^{\mathfrak{A}} = \{(\boldsymbol{t_1}, \boldsymbol{t_2}) \in \mathcal{T}_{\mathcal{L}_1}^2 : \boldsymbol{t_1} \boldsymbol{\approx} \boldsymbol{t_2} \in \Delta\},$$

$$\mathfrak{A}(\boldsymbol{f}) = f^{\mathfrak{A}} \text{ where } f^{\mathfrak{A}}(\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_n}) = \boldsymbol{f t_1 t_2 \cdots t_n}$$

$$\text{for each } n\text{-place function symbol } \boldsymbol{f}, \text{ and}$$

$$c^{\mathfrak{A}} = \boldsymbol{c} \text{ for a constant symbol } \boldsymbol{c}.$$

Now define $s : \mathcal{V}_{\mathcal{L}_1'} \longrightarrow \mathcal{T}_{\mathcal{L}_1}$ by $s(\boldsymbol{v}) = \boldsymbol{v}$. A straightforward induction argument will show that $\overline{s}(\boldsymbol{t}) = \boldsymbol{t}$ for any term $\boldsymbol{t}$ since $\mathcal{T}_{\mathcal{L}_1'} = \mathcal{T}_{\mathcal{L}_1}$. This structure and this function will be our foundation for satisfying the members of Γ. Denote the formula in $\mathcal{L}_1'$ obtained from the formula $\boldsymbol{\varphi}$ in $\mathcal{L}_1$ by replacing all instances of $\boldsymbol{\approx}$ with $\boldsymbol{E}$ in $\boldsymbol{\varphi}^*$. We wish to demonstrate that

$$\models_{\mathfrak{A}} \boldsymbol{\varphi}^*[s] \text{ iff } \boldsymbol{\varphi} \in \Delta.$$

The argument is by induction, but the induction is standard natural number induction on the total number of connectives and quantifier symbols occurring in a formula $\boldsymbol{\varphi}$ in the language $\mathcal{L}_1$.

First, we demonstrate that all formulas having no quantifiers and no connective symbols fulfill the above statement. In $\mathcal{L}_1$, the set of all such formulas is precisely the atomic formulas. For the atomic formula $\boldsymbol{t_1 \approx t_2}$ we have that $(\boldsymbol{t_1 \approx t_2})^* = \boldsymbol{t_1 E t_2}$, and

$$\models_{\mathfrak{A}} \boldsymbol{t_1 E t_2}[s] \text{ iff } (\overline{s}(\boldsymbol{t_1}), \overline{s}(\boldsymbol{t_2})) = (\boldsymbol{t_1}, \boldsymbol{t_2}) \in E^{\mathfrak{A}} \text{ iff}$$

$$\boldsymbol{t_1 \approx t_2} \in \Delta,$$

and the statement holds. If $\boldsymbol{P}$ is an n-place predicate symbol where $\boldsymbol{P} \neq \boldsymbol{E}$, then $(\boldsymbol{P t_1 t_2 \cdots t_n})^* = \boldsymbol{P t_1 t_2 \cdots t_n}$, and

$$\models_{\mathfrak{A}} \boldsymbol{P t_1 t_2 \cdots t_n}[s] \text{ iff } (\overline{s}(\boldsymbol{t_1}), \overline{s}(\boldsymbol{t_2}), \ldots, \overline{s}(\boldsymbol{t_n})) = (\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_n}) \in P^{\mathfrak{A}} \text{ iff}$$

$$\boldsymbol{P t_1 t_2 \cdots t_n} \in \Delta,$$

and again, the statement holds.

For our inductive step, we assume that the statement holds for any $\boldsymbol{\varphi}$ in $\mathcal{L}_1$ having a total of N quantifier and connective symbols or less. A formula with a total number of $N+1$ connective and quantifier symbols will take one of three forms: $(\neg\psi)$, $(\psi \to \chi)$, or $\forall x\psi$ where ψ and χ will have a total number of N or fewer connective and quantifier symbols. The cases of negation and implication are fairly straightforward to show and are left to the reader. We proceed to show that

$$\models_{\mathfrak{A}} \forall x\varphi^*[s] \text{ iff } \forall x\varphi \in \Delta$$

(note that $(\forall x\varphi)^* = \forall x\varphi^*$) where $\forall x\varphi$ has a total number of $N+1$ connective and quantifier symbols.

For the first direction, note that $\theta = ((\neg\forall x\varphi \to (\neg\varphi_{x|c}))$ for a constant symbol c that does not occur in φ. Assume that $\models_{\mathfrak{A}} \forall x\varphi^*[s]$. This statement implies that $\models_{\mathfrak{A}} \varphi^*[s_{x|c}]$ (note that $c = c^{\mathfrak{A}}$). Since c does not occur in φ, c is substitutable for x in φ^* and by the Substitution Lemma,

$$\models_{\mathfrak{A}} \varphi^*[s_{x|c}] \text{ iff } \models_{\mathfrak{A}} (\varphi^*)_{x|c}[s] \text{ iff}$$

$$\models_{\mathfrak{A}} {\varphi_{x|c}}^*[s]$$

since in φ^*, the only symbol modified is the $\approx$ symbol and no variables or constants are modified. Hence, $(\varphi^*)_{x|c} = {\varphi_{x|c}}^*$. Now, φ must have N connective and quantifier symbols occurring in it since $\forall x\varphi$ is assumed to have $N+1$ connective and quantifier symbols occurring in it. Thus, $\varphi_{x|c}$ also has N connective and quantifier symbols occurring in it, and by our inductive hypothesis,

$$\models_{\mathfrak{A}} {\varphi_{x|c}}^*[s] \text{ iff } \varphi_{x|c} \in \Delta.$$

Thus, $(\neg\varphi_{x|c}) \notin \Delta$, by the maximality of Δ. Now if $(\neg\forall x\varphi) \in \Delta$, since $\theta = ((\neg\forall x\varphi \to (\neg\varphi_{x|c})) \in \Delta$, we would have $\Delta \vdash (\neg\varphi_{x|c})$, and since Δ is deductively closed, $(\neg\varphi_{x|c}) \in \Delta$, a contradiction. Thus, $(\neg\forall x\varphi) \notin \Delta$. By the maximality of Δ, $\forall x\varphi \in \Delta$. We have thus shown one direction of the equivalence.

For the other direction, assume $\not\models_{\mathfrak{A}} \forall \boldsymbol{x}\boldsymbol{\varphi}^*[s]$. Then, there is a term $\boldsymbol{t}$ in the language $\mathcal{L}_1$ such that $\not\models_{\mathfrak{A}} \boldsymbol{\varphi}^*[s_{\boldsymbol{x}|\boldsymbol{t}}]$ (note that a generic element in the universe of $\mathfrak{A}$ is a term in $\mathcal{L}_1$). Now, we are not guaranteed that $\boldsymbol{t}$ is substitutable for $\boldsymbol{x}$ in $\boldsymbol{\varphi}^*$, and so we cannot apply the Substitution Lemma as we would like. However, by Theorem 6.3.2, there is an alphabetic variant of $\boldsymbol{\varphi}^*$, say $\boldsymbol{\psi}^*$, in $\mathcal{L}_1'$ such that the term $\boldsymbol{t}$ is substitutable for $\boldsymbol{x}$ in $\boldsymbol{\psi}^*$, and the two formulas are logically equivalent. So, since

$$\not\models_{\mathfrak{A}} \boldsymbol{\varphi}^*[s_{\boldsymbol{x}|\boldsymbol{t}}], \text{ we have that } \not\models_{\mathfrak{A}} \boldsymbol{\psi}^*[s_{\boldsymbol{x}|\boldsymbol{t}}].$$

By the Substitution lemma and since $(\boldsymbol{\varphi}^*)_{\boldsymbol{x}|\boldsymbol{t}} = {\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}}}^*$, we have $\not\models_{\mathfrak{A}} {\boldsymbol{\psi}_{\boldsymbol{x}|\boldsymbol{t}}}^*[s]$. Note that $\boldsymbol{\psi}$ will be an alphabetic variant of $\boldsymbol{\varphi}$ where $\boldsymbol{\varphi} \vdash \boldsymbol{\psi}$ and $\boldsymbol{\psi} \vdash \boldsymbol{\varphi}$, and these formulas will have the same number of quantifier and connective symbols. Thus, since we have assumed that $\forall \boldsymbol{x}\boldsymbol{\varphi}$ has a total of $N + 1$ quantifier and connective symbols, both $\boldsymbol{\varphi}_{\boldsymbol{x}|\boldsymbol{t}}$ and $\boldsymbol{\psi}_{\boldsymbol{x}|\boldsymbol{t}}$ will have a total number of N quantifier and connective symbols. Thus, by our induction hypothesis since

$$\not\models_{\mathfrak{A}} {\boldsymbol{\psi}_{\boldsymbol{x}|\boldsymbol{t}}}^*[s],$$

we have that $\boldsymbol{\psi}_{\boldsymbol{x}|\boldsymbol{t}} \notin \Delta$. Now, $(\forall \boldsymbol{x}\boldsymbol{\psi} \to \boldsymbol{\psi}_{\boldsymbol{x}|\boldsymbol{t}}) \in \Lambda \subseteq \Delta$ so that $\forall \boldsymbol{x}\boldsymbol{\psi} \notin \Delta$, otherwise $\Delta \vdash \boldsymbol{\psi}_{\boldsymbol{x}|\boldsymbol{t}}$ and $\boldsymbol{\psi}_{\boldsymbol{x}|\boldsymbol{t}} \in \Delta$ since Δ is deductively closed. Since $\boldsymbol{\varphi} \vdash \boldsymbol{\psi}$, by the Deduction Theorem, this is equivalent to $\vdash (\boldsymbol{\varphi} \to \boldsymbol{\psi})$. By the Generalization Theorem, this implies that $\vdash \forall \boldsymbol{x}(\boldsymbol{\varphi} \to \boldsymbol{\psi})$, and this implies $\vdash (\forall \boldsymbol{x}\boldsymbol{\varphi} \to \forall \boldsymbol{x}\boldsymbol{\psi})$ by axiom group 3. By the Deduction Theorem, this is equivalent to $\forall \boldsymbol{x}\boldsymbol{\varphi} \vdash \forall \boldsymbol{x}\boldsymbol{\psi}$. Thus, $\forall \boldsymbol{x}\boldsymbol{\varphi} \notin \Delta$, otherwise $\forall \boldsymbol{x}\boldsymbol{\psi} \in \Delta$. We thus have

$$\not\models_{\mathfrak{A}} \forall \boldsymbol{x}\boldsymbol{\varphi}^*[s] \text{ implies } \forall \boldsymbol{x}\boldsymbol{\varphi} \notin \Delta$$

. We have thus proven the equivalence,

$$\models_{\mathfrak{A}} \forall \boldsymbol{x}\boldsymbol{\varphi}^*[s] \text{ iff } \forall \boldsymbol{x}\boldsymbol{\varphi} \in \Delta.$$

This concludes our induction argument, and we can say that for any wff $\boldsymbol{\varphi}$ in $\mathcal{L}_1$,

$$\models_{\mathfrak{A}} \boldsymbol{\varphi}^*[s] \text{ iff } \boldsymbol{\varphi} \in \Delta.$$

Now, if $\mathcal{L}_1$ does not include the symbol $\approx$, then we are done, for then $\boldsymbol{\varphi}^* = \boldsymbol{\varphi}$, and if we restrict ourselves to the language $\mathcal{L}_0$, the structure $\mathfrak{A}$ and the function s satisfy the set Γ. However, suppose that for constant symbols $\boldsymbol{c}$ and $\boldsymbol{d}$ that $\boldsymbol{c} \approx \boldsymbol{d} \in \Gamma$, then our structure $\mathfrak{A}$ does not satisfy this wff since the structure $\mathfrak{A}$ is a function for $\mathcal{L}_1'$ and not $\mathcal{L}_1$. However, our work with $\mathfrak{A}$ has not been for naught. Given our structure $\mathfrak{A}$ for $\mathcal{L}_1'$, we will create what we will call the quotient structure $\mathfrak{A}/\boldsymbol{E}$, to be defined rigorously in what follows.

Step 5: Constructing the Quotient Structure for Satisfaction

Recall that we obtained the language $\mathcal{L}_1'$ from the language $\mathcal{L}_1$ by replacing the symbol $\approx$ with the binary predicate symbol $\boldsymbol{E}$. Given our structure $\mathfrak{A}$, $E^{\mathfrak{A}}$ will be a binary relation on the terms of $\mathcal{L}_1$ (the universe of $\mathfrak{A}$). We demonstrate that $E^{\mathfrak{A}}$ is an equivalence relation on $\mathcal{T}_{\mathcal{L}_1}$. This should seem intuitively clear given that $\boldsymbol{t_1} E^{\mathfrak{A}} \boldsymbol{t_2}$ iff $\boldsymbol{t_1} \approx \boldsymbol{t_2} \in \Delta$. Essentially, Theorem 5.19 demonstrates that $E^{\mathfrak{A}}$ is an equivalence relation. we demonstrate the reflexivity of $E^{\mathfrak{A}}$, leaving symmetry and transitivity to the reader.

By Theorem 5.19(i), $\Delta \vdash \forall \boldsymbol{x}(\boldsymbol{x} \approx \boldsymbol{x})$. We have the logical axiom $(\forall \boldsymbol{x}\boldsymbol{x} \approx \boldsymbol{x} \rightarrow \boldsymbol{t} \approx \boldsymbol{t})$ for any term $\boldsymbol{t}$ since any term $\boldsymbol{t}$ is substitutable for x in $\boldsymbol{x} \approx \boldsymbol{x}$. Hence, $\Delta \vdash \boldsymbol{t} \approx \boldsymbol{t}$ for every term $\boldsymbol{t}$ in $\mathcal{L}_1$. Hence, $\boldsymbol{t} \approx \boldsymbol{t} \in \Delta$. Hence, $\boldsymbol{t} E^{\mathfrak{A}} \boldsymbol{t}$ for every term $\boldsymbol{t}$ in $\mathcal{L}_1$. Thus, $E^{\mathfrak{A}}$ is reflexive. Symmetry and transitivity follow by similar arguments.

We can say a bit more. For each n-place predicate symbol $\boldsymbol{P}$, if $(\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_n}) \in P^{\mathfrak{A}}$ and $\boldsymbol{t_i} E^{\mathfrak{A}} \boldsymbol{t_i'}$ for every $i \leq n$, then $(\boldsymbol{t_1'}, \boldsymbol{t_2'}, \ldots, \boldsymbol{t_n'}) \in P^{\mathfrak{A}}$. By

Theorem 5.20, and by the use of axiom group 2 (and perhaps some alphabetic variants), we have that

$$\begin{aligned}&\Delta \vdash \\ &\boldsymbol{t_1 \approx t'_1 \to t_2 \approx t'_2 \to \cdots \to} \\ &\boldsymbol{t_n \approx t'_n \to Pt_1t_2\cdots t_n \to Pt'_1t'_2\cdots t'_n = \delta}\end{aligned}.$$

Since Δ is deductively closed, we have that $\boldsymbol{\delta} \in \Delta$. By what we showed in step 4, our last statement is equivalent to $\models_{\mathfrak{A}} \boldsymbol{\delta}^*[s]$. By repeated application of Definition 4.22(iv) and the assumptions that $\boldsymbol{t_i}E^{\mathfrak{A}}\boldsymbol{t'_i}$ for every $i \leq n$ and $(\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_n}) \in P^{\mathfrak{A}}$, we have that $\models_{\mathfrak{A}} \boldsymbol{Pt'_1t'_2\cdots t'_n}[s]$. Again by Definition 4.22, we have $(\boldsymbol{t'_1}, \boldsymbol{t'_2}, \ldots, \boldsymbol{t'_n}) \in P^{\mathfrak{A}}$. By similar reasoning and using Theorem 5.21, we also have for each n-place function symbol $\boldsymbol{f}$ and for $\boldsymbol{t_i}E^{\mathfrak{A}}\boldsymbol{t'_i}$ for each $i \leq n$,

$$f^{\mathfrak{A}}(\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_n})E^{\mathfrak{A}}f^{\mathfrak{A}}(\boldsymbol{t'_1}, \boldsymbol{t'_2}, \ldots, \boldsymbol{t'_n}).$$

We are now in position to define the structure that we want. Let $[\boldsymbol{t}]$ designate the equivalence class for $\boldsymbol{t}$ in $\mathcal{T}_{\mathcal{L}_1}$ determined by $E^{\mathfrak{A}}$. We will designate our structure by the symbol $\mathfrak{A}/\boldsymbol{E}$, and the universe of this structure will be $\mathcal{T}_{\mathcal{L}_1}/E^{\mathfrak{A}}$ (the universe of $\mathfrak{A}$ modulo the equivalence relation $E^{\mathfrak{A}}$). We define for each n-place predicate symbol $\boldsymbol{P}$ and each n-place function symbol $\boldsymbol{f}$ the following:

$$([\boldsymbol{t_1}], [\boldsymbol{t_2}], \ldots, [\boldsymbol{t_n}]) \in P^{\mathfrak{A}/\boldsymbol{E}} \text{ iff } (\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_n}) \in P^{\mathfrak{A}} \text{ and}$$

$$f^{\mathfrak{A}/\boldsymbol{E}}([\boldsymbol{t_1}], [\boldsymbol{t_2}], \ldots, [\boldsymbol{t_n}]) = [f^{\mathfrak{A}}(\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_n})].$$

These relations and functions are well defined given our results above. Also for the structure $\mathfrak{A}/\boldsymbol{E}$ we define $c^{\mathfrak{A}/\boldsymbol{E}} = [c^{\mathfrak{A}}]$.

Now let $h : \mathcal{T}_{\mathcal{L}_1} \longrightarrow \mathcal{T}_{\mathcal{L}_1}/E^{\mathfrak{A}}$ be defined by $h(t) = [t]$. By our results above, h will be a homomorphism of $\mathfrak{A}$ onto $\mathfrak{A}/\boldsymbol{E}$. Note that the symbol $\boldsymbol{\approx}$

does not occur in any formula in $\mathcal{L}_1'$. So, by the Homomorphism Theorem,

$$\models_{\mathfrak{A}} \boldsymbol{\varphi}^*[s] \text{ iff } \models_{\mathfrak{A}/\boldsymbol{E}} \boldsymbol{\varphi}^*[h \circ s].$$

Now, $\mathfrak{A}/\boldsymbol{E}$ is a structure for the language $\mathcal{L}_1'$ not $\mathcal{L}_1$. We require a structure for $\mathcal{L}_1$. The structure that we require will be identical to $\mathfrak{A}/\boldsymbol{E}$ except that our new structure will assign $\boldsymbol{\approx}$ to $E^{\mathfrak{A}/\boldsymbol{E}}$ instead of the symbol $\boldsymbol{E}$ being assigned to this relation. This assignment is justified since for $\boldsymbol{t_1}, \boldsymbol{t_2} \in \mathcal{T}_{\mathcal{L}_1}$,

$$[\boldsymbol{t_1}]E^{\mathfrak{A}/\boldsymbol{E}}[\boldsymbol{t_2}] \text{ iff } \boldsymbol{t_1}E^{\mathfrak{A}}\boldsymbol{t_2}$$

$$\text{iff } [\boldsymbol{t_1}] = [\boldsymbol{t_2}].$$

This says that $E^{\mathfrak{A}/\boldsymbol{E}}$ is equality in $\mathcal{T}_{\mathcal{L}_1}/E^{\mathfrak{A}}$. Call our new structure $(\mathfrak{A}/\boldsymbol{E})^\sharp$. We wish to show that

$$\models_{\mathfrak{A}/\boldsymbol{E}} \boldsymbol{\varphi}^*[h \circ s] \text{ iff } \models_{(\mathfrak{A}/\boldsymbol{E})^\sharp} \boldsymbol{\varphi}[h \circ s].$$

It will be sufficient to demonstrate that for $\boldsymbol{t_1}, \boldsymbol{t_2} \in \mathcal{T}_{\mathcal{L}_1}$,

$$\models_{\mathfrak{A}/\boldsymbol{E}} \boldsymbol{t_1}\boldsymbol{E}\boldsymbol{t_2}[h \circ s] \text{ iff } \models_{(\mathfrak{A}/\boldsymbol{E})^\sharp} \boldsymbol{t_1} \boldsymbol{\approx} \boldsymbol{t_2}[h \circ s].$$

Now,

$$\models_{\mathfrak{A}/\boldsymbol{E}} \boldsymbol{t_1}\boldsymbol{E}\boldsymbol{t_2}[h \circ s] \text{ iff } \overline{h \circ s}(\boldsymbol{t_1})E^{\mathfrak{A}/\boldsymbol{E}}\overline{h \circ s}(\boldsymbol{t_2}) \text{ iff}$$

$$\overline{h \circ s}(\boldsymbol{t_1}) = \overline{h \circ s}(\boldsymbol{t_2}) \text{ iff } \models_{(\mathfrak{A}/\boldsymbol{E})^\sharp} \boldsymbol{t_1} \boldsymbol{\approx} \boldsymbol{t_2}[h \circ s].$$

Thus, we have

$$\boldsymbol{\varphi} \in \Delta \text{ iff } \models_{\mathfrak{A}} \boldsymbol{\varphi}^*[s] \text{ iff } \models_{\mathfrak{A}/\boldsymbol{E}} \boldsymbol{\varphi}^*[h \circ s] \text{ iff } \models_{(\mathfrak{A}/\boldsymbol{E})^\sharp} \boldsymbol{\varphi}[h \circ s].$$

Hence, $(\mathfrak{A}/\boldsymbol{E})^\sharp$ is a structure for the language $\mathcal{L}_1$ satisfying every member of Δ with $h \circ s$, and hence also satisfying Γ. If we restrict the structure $(\mathfrak{A}/\boldsymbol{E})^\sharp$ to be a function on $\mathcal{L}_0$, it is clear that this structure will satisfy every member of Γ with $h \circ s$. Hence, Γ being consistent implies that Γ is satisfiable, and the Completeness Theorem is proved. ∎

As a historical note, Gödel, of incompleteness fame, proved a version of the completeness theorem for his doctoral dissertation in 1930, although the proof given here is different from Gödel's proof.

Having proved soundness and completeness, we now prove one more powerful result before discussing what we have proved.

6.3 Compactness

Like with our sentential model for logic, we have a compactness theorem for first-order logic.

Theorem 6.8 (Compactness Theorem for First-Order Logic)

(i) If $\Gamma \models \varphi$, then for some finite set $\Gamma_f \subseteq \Gamma$, we have $\Gamma_f \models \varphi$.

(ii) If every finite subset of Γ is satisfiable, then Γ is satisfiable.

In fact, items (i) and (ii) are equivalent, but we do not prove this here. We instead prove each version.

Proof: For item (i), by completeness we have that $\Gamma \vdash \varphi$. Since deductions are finite, we may find Γ_f, a finite subset of Γ such that $\Gamma_f \vdash \varphi$. By Completeness, we have that $\Gamma_f \models \varphi$.

For item (ii), suppose by way of contradiction that Γ is not satisfiable. By the contrapositive of the Completeness Theorem, Γ cannot be a consistent set, that is, there is a formula φ such that $\Gamma \vdash \varphi$ and $\Gamma \vdash (\neg\varphi)$. Since deductions are finite, we my find finite $\Gamma_f \subseteq \Gamma$ such that $\Gamma_f \vdash \varphi$ and $\Gamma_f \vdash (\neg\varphi)$. By soundness, we must have $\Gamma_f \models \varphi$ and

$\Gamma_f \models (\neg\varphi)$. Since every finite subset of Γ is assumed to be satisfiable, there is a structure $\mathfrak{S}$ and a function $s : \mathcal{V} \longrightarrow \mathbb{U}$ such that $\mathfrak{S}$ satisfies every member of Γ_f with s. Hence we must also have

$$\models_{\mathfrak{S}} \varphi[s] \text{ and } \models_{\mathfrak{S}} (\neg\varphi)[s].$$

However, this is impossible, and we conclude that Γ must in fact be satisfiable.

∎

Corollary 6.8.1 *A set Σ of sentences has a model if and only if every finite subset has a model.*

Proof: It is clear that if Σ has a model, that every finite subset will have a model. If every finite subset has a model, by the Compactness Theorem, then Σ itself will have a model. ∎

Much like the Compactness Theorem for sentential logic, this theorem indicates that if we have a set of formulas that logically implies another formula, we only need finitely many formulas from that set to show this logical implication. Notice in the proof of the Compactness Theorem that the finite property came from the finite nature of formal deductions.

The corollary is quite interesting, for it allows us to show some powerful results about models quite simply.

Corollary 6.8.2 *If a set Σ of sentences has arbitrarily large finite models, then it has an infinite model*

Proof: For each integer $k \geq 2$, there is a sentence $\boldsymbol{\lambda_k}$ that will translate via any structure that there are at least k things in the universe. For instance,

$$\begin{aligned}
\boldsymbol{\lambda_2} &= \exists v_1 \exists v_2 v_1 \not\approx v_2 \\
\boldsymbol{\lambda_3} &= \exists v_1 \exists v_2 \exists v_3 (v_1 \not\approx v_2 \wedge v_2 \not\approx v_3 \wedge v_1 \not\approx v_3)
\end{aligned}$$

Consider now the set $\Sigma \cup \{\boldsymbol{\lambda}_2, \boldsymbol{\lambda}_3, \ldots\}$. Since Σ has arbitrarily large models, every finite subset of $\Sigma \cup \{\boldsymbol{\lambda}_2, \boldsymbol{\lambda}_3, \ldots\}$ will be satisfiable (there will be arbitrarily large universes available to satisfy the $\boldsymbol{\lambda}$'s). So, by compactness, $\Sigma \cup \{\boldsymbol{\lambda}_2, \boldsymbol{\lambda}_3, \ldots\}$ has a model, and the universe of this model must be infinite.
∎

Example 6.9 *Take the set of sentences that formalize the group axioms. This set has arbitrarily large finite models, since for every positive integer* n*,* $\mathbb{Z}_n$ *is a group. Hence there must be an infinite model of the group axioms by this last corollary. This fact we already know,(*$\mathbb{Z}$ *with* $+$ *is an example), but notice that our knowledge coming from the perspective of the corollary, comes not from the demonstration of the existence of such a group directly, but indirectly through the general property of the Compactness Theorem.*

The Compactness Theorem is also a powerful result to show the existence of "non-standard" models of things like arithmetic and analysis.

Our work in this chapter has given us powerful results and insights concerning first-order logical statements. Recall that we are using first-order logic as a model for the deductive thought processes that we reasoners carry out in the real world. Let us consider what indications about real-world meta-level reasoning our results have given us. The Soundness Theorem indicates that if we have a proof of a statement given an assumed set of premises, then we will be guaranteed the truth of the proven statement as long as the premises are indeed true. So, for instance, if we formalize the physical laws of the universe that we believe, given scientific measurement and testing, to be true and from the formalization of these laws we are able to deduce the formal statement of the existence of black holes, then by Soundness we should expect the existence of black holes *even if one has never been observed.* So, perhaps from soundness we should be encouraged to continue scientific observation and exploration. Of course, if we find data that contradicts or seems to contradict what we have reasoned through formally, we do not throw out the soundness theorem since this a logical result of our model of logic. What we would do as scientists would be to adjust our set of *assumed* sentences. In other words, our inherent assumptions are wrong and need to be adjusted.

As indicated with the example of Goldbach's Conjecture, the Completeness Theorem indicates that if we have copious amounts of evidence that in any model of a set of assumed statements, another result also holds, we would expect that there should be a finite formal proof demonstrating the result formally from the set of formal statements that represent our set of assumptions.

Compactness indicates that if a set of statements logically imply a statement, then we only need finitely many statements to demonstrate that implication. Understanding these powerful results paves the way to understanding Gödel's Incompleteness Theorem.

Chapter 7

Models, Theories, and Models of Theories

Having rigorously developed first-order logic, first-order deductions, and the powerful results of soundness, completeness, and compactness, we discuss in more depth first-order models and develop definitions and concepts dealing with models that will be necessary in the discussion of Gödel's Incompleteness Theorem.

7.1 Classes of Models

We begin first with a set of first-order sentences Σ. It is helpful at this point to recall Definition 4.32, and remember that a sentence $\boldsymbol{\sigma}$ is said to be true in a structure $\mathfrak{S}$ for a particular language if $\models_{\mathfrak{S}} \boldsymbol{\sigma}$ (in this case $\models_{\mathfrak{S}} \boldsymbol{\sigma}[s]$ for every $s : \mathcal{V} \to \mathbb{U}$), and we also say that $\mathfrak{S}$ models $\boldsymbol{\sigma}$.

Definition 7.1 *For the set of first-order sentences Σ, Mod Σ is the* **class of all models of** Σ*, that is, the class of all structures for which every member* *of Σ is true.*

As long as there is at least one structure in Mod Σ, this collection will be too large to be a set. The inherent reason for this is that structures are defined to be functions, and the collection of all functions on a particular set is too large to be a set.

Example 7.2 *Let Σ be the set of first-order sentences that are intended to translate the group axioms. Then Mod Σ will be the collection of all groups which is a class.*

Definition 7.3 *A class $\mathfrak{K}$ of structures is* **elementary (*EC*)** *if and only if $\mathfrak{K}$ =Mod $\boldsymbol{\sigma}$ for some sentence $\boldsymbol{\sigma}$. A class $\mathfrak{K}$ is* **elementary in the wider sense ($\boldsymbol{EC_\Delta}$)** *if $\mathfrak{K}$ =Mod Σ where Σ is a set of sentences.*

Of course, any class of structures that is EC must also be EC_Δ.

Example 7.4 *The class of structures for all groups must be EC (an elementary class) since the group axioms could be represented as a single formula that is the conjunction of all of the group axiom formulas.*

Corollary 7.4.1 (to Corollary 6.8.2**)** *The class of all finite structures for a fixed language is not EC_Δ. The class of all infinite structures is not EC. However, the class of all infinite structures is EC_Δ.*

Proof: Suppose there was a set of sentences Σ such that Mod Σ is exactly the set of all finite structures for a fixed language. Since there will be arbitrarily large finite models of the sentence Σ, by Corollary 6.8.2 Σ will also have an infinite model which will be in Mod Σ by definition, a contradiction. So, the first statement holds.

Suppose that there was a sentence $\boldsymbol{\sigma}$ such that the class of all infinite models is Mod $\boldsymbol{\sigma}$. Essentially, $\boldsymbol{\sigma}$ has the necessary form to support the statement "This a structure for the particular language at hand, and the structure is infinite." $(\neg\boldsymbol{\sigma})$ will support the statement "This is not a structure for the particular language at hand or the structure is finite." Since Mod $(\neg\boldsymbol{\sigma})$ is by definition the class of all structures of the language satisfying $(\neg\boldsymbol{\sigma})$, Mod $(\neg\boldsymbol{\sigma})$ must in fact be all finite structures of the language, but as we showed above, an infinite model would have to be in this class, a contradiction. Hence, the set of all infinite structures of a language is not EC.

That the class of all infinite structures is in fact EC_Δ is true because this will be exactly Mod $\{\boldsymbol{\lambda_2}, \boldsymbol{\lambda_3}, \ldots\}$ where the $\boldsymbol{\lambda_i}$'s are as in Corollary 6.8.2, expressing that there are at least i elements. ∎

Example 7.5 *There is no set of sentences* Σ *such that Mod* Σ *is the set of all finite groups. However, the class of all infinite groups will be* Mod $\{\boldsymbol{\sigma}, \boldsymbol{\lambda_2}, \boldsymbol{\lambda_3}, \ldots\}$ *where* $\boldsymbol{\sigma}$ *is the conjunction of all of the group axioms.*

Now that we have a way to refer to all of the models of a set of sentences we are in a position to discuss theories

7.2 Theories

First a definition.

Definition 7.6 *For a fixed language, a* **theory** *is a set of sentences closed under logical implication. That is, a set of sentences in the language,* T*, is a theory if for any sentence* $\boldsymbol{\sigma}$ *in the language* $T \models \boldsymbol{\sigma}$ *implies that* $\boldsymbol{\sigma} \in T$*.*

This definition of theory matches our intuition. As mathematicians, when we talk of the "Theory of Groups," or the "Theory of Rings," or "Number

Theory," we mean the set of all statements (sentences) true of Groups, Rings, or Arithmetic. If we are working in the context of a theory, T, and are able to demonstrate a proof of a statement $\boldsymbol{\tau}$ ($T \vdash \boldsymbol{\tau}$, that is, $\boldsymbol{\tau}$ is a theorem of T), then we know by soundness that $\boldsymbol{\tau}$ is true of our theory ($T \vDash \boldsymbol{\tau}$), and hence we can include $\boldsymbol{\tau}$ as a part of the theory ($\boldsymbol{\tau} \in T$).

Example 7.7 *The set of valid sentences for a fixed language is a theory. Recall that a sentence $\boldsymbol{\sigma}$ will be valid if $\vDash \boldsymbol{\sigma}$. Let V be the set of valid sentences, and suppose that $V \vDash \boldsymbol{\sigma}$ for some sentence $\boldsymbol{\sigma}$. Note that since all of members of V are valid, every structure $\mathfrak{S}$ and $s : \mathcal{V} \longrightarrow \mathbb{U}$ for the language, will satisfy every member of V. Since we are assuming that $V \vDash \boldsymbol{\sigma}$ every structure $\mathfrak{S}$ and function $s : \mathcal{V} \longrightarrow \mathbb{U}$ will also satisfy $\boldsymbol{\sigma}$. Hence, $\boldsymbol{\sigma}$ is also a valid sentence, and $\boldsymbol{\sigma} \in V$.*

The set of all sentences in a fixed language is a theory. Denote the set by $\mathcal{S}$. Since $\boldsymbol{\sigma} \in \mathcal{S}$ is always true. The implication "$\mathcal{S} \vDash \boldsymbol{\sigma}$ implies $\boldsymbol{\sigma} \in \mathcal{S}$" is also always true, so that $\mathcal{S}$ is a theory. However, there are no models for this theory since $\mathcal{S}$ is unsatisfiable since it contains both $\boldsymbol{\sigma}$ and $(\neg\boldsymbol{\sigma})$.

Now, take a class of structures $\mathfrak{K}$ for a fixed language. Define

$$\text{Th } \mathfrak{K} = \{\boldsymbol{\sigma} \in \mathcal{S} : \boldsymbol{\sigma} \text{ is true in every member of } \mathfrak{K}\}$$

Theorem 7.8 *Th $\mathfrak{K}$ is a theory.*

Proof: Suppose that Th $\mathfrak{K} \vDash \boldsymbol{\sigma}$ for some sentence $\boldsymbol{\sigma}$ of the language. Let $\mathfrak{S}$ be a structure in $\mathfrak{K}$. Then since $\mathfrak{S}$ models Th $\mathfrak{K}$ by definition, we must also have $\vDash_{\mathfrak{S}} \boldsymbol{\sigma}$ since Th $\mathfrak{K} \vDash \boldsymbol{\sigma}$. But then, $\boldsymbol{\sigma}$ is true of $\mathfrak{S}$, and since our choice for $\mathfrak{S}$ in $\mathfrak{K}$ was arbitrary, $\boldsymbol{\sigma}$ will be true for every structure in the class $\mathfrak{K}$. Hence $\boldsymbol{\sigma} \in$ Th $\mathfrak{K}$ by definition, and Th $\mathfrak{K}$ is indeed a theory. ∎

This theorem justifies us referring to the set Th $\mathfrak{K}$ as the *theory of the class* $\mathfrak{K}$.

Example 7.9 *Let $\mathfrak{G}$ be the class of all groups, $\mathfrak{R}$ the class of all rings, and $\mathfrak{W}$ the class of all sets. Then Th $\mathfrak{G}$, Th $\mathfrak{R}$, and Th $\mathfrak{W}$ are the theories of groups, rings, and sets respectively. Note that our language might need to change. For instance for the theory of groups, we could use the parameters $\forall$, $\boldsymbol{e}$, $*$, whereas for the theory of rings we could use a language with parameters $\forall$, $\mathbf{0}$, $\mathbf{1}$, $+$, and $\cdot$.*

Now that we have a way to refer to formal theories, we define another set.

Definition 7.10 *The set of* **consequences** *of Σ is the following set of sentences:*

$$Cn\ \Sigma = \{\boldsymbol{\sigma} : \Sigma \vDash \boldsymbol{\sigma}\}.$$

Of course the set of consequences is the set of all formulas that Σ logically implies. Also, we have $\Sigma \subseteq \mathrm{Cn}\ \Sigma$. We can say even more.

Theorem 7.11 *$Cn\ \Sigma = Th\ Mod\ \Sigma$*

Proof: Mod Σ is the class of all models of Σ. So, Th Mod Σ is exactly the set of sentences true about all models of the set of sentences Σ. Suppose $\boldsymbol{\sigma} \in \mathrm{Cn}\ \Sigma$, then $\Sigma \vDash \boldsymbol{\sigma}$. So every model of Σ must also model $\boldsymbol{\sigma}$. Hence, $\boldsymbol{\sigma} \in \mathrm{Th\ Mod}\ \Sigma$.

Now suppose that $\boldsymbol{\sigma} \in \mathrm{Th\ Mod}\ \Sigma$. Then $\boldsymbol{\sigma}$ is true of every model of Σ, that is, if a structure models Σ, it will also model $\boldsymbol{\sigma}$ as well. Hence, $\Sigma \vDash \boldsymbol{\sigma}$, and $\boldsymbol{\sigma} \in \mathrm{Cn}\ \Sigma$. ∎

What this theorem says is that the set of consequences of a set of sentences is exactly the set of sentences true in every model of Σ. This theorem in fact justifies the axiomatic treatment of mathematical theories. Intuitively, the axioms (Σ) of a particular mathematical theory are a set of sentences assumed to be true without proof. Mathematical work within that theory then entails finding the consequences of the assumed set of axioms. This work entails demonstrating proofs of new theorems ($\Sigma \vdash \boldsymbol{\tau}$) which by the soundness and compactness theorems will be equivalent to demonstrating a new truth of that theory ($\Sigma \models \boldsymbol{\tau}$). The collection of all such possible truths that are demonstrable (Cn Σ) is exactly the entire theory that we are working with (Th Mod Σ). Of course, we want our set of axioms for our theory to have an important property.

Definition 7.12 *A theory T is* **axiomatizable** *if there is a decidable (in the sense of Section 3.3) set of sentences Σ such that $T = Cn\ \Sigma$. A theory is* **finitely axiomatizable** *if there is a finite set of sentences Σ such that $T = Cn\ \Sigma$ (equivalently if there is a single sentence $\boldsymbol{\sigma}$ such that $T = Cn\ \{\boldsymbol{\sigma}\}$ since we could take $\boldsymbol{\sigma}$ to be the conjunction of all the members of Σ).*

Theorem 7.13 *If the theory $Cn\ \Sigma$ is finitely axiomatizable, then there is a finite $\Sigma_f \subseteq \Sigma$ such that $Cn\ \Sigma_f = Cn\ \Sigma$.*

Proof: By definition, there is a sentence $\boldsymbol{\sigma}$ such that Cn Σ = Cn $\{\boldsymbol{\sigma}\}$. Now, $\boldsymbol{\sigma}$ may not be in Σ, but of course $\Sigma \models \boldsymbol{\sigma}$. By the Compactness Theorem, there is finite $\Sigma_f \subseteq \Sigma$ such that $\Sigma_f \models \boldsymbol{\sigma}$. Thus, we have

$$\text{Cn } \{\boldsymbol{\sigma}\} \subseteq \text{Cn } \Sigma_f \subseteq \text{Cn } \Sigma = \text{Cn } \{\boldsymbol{\sigma}\}$$

so that "$\subseteq$" may be replaced with "$=$". ∎

Example 7.14 *Group theory is a finitely axiomatizable theory. Let* $\mathcal{G}$ *be the set of formal sentences for the group axioms (see Example* 4.5*). Then* $Th\ \mathfrak{G} = Cn\ \mathcal{G}$*. The theory of infinite groups is axiomatizable but not finitely axiomatizable. Let* $\mathfrak{G}^\infty$ *be the subclass of* $\mathfrak{G}$ *containing only the infinite models of* $\mathfrak{G}$*. Then* $Th\ \mathfrak{G}^\infty = Cn\ \mathcal{G} \cup \{\boldsymbol{\lambda_2}, \boldsymbol{\lambda_3}, \ldots\}$ *where the* $\boldsymbol{\lambda_i}$*'s are as in Corollary* 6.8.2*.*

Number theory will of course be axiomatizable since it is built on the Peano axioms which can be expressed formally as first-order logical statements. We will discuss this theory in more depth in the next chapter as a set up for Gödel's Incompleteness Theorem.

We make one more definition before leaving this section.

Definition 7.15 *A* **subtheory** *of a theory* T *is a set of sentences* S *such that* $S \subseteq T$ *and* S *is theory in its own right.*

Theorem 7.16 *Let* $\mathfrak{K}_1$ *and* $\mathfrak{K}_2$ *be two classes of structures for the same fixed language such that* $\mathfrak{K}_1$ *is a subclass of* $\mathfrak{K}_2$*. Then* $Th\ \mathfrak{K}_2$ *is a subtheory of* $Th\ \mathfrak{K}_1$*.*

Proof: Th $\mathfrak{K}_2$ is already a theory. Let $\boldsymbol{\sigma} \in$ Th $\mathfrak{K}_2$. By the definition of Th $\mathfrak{K}_2$, every member of $\mathfrak{K}_2$ models $\boldsymbol{\sigma}$. Since $\mathfrak{K}_1$ is a subclass of $\mathfrak{K}_2$, we must have that every member of $\mathfrak{K}_1$ models $\boldsymbol{\sigma}$. Hence $\boldsymbol{\sigma} \in$ Th $\mathfrak{K}_1$ by definition, and we must have that Th $\mathfrak{K}_2 \subseteq$ Th $\mathfrak{K}_1$. ∎

Theorem 7.17 *Let* Σ_1 *and* Σ_2 *be two sets of sentences in the fixed language where* $\Sigma_1 \subseteq \Sigma_2$*. Then* $Cn\ \Sigma_1$ *is a subtheory of* $Cn\ \Sigma_2$*.*

Proof: Cn Σ_1 and Cn Σ_2 are both theories by Theorem 7.11. Let $\boldsymbol{\sigma} \in$ Cn Σ_1. Then, by definition, $\Sigma_1 \models \boldsymbol{\sigma}$. Clearly then, $\Sigma_2 \vdash \boldsymbol{\sigma}$ since $\Sigma_1 \subseteq \Sigma_2$. Hence, $\boldsymbol{\sigma} \in$ Cn Σ_2, and we have that Cn $\Sigma_1 \subseteq$ Cn Σ_2. ∎

Example 7.18 *From the last example, we see that by the two theorems we have just stated, and proved,* $Th\ \mathfrak{G} = Cn\ \mathcal{G} \subseteq Th\ \mathfrak{G}^{\infty} = Cn\ \mathcal{G} \cup \{\boldsymbol{\lambda_2}, \boldsymbol{\lambda_3}, \ldots\}$. *Thus, the theory of groups is a subtheory of the theory of infinite groups. The idea is that the theory of infinite groups has a richer structure than the theory of groups.*

As above, let $\mathfrak{F}$ *be the class of all fields. Note that the language used to support field theory will be rich enough to support ring theory and group theory. So, we can think of* $\mathfrak{G}$ *as a subclass of* $\mathfrak{R}$ *and of* $\mathfrak{R}$ *as a subclass of* $\mathfrak{F}$. *By Theorem* 7.16*, we have that* $Th\ \mathfrak{F} \subseteq Th\ \mathfrak{R} \subseteq Th\ \mathfrak{G}$. *Thus, the theory of fields is a subtheory of both the theory of rings and of groups, and the theory of rings is a subtheory of the theory of groups. These facts make sense since we can think of a ring as a specific type of group and of a field as a specific type of ring.*

Before leaving this chapter, we discuss a very important concept for Gödel's Incompleteness Theorem.

7.3 Completeness

To discuss incompleteness, we must know what we mean by completeness.

Definition 7.19 *A* **complete theory** $\boldsymbol{T}$ *is a theory such that for every sentence* $\boldsymbol{\sigma}$ *either* $\boldsymbol{\sigma} \in T$ *or* $(\neg\boldsymbol{\sigma}) \in T$.

Intuitively, what this definition says is that when we have a complete theory, we will be able to decide exactly what statements belong to T and which do not. Now it is important at this point to notice that the completeness defined here is not the same as the completeness of the Completeness Theorem. Recall that the Completeness Theorem (Version 1) stated that given a logical implication of a wff (which may not be a sentence), there is a formal deduction (proof) of that wff. However, the definition of *complete theory* discusses, not the existence of deductions inherently but whether a a certain sentence or its negation are in that theory.

Suppose we have a complete theory T, and take an arbitrary sentence $\boldsymbol{\sigma}$. By definition, of complete theory, we have that either $\boldsymbol{\sigma} \in T$ or $(\neg\boldsymbol{\sigma}) \in T$. This implies that either $T \vDash \boldsymbol{\sigma}$ or $T \vDash (\neg\boldsymbol{\sigma})$. Since $\nvDash_{\mathfrak{S}} \boldsymbol{\sigma}$ if and only if $\vDash_{\mathfrak{S}} (\neg\boldsymbol{\sigma})$, our last statement is equivalent to "either $T \vDash \boldsymbol{\sigma}$ or $T \nvDash \boldsymbol{\sigma}$. By the Soundness and Completeness Theorems, we have that $T \vdash \boldsymbol{\sigma}$ or $T \nvdash \boldsymbol{\sigma}$.

Now we attempt a converse argument to see where the definition of complete theory will not, in general, be equivalent to the Soundness and Completeness Theorems. In other words, we use an attempted converse argument to demonstrate that the definition of complete theory defines a new concept different from those involved in the Soundness and Completeness Theorems.

Suppose T is a theory (not necessarily complete) and $\boldsymbol{\sigma}$ is an arbitrary sentence in the first-order language in which we are dealing. If we can demonstrate that T is a complete theory, then the completeness of the Completeness Theorem and the completeness of having a complete theory are equivalent concepts. "Either $T \vdash \boldsymbol{\sigma}$ or $T \nvdash \boldsymbol{\sigma}$" is a tautology by Soundness and Completeness this implies that either $T \vDash \boldsymbol{\sigma}$ or $T \nvDash \boldsymbol{\sigma}$. Since T is a theory and hence closed under logical implication (i.e. $T \vDash \boldsymbol{\sigma}$ if and only if $\boldsymbol{\sigma} \in T$), our last statement is equivalent to either $\boldsymbol{\sigma} \in T$ or $\boldsymbol{\sigma} \notin T$. However, $\boldsymbol{\sigma} \notin T$ does not necessarily imply that $(\neg\boldsymbol{\sigma}) \in T$. To see the breaking point, back up a stage to the case when $T \nvDash \boldsymbol{\sigma}$. By definition of logical implication, there is a structure $\mathfrak{S}$ that models the set of sentences T but such that $\not\models_{\mathfrak{S}} \boldsymbol{\sigma}$. Now, $\not\models_{\mathfrak{S}} \boldsymbol{\sigma}$ is equivalent to $\models_{\mathfrak{S}} (\neg\boldsymbol{\sigma})$, but we are only guaranteed that $(\neg\boldsymbol{\sigma})$ is modelled by the structure $\mathfrak{S}$. To say that $T \models (\neg\boldsymbol{\sigma})$ (and hence $(\neg\boldsymbol{\sigma}) \in T$) *every* model of T (not just $\mathfrak{S}$) would have to model $(\neg\boldsymbol{\sigma})$). However, it may be the case that there is a model of T say, $\mathfrak{A}$ such that $\not\models_{\mathfrak{A}} (\neg\boldsymbol{\sigma})$. This fact would not contradict that information that we are given.

For instance, in the language that we have been using to support group theory, the sentence

$$\boldsymbol{\delta} = \forall \boldsymbol{x} \forall \boldsymbol{y}\ \boldsymbol{x} * \boldsymbol{y} \approx \boldsymbol{y} * \boldsymbol{x}$$

formalizes the statement that a group is commutative (abelian). Of course we know that $\text{Th}\ \mathfrak{G} \not\models \boldsymbol{\delta}$ since D_3, the group of orientation preserving rotations and flips of an equilateral triangle is a non-abelian group (rotating by $60°$ and then flipping will not be the same as flipping and then rotating by $60°$). However, we also know that $\text{Th}\ \mathfrak{G} \not\models (\neg\boldsymbol{\delta})$ since $\mathbb{Z}_2 \times \mathbb{Z}_2$ is a commutative group. So, $\boldsymbol{\delta} \notin \text{Th}\ \mathfrak{G}$ and $(\neg\boldsymbol{\delta}) \notin \text{Th}\ \mathfrak{G}$, and the the theory of groups is incomplete. We can see that this idea of a complete theory is a different concept than the Completeness Theorem.

There are also powerful results that are sufficient to show when a model is complete, but we do not discuss these in this thesis. The interested reader is encouraged to consult *A Mathematical Introduction to Logic* by Enderton for a more complete discussion of this topic. For an example of a complete theory, we state the following theorem without proof. Recall that an algebraically closed field is one that contains the roots of any polynomial with coefficients from the field (e.g. $\mathbb{C}$ is an algebraically closed field by the Fundamental Theorem of Algebra). A ring (of which a field is a special case) is of characteristic 0 if there is no positive integer n for which $n \cdot a = 0$ for every element a in the ring

([2] page 218).

Theorem 7.20 *The theory of algebraically closed fields of characteristic 0 is complete.*

The language for this theory will be the language used to support fields and will have the parameters $\forall$, $\boldsymbol{\approx}$, $\boldsymbol{M}$ (a unary predicate symbol whose intended translation will be "a member of the field"), $\mathbf{0}$, $\mathbf{1}$, $\boldsymbol{+}$ (a binary function symbol), and $\boldsymbol{\cdot}$ (another binary function symbol). Let $\overline{\mathfrak{F}_0}$ be the class of algebraically closed fields of characteristic 0. Then for any sentence $\boldsymbol{\sigma}$ in the language for fields, either $\boldsymbol{\sigma} \in \text{Th}\ \overline{\mathfrak{F}_0}$ or $(\neg\boldsymbol{\sigma}) \in \text{Th}\ \overline{\mathfrak{F}_0}$. For instance

$$\boldsymbol{\delta}_n = \forall \boldsymbol{x}(\boldsymbol{x} \cdot \boldsymbol{x} \cdot \boldsymbol{x} \cdot \cdots \cdot \boldsymbol{x} \approx \mathbf{1} \rightarrow \boldsymbol{M}\boldsymbol{x})$$

where n is a positive integer and there are n copies of the variable symbol $\boldsymbol{x}$ involved in $\boldsymbol{x} \cdot \boldsymbol{x} \cdot \boldsymbol{x} \cdot \cdots \cdot \boldsymbol{x}$, is a sentence in the language such that $\boldsymbol{\delta_n} \in \text{Th}\ \overline{\mathfrak{F}_0}$ or $(\neg\boldsymbol{\delta_n}) \in \text{Th}\ \overline{\mathfrak{F}_0}$. Essentially, this sentence claims that the nth roots of unity are in the field which will be true in every structure in $\overline{\mathfrak{F}_0}$, and hence $\boldsymbol{\delta}_n \in \text{Th}\ \overline{\mathfrak{F}_0}$ for every positive integer n.

Completeness says that we can decide exactly which statements belong to our theory and which do not. In some sense, it indicates that we have reached the full range of expressibility with our given language. This claim is supported by the following theorem.

Theorem 7.21 *Let T_1 and T_2 be theories such that*

(i) $T_1 \subseteq T_2$

(ii) T_1 *is complete*

(iii) T_2 *is satisfiable.*

Then $T_1 = T_2$.

Proof: We show this fact by contradiction. Suppose $T_1 \subset T_2$. Then, there is a sentence $\boldsymbol{\sigma} \in T_2$ such that $\boldsymbol{\sigma} \notin T_1$. By the completeness of the theory T_1, $(\neg\boldsymbol{\sigma}) \in T_1$. Since $T_1 \subset T_2$, $(\neg\boldsymbol{\sigma}) \in T_2$. So, both $\boldsymbol{\sigma}$ and $(\neg\boldsymbol{\sigma})$ are in T_2. It is clear then that T_2 cannot be satisfied by any structure for the language since $\models_{\mathfrak{S}} (\neg\boldsymbol{\sigma})$ if and only if $\not\models_{\mathfrak{S}} \boldsymbol{\sigma}$ for every structure $\mathfrak{S}$. This contradicts our assumption that T_2 was satisfiable. So, in fact $T_1 = T_2$. ∎

Example 7.22 *Consider the chain*

$$Th\ \mathfrak{G} \subset Th\ \mathfrak{R} \subset Th\ \mathfrak{F} \subset Th\ \mathfrak{F}_0 \subset Th\ \overline{\mathfrak{F}_0}.$$

Where $\mathfrak{F}_0$ *is the class of all fields of characteristic 0. We are justified in stopping the chain with* $Th\ \overline{\mathfrak{F}_0}$ *by Theorem* 7.20 *and by the last theorem. With the language that we have at our disposal, and for this chain, we cannot obtain a richer satisfiable theory than the theory of algebraically closed fields of characteristic 0.*

Now that we have the terminology and concepts in this chapter under our belt, we are finally ready to discuss the main goal of this thesis.

Chapter 8

Gödel's Incompleteness Theorem

As the main goal of this thesis, the last several chapters have gathered many of the main concepts and tools we will need to prove a version of Gödel's famous theorem.

8.1 Overview of the Theorem and Its Proof

In this chapter we will be focusing our attention on one particular structure for one particular fixed language. The language that we will use will be the language used thus far to support the structure of number theory, and it is this structure to which we will direct our focus.

Recall that the language of number theory has the logical equality symbol $\approx$ and has one, 2-place predicate symbol $<$. Furthermore, the language has one 1-place function symbol $\boldsymbol{S}$, and three 2-place function symbols, $+$, $\cdot$, and $\boldsymbol{E}$. Finally, it has one constant symbol, $\mathbf{0}$.

Now recall (see Example 4.20) we have defined a structure $\mathfrak{N}$ by $\mathfrak{N}(\forall) = \mathbb{N}$, $\mathfrak{N}(\boldsymbol{<}) = < \subseteq \mathbb{N}^2$, $\mathfrak{N}(\mathbf{0}) = 0$, $\mathfrak{N}(\boldsymbol{S}) = S$ where $S(x) = x + 1$, $\mathfrak{N}(\boldsymbol{+}) = +$ (addition), $\mathfrak{N}(\boldsymbol{\cdot}) = \cdot$ (multiplication), and $\mathfrak{N}(\boldsymbol{E}) = E$ (exponentiation). This is of course the actual structure of number theory. Using the terminology of the last chapter, we are studying the theory Th $\mathfrak{N}$ where we abuse notation and say that $\mathfrak{N}$ is the same as $\{\mathfrak{N}\}$, the set/class with the single structure $\mathfrak{N}$. This will indeed be a theory by Theorem 7.8, and *is* number theory.

In fact, we will not study this full theory but a subtheory of this theory. Before describing this subtheory, we specify some abbreviations that we will use with the formal language. First, $\boldsymbol{S}^n\mathbf{0}$ will be shorthand for $\boldsymbol{SSSS}\cdots\boldsymbol{S}\mathbf{0}$ where the function symbol $\boldsymbol{S}$ appears n (a positive integer) number of times. Let $s : \mathcal{V} \longrightarrow \mathbb{N}$. Then $\overline{s}(\boldsymbol{S}^n\mathbf{0}) = n$ since $\boldsymbol{S}^{\mathfrak{N}} = S$, the successor function, and $\mathbf{0}^{\mathfrak{N}} = 0$. Second $\boldsymbol{x} \leq \boldsymbol{y}$ will be shorthand for $\boldsymbol{x} < \boldsymbol{y} \vee \boldsymbol{x} \approx \boldsymbol{y}$. Third, notice that we will write function and predicate symbols in a more natural way (e.g. $\boldsymbol{x} < \boldsymbol{y}$ instead of $< \boldsymbol{xy}$), and we are including the logical connectives $\exists$, $\vee$, $\leftrightarrow$, and $\wedge$, even though, officially, the formal language does not contain these symbols.

We will now create a finitely axiomatizable subtheory of Th $\mathfrak{N}$. We denote the set of our axioms by A, and the sentences (axioms) to be included in A are specified in the list below

S1	$\forall \boldsymbol{x}\ \boldsymbol{Sx} \not\approx \mathbf{0}$
S2	$\forall \boldsymbol{x} \forall \boldsymbol{y} (\boldsymbol{Sx} \approx \boldsymbol{Sy} \rightarrow \boldsymbol{x} \approx \boldsymbol{y})$
L1	$\forall \boldsymbol{x} \forall \boldsymbol{y} (\boldsymbol{x} < \boldsymbol{Sy} \leftrightarrow \boldsymbol{x} \leq \boldsymbol{y})$
L2	$\forall \boldsymbol{x}\ \boldsymbol{x} \not< \mathbf{0}$
L3	$\forall \boldsymbol{x} \forall \boldsymbol{y} (\boldsymbol{x} < \boldsymbol{y} \vee \boldsymbol{x} \approx \boldsymbol{y} \vee \boldsymbol{y} < \boldsymbol{x})$
A1	$\forall \boldsymbol{x}\ \boldsymbol{x} + \mathbf{0} \approx \boldsymbol{x}$
A2	$\forall \boldsymbol{x} \forall \boldsymbol{y}\ \boldsymbol{x} + \boldsymbol{Sy} \approx \boldsymbol{S}(\boldsymbol{x} + \boldsymbol{y})$
M1	$\forall \boldsymbol{x}\ \boldsymbol{x} \cdot \mathbf{0} \approx \mathbf{0}$
M2	$\forall \boldsymbol{x} \forall \boldsymbol{y}\ \boldsymbol{x} \cdot \boldsymbol{Sy} \approx \boldsymbol{x} \cdot \boldsymbol{y} + \boldsymbol{x}$
E1	$\forall \boldsymbol{x}\ \boldsymbol{xE}\mathbf{0} \approx \boldsymbol{S}\mathbf{0}$
E2	$\forall \boldsymbol{x} \forall \boldsymbol{y}\ \boldsymbol{xESy} \approx \boldsymbol{xEy} \cdot \boldsymbol{x}$

Now, Cn A is a theory by Theorem 7.11 and is clearly finitely axiomatizable. Since a quick examination of the axioms will show that $\mathfrak{N}$ models all of these 11 sentences, Cn $A \subseteq$ Th $\mathfrak{N}$.

We are almost in position to outline both our main result and our approach to it. We need to sketch one concept that we will discuss in more depth later on. It is possible to associate with each formula in the formal language a natural number. Furthermore it is possible to extend this assignment to deductions using formulas in such a way that the structure of a deduction will be reflected in the relationships between the natural numbers assigned to the formal formulas involved. In this way, "statements" made by formulas can be translated into statements about natural numbers. This type of numbering is called Gödel numbering, and given its existence, for any formula $\boldsymbol{\varphi}$ we have the natural number assigned to $\boldsymbol{\varphi}$, $\#\boldsymbol{\varphi}$. For any set of formulas S then, we have an associated set of natural numbers $\#S = \{\#\boldsymbol{\varphi} : \boldsymbol{\varphi} \in S\}$.

Now we can state a form of Gödel's Theorem, but before doing so, it will be helpful to the reader to briefly review Section 4.4 on the definability of relations within structures.

Theorem 8.1 (Gödel's Incompleteness Theorem) *Let $S \subseteq Th\, \mathfrak{N}$ be a set of sentences true in $\mathfrak{N}$, and assume that the set $\#S$ of Gödel numbers of the members of S is a set definable in $\mathfrak{N}$. Then we can find a sentence $\boldsymbol{\sigma}$ such that $\boldsymbol{\sigma}$ is true in $\mathfrak{N}$ but such that $\boldsymbol{\sigma}$ is not deducible from S.*

For Gödel's theorem, we start with a set of true sentences, translate those numbers into natural numbers via Gödel numbering. But after using Gödel numbering, we are in the universe $\mathbb{N}$ determined by the structure $\mathfrak{N}$. We can see if $\#S$ is representable in the structure $\mathfrak{N}$. If this is the case, then essentially all the statements that S expresses can be translated into relationships between natural numbers (this is the definability piece). If this is so, then Gödel's theorem states that we can find a sentence that true in Th $\mathfrak{N}$ but not deducible from S. In particular, if we are dealing with a subtheory S of Th $\mathfrak{N}$ in which every statement we make *about* natural numbers can be represented as a relationship *between* natural numbers, we will always be able to find such a sentence, true in Th $\mathfrak{N}$, but not deducible in the subtheory. Note then, that any such subtheory S will be *incomplete* since for this particular sentence $\boldsymbol{\sigma}$ that the theorem speaks of, $\boldsymbol{\sigma} \notin S$, otherwise $S \vDash \boldsymbol{\sigma}$, and by the Completeness Theorem $S \vdash \boldsymbol{\sigma}$. However, neither can we have $(\neg\boldsymbol{\sigma}) \in S$ for then $(\neg\boldsymbol{\sigma}) \in$ Th $\mathfrak{N}$ and $\boldsymbol{\sigma} \in$ Th $\mathfrak{N}$ ($\boldsymbol{\sigma}$ is true in Th $\mathfrak{N}$). These facts would imply that Th $\mathfrak{N}$ is unsatisfiable, but $\mathfrak{N}$ satisfies this set of sentences by definition. So, any such subtheory S must be incomplete, and this fact is the incompleteness of which the theorem speaks.

Note also that for Th $\mathfrak{N} \subseteq$ Th $\mathfrak{N}$, there must be some sentences in the theory that cannot be represented as statements between natural numbers. That is, there are statements *about* natural numbers that cannot be re-represented as statements concerning a relationship *between* natural numbers. Otherwise, the theorem would state contradictory assertions. Perhaps a more concise way to say what we are getting at is that more can be said *about* the natural numbers than can be said *with* the natural numbers.

Also notice that Th $\mathfrak{N}$ is a complete theory. Take any sentence in the language $\boldsymbol{\sigma}$. Either $\models_{\mathfrak{N}} \boldsymbol{\sigma}$ or $\not\models_{\mathfrak{N}} (\neg\boldsymbol{\sigma})$. In the former case $\boldsymbol{\sigma} \in$ Th $\mathfrak{N}$ by definition. By definition, the latter case is equivalent to $\models_{\mathfrak{N}} (\neg\boldsymbol{\sigma})$. In this case, again by definition, we have that $(\neg\boldsymbol{\sigma}) \in$ Th $\mathfrak{N}$. So, Th $\mathfrak{N}$ is complete since for any sentence $\boldsymbol{\sigma}$, either $\boldsymbol{\sigma}$ or $(\neg\boldsymbol{\sigma})$ are in Th $\mathfrak{N}$.

What exactly will the sentence spoken of in the theorem be? While we cannot give the full technical details at this moment, we will be able to characterize this sentence. Given the set of sentences S described in the theorem, our sentence $\boldsymbol{\sigma}$ will essentially say

"This sentence is not deducible from S".

The proof that the sentence $\boldsymbol{\sigma}$ is true in Th $\mathfrak{N}$ but not deducible from S will then essentially be by contradiction. We give an informal sketch of it here. Suppose that the above sentence is false. Then $\boldsymbol{\sigma}$ is in fact deducible from S. Then by the Soundness Theorem, $S \models \boldsymbol{\sigma}$, and since $\mathfrak{N}$ is a model of S ($S \subseteq$ Th $\mathfrak{N}$), $\boldsymbol{\sigma}$ is true in $\mathfrak{N}$. These next steps are where we will need to fill in many formal details. Since $\boldsymbol{\sigma}$ is true in $\mathfrak{N}$, then in fact $S \nvdash \boldsymbol{\sigma}$ since this is what the sentence $\boldsymbol{\sigma}$ asserts. This fact contradicts our original assumption that $\boldsymbol{\sigma}$ is false. So, in fact, $\boldsymbol{\sigma}$ must be true, that is, true in $\mathfrak{N}$ ($\boldsymbol{\sigma} \in$ Th $\mathfrak{N}$).

Since it is true in $\mathfrak{N}$, what it asserts is the case, that is $S \nvdash \boldsymbol{\sigma}$.

The reader may object and claim that the sentence "This sentence is not deducible from S," is a meaningless sentence much like the sentence "This statement is false." The last sentence is nonsensical since if it is true, then it is false, and if it is false then it is true. Later in the discussion, we will show that we can formally and coherently construct such a sentence $\boldsymbol{\sigma}$, but even now, we can informally see that our sentence is not inherently self-contradictory. It almost is, but not quite.

Consider our discussion of completeness of theories in the last chapter. We can say that Th $\mathfrak{G}$, the theory of groups, was an incomplete theory since the formal sentence that represents the abelian property, $\boldsymbol{\delta} = \forall \boldsymbol{x} \forall \boldsymbol{y}\ \boldsymbol{x} * \boldsymbol{y} \approx \boldsymbol{y} * \boldsymbol{x}$, was true in some groups but not in others. To draw the analogy with the current case, note that Th $\mathfrak{G} \subset$ Th $\overline{\mathfrak{F}_0}$. The theory of algebraically closed fields of characteristic 0 is a complete theory as noted in the last chapter, a theory of which the theory of groups is a subtheory. Also, $\boldsymbol{\delta}$ is true in every member of $\overline{\mathfrak{F}_0}$. So, $\boldsymbol{\delta}$ is a sentence true in every member of $\overline{\mathfrak{F}_0}$ but not deducible from Th $\mathfrak{G}$ much as in our present case. The sentence $\boldsymbol{\delta}$ might just have easily said "This sentence is not deducible from Th $\mathfrak{G}$," and the results we have just described would be the same. So, our sentence $\boldsymbol{\sigma}$ is not inherently self-contradictory. It merely asserts that it as a sentence is true in a larger theory but not deducible from a smaller set of sentences.

The reader may then ask, "If this argument by contradiction is valid, why does it not work with S replaced with Th $\mathfrak{N}$?" If this were possible, then Gödel's Theorem would be nonsense since if a sentence $\boldsymbol{\sigma}$ were true in $\mathfrak{N}$ ($\boldsymbol{\sigma} \in$ Th $\mathfrak{N}$) it must also be deducible from Th $\mathfrak{N}$ by the Completeness

Theorem (a version of which Gödel proved!). This time our sentence is

"This sentence is not deducible from Th $\mathfrak{N}$".

Remember that this is only an informal argument. Formally, this will be a sentence in the first-order language. This sentence is now supposed to be our sentence $\boldsymbol{\sigma}$. Now, the sentence is self-contradictory since Th $\mathfrak{N}$ is a complete theory. Either $\boldsymbol{\sigma} \in$ Th $\mathfrak{N}$ ($\boldsymbol{\sigma}$ is true in $\mathfrak{N}$) or $(\neg\boldsymbol{\sigma}) \in$ Th $\mathfrak{N}$ ($\boldsymbol{\sigma}$ is false in $\mathfrak{N}$). If $\boldsymbol{\sigma} \in$ Th $\mathfrak{N}$, then of course Th $\mathfrak{N} \vdash \boldsymbol{\sigma}$ since any theory is closed under logical implication and by the Completeness Theorem. But this contradicts what $\boldsymbol{\sigma}$ asserts, so we must have $(\neg\boldsymbol{\sigma}) \in$ Th $\mathfrak{N}$. Thus, $(\neg\boldsymbol{\sigma})$ is true in $\mathfrak{N}$, or equivalently $\boldsymbol{\sigma}$ is false in $\mathfrak{N}$. But if it is false, $\boldsymbol{\sigma}$ must in fact be deducible from Th $\mathfrak{N}$, hence also true in Th $\mathfrak{N}$ by the Soundness Theorem, another contradiction. In this case, this sentence is meaningless in the informal (and certainly the formal as well) sense. It cannot be even legitimately stated about Th $\mathfrak{N}$ since it is inherently self-contradictory. So, our argument that we used above will not work if S is replaced with Th $\mathfrak{N}$ since we will be unable to construct the necessary formal sentence $\boldsymbol{\sigma}$.

Having an intuitive sense for the theorem and a sketch of the proof, we indicate what pieces we will need to develop in the rest of the chapter before giving a rigorous proof for Gödel's Incompleteness Theorem.

The two main pieces that are missing from the proof are Gödel numbering which provides the translation of sentences in the formal language into the structure $\mathfrak{N}$ and representability within the theory Cn A. The idea is that we can translate statements about arithmetic into arithmetic and then reverse the process to talk within a specific subtheory of number theory which will then allow us to create the sentence that we need. We start with Gödel numbering.

8.2 Gödel Numbering

We will use this numbering to number every expression in the formal language. Formally, we begin this numbering with a function on the parameters and logical symbols of the language given in the table below.

$(\longmapsto 0$	$) \longmapsto 1$
$\neg \longmapsto 2$	$\rightarrow \longmapsto 3$
$\approx \longmapsto 4$	$\forall \longmapsto 5$
$\mathbf{0} \longmapsto 6$	$\boldsymbol{S} \longmapsto 7$
$< \longmapsto 8$	$\cdot \longmapsto 9$
$\boldsymbol{E} \longmapsto 10$	$\boldsymbol{v_0} \longmapsto 11$
	$\boldsymbol{v_1} \longmapsto 13$
	$\boldsymbol{v_i} \longmapsto 2i + 11$

Notice that so far, none of the even numbers past 10 are the image of any expression in the language but that all odd numbers are the image of either a logical symbol (including all of the variables) or a parameter in the language.

Recall that an expression in the language will be some finite concatenation of the symbols represented in the table above. Now, for $a_0, \ldots, a_m \in \mathbb{N}$ define the following operation on these numbers:

$$\lfloor a_0, \ldots, a_m \rfloor = \prod_{i=0}^{m} p_i^{a_i+2} \text{ if } m > 0 \text{ and}$$

$$\lfloor a_0, \ldots, a_m \rfloor = a_0 \text{ if } m = 0.$$

Note that $p_0 = 2$, $p_1 = 3$, etc. Now take an expression from the first-order language $\boldsymbol{\varepsilon} = \boldsymbol{s_0 s_2 \cdots s_m}$ where the $\boldsymbol{s_i}$'s are among the indecomposable symbols coming from the alphabet of our language to which we have already assigned natural numbers. We will define the Gödel number for the expression $\boldsymbol{\varepsilon}$ as follows:

$$\#\boldsymbol{\varepsilon} = \lfloor \#\boldsymbol{s_0}, \#\boldsymbol{s_1}, \ldots, \#\boldsymbol{s_m} \rfloor.$$

Notice that if $\boldsymbol{\varepsilon} = \boldsymbol{s}$ where $\boldsymbol{s}$ is an indecomposable symbol in the alphabet of the language, then $\#\boldsymbol{\varepsilon} = \lfloor \#\boldsymbol{s} \rfloor = \#\boldsymbol{s}$ by how we defined the $\lfloor \ldots \rfloor$ operation. Thus, this numbering for all of the expressions in the language is well defined. If Ψ is a set of expressions, then we designate the set of all the Gödel numbers of all the expressions in Ψ as

$$\#\Psi = \{\#\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} \in \Psi\}.$$

Notice that our Gödel numbering of all expressions automatically gives us numbers associated with sequences of expressions (a formal deduction is such a sequence, and we wish to encode deductions as natural numbers). The reason for this is that a sequence of expressions will ultimately be an expression itself.

Now we verify that our numbering is one-to-one. Note that as long as our expression is made up of more than one indecomposable symbol, the Gödel number for that expression will be a positive even number since it will have at least one power of 2 involved (since $2^{\#\boldsymbol{s_0}+2}$ is a factor of the product). Hence, we will not assign any expression to an odd number (to which we have assigned all of the variables and a few other indecomposable symbols) if it made up of two or more indecomposable symbols. Note also that if we have an expression built up of more than one indecomposable symbol, the smallest its Gödel number could be would be

$$\#((= \lfloor \#(, \#(\rfloor = \lfloor 0, 0 \rfloor = 2^{0+2}3^{0+2} = 36.$$

Thus, we are assured that the Gödel number of any expression built up of more than one indecomposable symbol will be an even number bigger than or equal to 36. We are thus assured that there will be no overlap with the assignments we have already made for the alphabet. The rest of the argument that the assignment of expressions to natural numbers is one-to-one follows immediately from the Fundamental Theorem of Arithmetic which guarantees a unique prime factorization for every natural number and the fact that a Gödel number of an expression must be a product of consecutive primes, all of which must have a power of at least 2. So, we will have a distinct natural number assigned to every distinct expression/sequence of expressions.

Example 8.2 *Consider the Gödel number for axiom S1 of the theory Cn A.*

$$\begin{aligned}
\#(\forall \boldsymbol{v_1}\ \boldsymbol{S v_1} \not\approx \boldsymbol{0}) &= \#(\forall \boldsymbol{v_1}(\neg \boldsymbol{S v_1} \approx \boldsymbol{0})) \\
&= \lfloor \#(\forall), \#(\boldsymbol{v_1}), \#((), \#(\neg), \#(\boldsymbol{S}), \#(\boldsymbol{v_1}), \#(\approx), \#(\boldsymbol{0}) \rfloor \\
&= \lfloor 5, 11, 0, 2, 7, 11, 4, 6 \rfloor \\
&= 2^{5+2}3^{11+2}5^{0+2}7^{7+2}11^{11+2}13^{4+2}17^{6+2} \\
&= 2^7 3^{13} 5^2 7^9 11^{13} 13^6 17^8 \\
&= 2393123115652347694455520869524208253574120928000
\end{aligned}$$

Designate this number by N_1.

Consider the Gödel number for the deduction

$$\langle \forall \boldsymbol{v_1 S v_1} \not\approx \boldsymbol{0}, (\forall \boldsymbol{v_1 S v_1} \not\approx \boldsymbol{0} \to \boldsymbol{S0} \not\approx \boldsymbol{0}), \boldsymbol{S0} \not\approx \boldsymbol{0} \rangle$$

(this deduction demonstrates that $\forall \boldsymbol{v_1 S v_1} \not\approx \boldsymbol{0} \vdash \boldsymbol{S0} \not\approx \boldsymbol{0}$ *since* $(\forall \boldsymbol{v_1 S v_1} \not\approx \boldsymbol{0} \to \boldsymbol{S0} \not\approx \boldsymbol{0})$ *is from logical axiom group 2). Note that following a similar process as above we have*

$$\#((\forall \boldsymbol{v_1 S v_1} \not\approx \boldsymbol{0} \to \boldsymbol{S0} \not\approx \boldsymbol{0})) = 2^2 3^7 5^{13} 7^2 11^4 13^9 17^{13} 19^6 23^8 29^3 31^5 37^2 41^4 43^9 47^8 53^6 59^8 61^3 67^3$$

Denote this number by N_2. *Also,*

$$\#(\boldsymbol{S0} \not\approx \boldsymbol{0}) = 2^2 3^4 5^9 7^8 11^6 13^8 17^3.$$

Denote this number by N_3.

$$
\begin{aligned}
&\#(\langle \forall \boldsymbol{v_1 S v_1} \not\approx \mathbf{0}, (\forall \boldsymbol{v_1 S v_1} \not\approx \mathbf{0} \rightarrow \boldsymbol{S}\mathbf{0} \not\approx \mathbf{0}), \boldsymbol{S}\mathbf{0} \not\approx \mathbf{0}\rangle)\\
&= \lfloor \#(\forall \boldsymbol{v_1 S v_1} \not\approx \mathbf{0}), \#((\forall \boldsymbol{v_1 S v_1} \not\approx \mathbf{0} \rightarrow \boldsymbol{S}\mathbf{0} \not\approx \mathbf{0})), \#(\boldsymbol{S}\mathbf{0} \not\approx \mathbf{0}) \rfloor\\
&= \lfloor N_1, N_2, N_3 \rfloor\\
&= 2^{N_1+2} 3^{N_2+2} 5^{N_3+2}\\
&= 2^{2^7 3^{13} 5^2 7^9 11^{13} 13^6 17^8 + 2} 3^{2^2 3^7 5^{13} 7^2 11^4 13^9 17^{13} 19^6 23^8 29^3 31^5 37^2 41^4 43^9 47^8 53^6 59^8 61^3 67^3 + 2}\\
&\cdot 5^{2^2 3^4 5^9 7^8 11^6 13^8 17^3 + 2}
\end{aligned}
$$

Obviously, the numbers involved in Gödel numbering are large even for small expressions, but Gödel numbering allows us to encode all expressions of our first-order language into the universe for the structure $\mathfrak{N}$. By one-to-oneness, we can also uniquely decode the images of the expressions and sequences of expressions back into the original expressions and sequences of expressions. This will be key to the proof of Gödel's Theorem, and we give a brief example of the decoding process.

Example 8.3 *Take the natural number* 16, 669, 800*. The prime factorization of this number is*

$$2^3 3^5 5^2 7^3 = 2^{1+2} 3^{3+2} 5^{0+2} 7^{1+2} = \lfloor 1, 3, 0, 1 \rfloor.$$

Looking at the table for the assignments for the alphabet, we have the expression $) \rightarrow ()$.

Note that in general, not every natural number has an expression associated with it. For instance, the number 36, 432 *factors as* $2^4 3^2 \cdot 11 \cdot 23$*. Since we do not have a product of consecutive primes, we know that this is not the Gödel number of an expression in the language.*

8.3 Representability in Cn A

In this section we discuss how we can translate numerical relations in $\mathfrak{N}$ back into the formal relations. In other words, we discuss how certain relations are able to be represented in the formal language. This will be the other piece that we need to formally prove Gödel's Theorem.

First, we develop some new notation. For the first-order formula $\boldsymbol{\varphi}$, let $\boldsymbol{\varphi}(\boldsymbol{t}) = \boldsymbol{\varphi}_{\boldsymbol{v_1}|\boldsymbol{t}}$, $\boldsymbol{\varphi}(\boldsymbol{t_1}, \boldsymbol{t_2}) = (\boldsymbol{\varphi}_{\boldsymbol{v_1}|\boldsymbol{t_1}})_{\boldsymbol{v_2}|\boldsymbol{t_2}}$, etc. Now, recall the definition of a definable relation.

Definition 8.4 *Given the structure $\mathfrak{S}$ and the wff $\boldsymbol{\varphi}$, whose free variables are among $\boldsymbol{v_1}$, $\boldsymbol{v_2}$,..., $\boldsymbol{v_k}$,*

$$\{(u_1, u_2, \ldots, u_k) : \models_{\mathfrak{S}} \boldsymbol{\varphi}[[u_1, u_2, \ldots, u_k]]\}$$

is the **relation that $\boldsymbol{\varphi}$ defines** *in $\mathfrak{S}$. Given a k-ary relation R in the universe $\mathbb{U}$ determined by $\mathfrak{S}$, if there is a $\boldsymbol{\varphi}$ such that*

$$R = \{(u_1, u_2, \ldots, u_k) : \models_{\mathfrak{S}} \boldsymbol{\varphi}[[u_1, u_2, \ldots, u_k]]\},$$

then R is said to be **definable** *in $\mathfrak{S}$.*

Take as our structure $\mathfrak{N}$ whose universe is $\mathbb{N}$. Then for a k-ary relation R in the universe $\mathbb{N}$, this relation is definable in $\mathfrak{N}$, if and only if there exists a formula $\boldsymbol{\varphi}$ such that for all $(n_1, n_2, \ldots, n_k) \in \mathbb{N}^k$,

$$(n_1, n_2, \ldots n_k) \in R \text{ iff } \models_{\mathfrak{N}} \boldsymbol{\varphi}[[n_1, n_2, \ldots, n_k]].$$

Recall that this last statement will be so if and only if there is a function $s : \mathcal{V} \longrightarrow \mathbb{N}$ such that $s(\boldsymbol{v_i}) = n_i$ and $\models_{\mathfrak{N}} \boldsymbol{\varphi}[s]$, and by Theorem 4.27, $\models_{\mathfrak{N}} \boldsymbol{\varphi}[s]$ for every function such that $s(\boldsymbol{v_i}) = n_i$. Note also that $s(\boldsymbol{v_i}) = n_i = \overline{s}(\boldsymbol{S}^{n_i}\boldsymbol{0})$

for any function $s : \mathcal{V} \longrightarrow \mathbb{N}$. Let

$$s^{1\le i\le k}_{\boldsymbol{v_i}|n_i} \equiv (\cdots((s_{\boldsymbol{v_1}|n_1})_{\boldsymbol{v_2}|n_2})\cdots)_{\boldsymbol{v_k}|n_k}.$$

Then,

$$s^{1\le i\le k}_{\boldsymbol{v_i}|n_i}(\boldsymbol{x}) = \begin{cases} n_i & \text{if } \boldsymbol{x} = \boldsymbol{v_i} \text{ for some } 1 \le i \le k \\ s(y) & \text{otherwise} \end{cases}$$

Of course, since $n_i = \overline{s}(\boldsymbol{S}^{n_i}\mathbf{0})$, the function above is the same as the following function:

$$s^{1\le i\le k}_{\boldsymbol{v_i}|\overline{s}(\boldsymbol{S}^{n_i}\mathbf{0})}(\boldsymbol{x}) = \begin{cases} \overline{s}(\boldsymbol{S}^{n_i}\mathbf{0}) & \text{if } \boldsymbol{x} = \boldsymbol{v_i} \text{ for some } 1 \le i \le k \\ s(y) & \text{otherwise} \end{cases}$$

Hence,

$\models_{\mathfrak{N}} \boldsymbol{\varphi}[[n_1, n_2, \dots, n_k]]$ iff there is a function s as described above such that

$$\models_{\mathfrak{N}} \boldsymbol{\varphi}[s] \text{ iff}$$

$$\models_{\mathfrak{N}} \boldsymbol{\varphi}\left[s^{1\le i\le k}_{\boldsymbol{v_i}|\overline{s}(\boldsymbol{S}^{n_i}\mathbf{0})}\right].$$

Now, since $\boldsymbol{S}^{n_i}\mathbf{0}$ involves no variables, it will always be substitutable for any variable symbol $\boldsymbol{x}$. Thus, by repeated application of the Substitution Lemma (Lemma 6.1.2) and by applying the new notation developed above,

$$\models_{\mathfrak{N}} \boldsymbol{\varphi}\left[s^{1\le i\le k}_{\boldsymbol{v_i}|\overline{s}(\boldsymbol{S}^{n_i}\mathbf{0})}\right] \text{ iff } \models_{\mathfrak{N}} \boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \dots, \boldsymbol{S}^{n_k}\mathbf{0})[s].$$

Since $\boldsymbol{\varphi}$'s free variables were assumed to be among $\boldsymbol{v_1}$, $\boldsymbol{v_2}$,…,$\boldsymbol{v_k}$, $\boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \dots, \boldsymbol{S}^{n_k}\mathbf{0})$ is in fact a sentence, since it has no free variables, whence

$$\models_{\mathfrak{N}} \boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \dots, \boldsymbol{S}^{n_k}\mathbf{0})[s] \text{ iff } \models_{\mathfrak{N}} \boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \dots, \boldsymbol{S}^{n_k}\mathbf{0}).$$

So, a k-ary relation R on $\mathbb{N}$ will be definable if and only if for every $(n_1, n_2, \dots, n_k) \in \mathbb{N}^k$,

$$(n_1, n_2, \dots, n_k) \in R \text{ iff } \models_{\mathfrak{N}} \boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \dots, \boldsymbol{S}^{n_k}\mathbf{0}).$$

If we split this last statement into two implications we have that

$$(n_1, n_2, \ldots, n_k) \in R \text{ implies } \models_{\mathfrak{N}} \boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0}) \text{ and}$$

$$(n_1, n_2, \ldots, n_k) \notin R \text{ implies } \models_{\mathfrak{N}} \boldsymbol{(\neg\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0})\boldsymbol{)}$$

if and only if R is definable in $\mathfrak{N}$.

Now let T be a theory in a language with the symbols $\mathbf{0}$ and $\boldsymbol{S}$. We define the following:

Definition 8.5 *The formula* $\boldsymbol{\varphi}$ **represents** *the* k*-ary relation* R *of the universe of a model for* T *if*

$$(n_1, n_2, \ldots, n_k) \in R \textit{ implies } \boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0}) \in T \textit{ and}$$

$$(n_1, n_2, \ldots, n_k) \notin R \textit{ implies } \boldsymbol{(\neg\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0})\boldsymbol{)} \in T.$$

A relation R *is said to be* **representable in theory** $\boldsymbol{T}$ *if there is a formula* $\boldsymbol{\varphi}$ *that represents* R *in that theory.*

Representability is a stronger notion than definability. Suppose $\mathfrak{S}$ is a model for the sentences in T (with universe $\mathbb{N}$) and suppose that $\boldsymbol{\varphi}$ represents the relation R in the theory T. Then, if $(n_1, n_2, \ldots, n_k) \in R$, then $\boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0}) \in T$, and of course $\models_{\mathfrak{S}} \boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0})$, which, by the previous discussion is so if and only if $\models_{\mathfrak{S}} \boldsymbol{\varphi}[[n_1, n_2, \ldots, n_k]]$ for any function $s : \mathcal{V} \longrightarrow \mathbb{N}$ such that $s(\boldsymbol{v_i}) = n_i$. If on the other hand, $(n_1, n_2, \ldots n_k) \notin R$, then $\boldsymbol{(\neg\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0})\boldsymbol{)} \in T$, and $\not\models_{\mathfrak{S}} \boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0})$. Again, by our previous discussion, this means that $\not\models_{\mathfrak{S}} \boldsymbol{\varphi}[[n_1, n_2, \ldots, n_k]]$. Thus, we have that $\boldsymbol{\varphi}$ defines R in $\mathfrak{S}$, a model for T, if $\boldsymbol{\varphi}$ represents R in T. However, if $\boldsymbol{\varphi}$ defines a relation R in a structure $\mathfrak{S}$ for the language, $\boldsymbol{\varphi}$ will only be guaranteed to be representable in the theory Th $\mathfrak{S}$.

What we will be most concerned about is representability of a relation of natural numbers in the theory Cn A. Following the definitions of representability for the theory Cn A, a relation R will be representable in Cn A if and only if

$$(n_1, n_2, \ldots, n_k) \in R \text{ implies } A \vdash \boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0}) \text{ and}$$

$$(n_1, n_2, \ldots, n_k) \notin R \text{ implies } A \vdash (\neg\boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0})).$$

Example 8.6 *As a simple example, the equality relation is representable in Cn A. If $m = n$, then clearly $\boldsymbol{S}^m\mathbf{0} = \boldsymbol{S}^n\mathbf{0}$. Clearly then, by one of the equality logical axioms, $\vdash \boldsymbol{S}^m\mathbf{0} \approx \boldsymbol{S}^n\mathbf{0}$, and $A \vdash \boldsymbol{S}^m\mathbf{0} \approx \boldsymbol{S}^n\mathbf{0}$. If $m \neq n$, then without loss of generality $m > n$, and there is $k > 0$ such that $m = n + k$. Now, by logical axioms and axiom S1, $S1 \vdash \boldsymbol{S}^k\mathbf{0} \not\approx \mathbf{0}$. By n applications of axiom S2, and formal contraposition, we can obtain that $\{S1, S2\} \vdash (\neg\boldsymbol{S}^{\boldsymbol{m}}\mathbf{0} \approx \boldsymbol{S}^{\boldsymbol{n}}\mathbf{0})$. Hence, equality may be represented in Cn A.*

We need to demonstrate/claim that several relations are representable in Cn A leading up to showing that Gödel numbering is representable in Cn A. The following definition and theorem is helpful in establishing results of representability.

Definition 8.7 *Let $\boldsymbol{\varphi}$ be a formula in which only the variables $\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_k}$ occur free. Then $\boldsymbol{\varphi}$ is* **numeralwise determined** *by A if for any k-tuple $(n_1, n_2, \ldots, n_k)$ either*

$$A \vdash \boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0}) \text{ or}$$

$$A \vdash (\neg\boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0})).$$

Theorem 8.8 *A formula* $\boldsymbol{\varphi}$ *represents a relation* R *in* $Cn\ A$ *if and only if*

(i) $\boldsymbol{\varphi}$ *is numeralwise determined by* A*, and*

(ii) $\boldsymbol{\varphi}$ *defines* R *in* $\mathfrak{N}$.

Proof: If $\boldsymbol{\varphi}$ represents a relation R in Cn A, then it is clear by the definition of representable relation applied to Cn A, that the two items hold since $\mathfrak{N}$ is a model of A, and hence all of Cn A (recall the discussion above about how representability is stronger than definability).

Suppose now that the two items hold for the relation R. If $(n_1, n_2, \ldots, n_k) \in R$, then since $\boldsymbol{\varphi}$ defines R in $\mathfrak{N}$, then $\models_{\mathfrak{N}} \boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0})$. Since $\mathfrak{N}$ is a model of A, then we must $A \nvdash (\neg\boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0}))$, otherwise we have a contradiction. By item (i) and the definition of "numeralwise determined," $A \vdash \boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0})$. The argument if $(n_1, n_2, \ldots, n_k) \notin R$ is completely analogous. We just need the observation that

$$(\neg(\neg\boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0})))$$

is tautologically equivalent to $\boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \boldsymbol{S}^{n_2}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0})$ and Theorem 5.6. ∎

Numeralwise determination has the following closure properties under the formula building operations indicated in the following theorem (stated without proof).

Theorem 8.9 *(i) Any atomic formula is numeralwise determined by* A.

(ii) If $\boldsymbol{\varphi}$ *and* $\boldsymbol{\psi}$ *are numeralwise determined by* A*, then so are* $(\neg\boldsymbol{\varphi})$ *and* $(\boldsymbol{\varphi} \rightarrow \boldsymbol{\psi})$.

(iii) If $\boldsymbol{\varphi}$ *is numeralwise determined by* A*, then so are the following formulas:*

$$\forall \boldsymbol{x}(\boldsymbol{x} < \boldsymbol{y} \rightarrow \boldsymbol{\varphi})$$

$$\exists \boldsymbol{x}(\boldsymbol{x} < \boldsymbol{y} \wedge \boldsymbol{\varphi}).$$

In what follows, we will also need the following notion.

Definition 8.10 *A relation* R *on the natural numbers is* **recursive** *if and only if it is representable in some consistent finitely axiomatizable theory (in a language with* $\mathbf{0}$ *and* $\boldsymbol{S}$*).*

Hence, if we can show that a relation is representable in Cn A it is recursive by definition. This notion of recursive relation is the precise notion of a decidable set (relation) discussed in Section 3.3 by what is known as Church's Thesis.

Church's Thesis A relation is decidable if and only if it is recursive.

The idea is that the effective procedure that we use is encoded in the formal language and vice versa. Intuitively, the "guts" behind a computer program are arithmetical operations with 1's and 0's, or 0's and the successor function applied to 0 a finite number of times. Thus, recursiveness is the formal counterpart to decidability.

We have a definition for what it means to represent a relation in a theory. From this point forward when we say that a relation R is representable, we will mean that it is representable in the theory Cn A. We extend the notion of representability to functions (a specific type of relation).

Definition 8.11 *Let $f : \mathbb{N}^k \longrightarrow \mathbb{N}$. A formula $\boldsymbol{\varphi}$ in which only $\boldsymbol{v_1}, \ldots, \boldsymbol{v_{k+1}}$ occur free will be said to* **functionally represent** $\boldsymbol{f}$ *(in the theory Cn A) if for any $n_1, \ldots, n_k \in \mathbb{N}$,*

$$A \vdash \forall \boldsymbol{v_{k+1}}[\boldsymbol{\varphi}(\boldsymbol{S}^{n_1}\mathbf{0}, \ldots, \boldsymbol{S}^{n_k}\mathbf{0}) \leftrightarrow \boldsymbol{v_{k+1}} \approx \boldsymbol{S}^{\boldsymbol{f(n_1,\ldots,n_k)}}\mathbf{0}]$$

The following theorems are intuitively plausible, and so we omit their proof.

Theorem 8.12 *If $\boldsymbol{\varphi}$ functionally represents f in Cn A, then it also represents f (as a relation) in Cn A.*

Theorem 8.13 *Let f be a function on $\mathbb{N}$ which is (as a relation) representable in Cn A. Then we can find a formula $\boldsymbol{\varphi}$ which functionally represents f in Cn A.*

Now we move into discussing the representability of functions and relations. Ultimately, this discussion will lead up to representability dealing with Gödel numbering. This representability with Gödel numbers will be the final piece required to prove the Incompleteness Theorem. We will state most of the following theorems without proof. Proofs and further development may be found in *A Mathematical Introduction to Logic* by Herbert Enderton on pages 202-227.

Theorem 8.14 *The following functions are representable:*

(i) The successor function by the formula $\boldsymbol{v_2} \approx \boldsymbol{S}\boldsymbol{v_1}$.

(ii) Any m-place constant function whose output is some b is represented by the formula $\boldsymbol{v_{m+1}} \approx \boldsymbol{S}^b\mathbf{0}$.

(iii) The projection function $I_i^m(a_1, a_2, \ldots, a_i, \ldots, a_m) = a_i$ *for any* $m \in \mathbb{Z}^+$ *and* $1 \le i \le m$ *is represented by* $\boldsymbol{v_{m+1} \approx v_i}$.

(iv) Addition, multiplication, and exponentiation are represented by the equations

$$\boldsymbol{v_3 \approx v_1 + v_2}$$

$$\boldsymbol{v_3 \approx v_1 \cdot v_2}$$

$$\boldsymbol{v_3 \approx v_1 E v_2}$$

respectively.

Theorem 8.15 *The class of representable functions is closed under composition.*

In what follows, it will be convenient to denote $\overrightarrow{a} = (a_1, a_2, \ldots, a_m)$ for any given $m \in \mathbb{Z}^+$.

Theorem 8.16 *(i) Any relation which has in* $\mathfrak{N}$ *a quantifier-free definition is representable.*

(ii) The class of representable relations is closed under unions, intersections, and complements.

(iii) If R *is representable, then so are*

$$\{(\overrightarrow{a}, b) : \textit{ for all } c < b \ (\overrightarrow{a}, c) \in R\} \textit{ and}$$

$$\{(\overrightarrow{a}, b) : \textit{ for some } c < b \ (\overrightarrow{a}, c) \in R\}.$$

Recall that for any set S, the characteristic function of S, denoted K_S is the function,

$$K_S(\overrightarrow{x}) = \begin{cases} 1 & \text{if } \overrightarrow{x} \in S \\ 0 & \text{if } \overrightarrow{x} \in S^{\complement} \end{cases}$$

where $S^{\complement}$ is the complement of the set S in the specified universe (note that S could be a set of m-tuples).

Theorem 8.17 *A relation R is representable if and only if its characteristic function K_R is.*

Theorem 8.18 *If R is a representable binary relation and f and g are representable functions, then $\{\overrightarrow{a} : (f(\overrightarrow{a}), g(\overrightarrow{a})) \in R\}$ is representable. Similarly for an m-ary relation R and functions $f_1, \ldots, f_m$.*

Theorem 8.19 *If R is a representable $(m+1)$-ary relation, then so is*

$$\{(\overrightarrow{a}, b) : \text{ for some } c \leq b, (\overrightarrow{a}, c) \in R\}.$$

Similarly for $\{(\overrightarrow{a}, b) : \text{ for all } c \leq b, (\overrightarrow{a}, c) \in R\}$.

Theorem 8.20 *The divisibility relation $\{(a, b) : a \text{ divides } b \text{ in } \mathbb{N}\}$ is representable.*

Theorem 8.21 *The set of primes is representable.*

Theorem 8.22 *The set of pairs of adjacent primes is representable.*

Theorem 8.23 *The mapping $a \longmapsto p_a$ where $a \in \mathbb{N}$, and p_a is the $(a+1)st$ prime is representable.*

Recall from the section on Gödel numbering that

$$\lfloor a_0, \ldots, a_m \rfloor = \prod_{i=0}^{m} p_i^{a_i+2} \text{ if } m > 0 \text{ and}$$

$$\lfloor a_0, \ldots, a_m \rfloor = a_0 \text{ if } m = 0.$$

We may extend this operation even further and say that for $m = -1$, we define the operation as $\lfloor \rfloor = 1$.

Theorem 8.24 *For each $m \in \mathbb{N}$, the mapping*

$$(a_0, a_1, \ldots, a_m) \longmapsto \lfloor a_0, a_1, \ldots, a_m \rfloor$$

is representable.

Theorem 8.25 *There is a representable function such that $(a, b) \longmapsto (a)_b$ where for $b \leq m$,*

$$(\lfloor a_0, \ldots, a_m \rfloor)_b = a_b$$

The last function is a decoding function for Gödel numbering so that if we had the Gödel number of an expression in the formal language, we can decode its Gödel number into the numbers that we assigned to the alphabet of the formal language and subsequently retrieve the original formal expression. Their representability essentially allows us to refer to Gödel numbering in the formal language itself. Thus, we have the potential to have formulas in the language that refer to their own Gödel number. This self-reference will be key to proving Gödel's Theorem.

Theorem 8.26 *Assume that R is a representable relation such that for every $\overrightarrow{a}$ there is some n such that $(\overrightarrow{a}, n) \in R$. Then the function defined by $f(\overrightarrow{a}) = \min\{n : (\overrightarrow{a}, n) \in R\}$ is representable.*

Definition 8.27 *Say that b is a* **sequence number** *if for some $m \geq -1$ and some $a_0, \ldots, a_m$, $b = \lfloor a_0, \ldots, a_m \rfloor$.*

This definition allows us to distinguish between which natural numbers are the images of a formal expression under Gödel numbering and which are not.

Theorem 8.28 *The set of sequence numbers is representable.*

Theorem 8.29 *There is a representable function* $lh : \mathbb{N} \longrightarrow \mathbb{N}$ *such that* $lh(\lfloor a_0, \ldots, a_m \rfloor) = m + 1$ *("lh" is intended to stand for "length").*

Definition 8.30 *Define a mapping from* $\mathbb{N} \times \mathbb{N}$ *into* $\mathbb{N}$*, called the* **restriction***, where* $(a, b) \longmapsto a \restriction b$*, and*

$$a \restriction b = \begin{cases} \lfloor a_0, \ldots, a_{b-1} \rfloor & \text{if } a \text{ is a sequence number such that } a = \lfloor a_0, \ldots, a_m \rfloor \\ & \text{and } b \leq m+1 \\ 0 & \text{otherwise} \end{cases}$$

Theorem 8.31 *The restriction function is representable.*

Given a $(k+1)$-place function f, we design a new function $\widehat{f}$ where

$$\widehat{f}(a, b_1, \ldots, b_k) = \lfloor f(0, \overrightarrow{b}), \ldots, f(a-1, \overrightarrow{b}) \rfloor$$

. This function $\widehat{f}$ will encode the first a values of $f(x, \overrightarrow{b})$. Suppose now that we have a $(k+2)$-place function g. Applying the Recursion Theorem, we may say that there is a unique $(k+1)$ function f such that

$$f(a, \overrightarrow{b}) = g(\widehat{f}(a, \overrightarrow{b}), a, \overrightarrow{b}).$$

These preliminary results are necessary to state the following theorem.

Theorem 8.32 *Let* g *be a* $(k+2)$*-place function and let* f *be the unique* $(k+1)$*-place function such that for all* a *and (*k*-tuples)* $\overrightarrow{b}$*,*

$$f(a, \overrightarrow{b}) = g(\widehat{f}(a, \overrightarrow{b}), a, \overrightarrow{b}).$$

If g *is representable, then so is* f*.*

Theorem 8.33 *For a representable function* F*, the mapping defined by*

$$(a, \overrightarrow{b}) \longmapsto \prod_{i<a} F(i, \overrightarrow{b})$$

is also representable. Similarly with Σ *in place of* $\prod$*.*

Definition 8.34 *If a and b are sequence numbers, define a binary operation between them, called* **concatenation**, *denoted by $\otimes$ where*

$$a \otimes b = a \cdot \prod_{i<lhb} p_{i+lha}^{(b)_i+2}$$

.

Note that

$$\lfloor a_0, \ldots, a_m \rfloor \otimes \lfloor b_0, \ldots, b_n \rfloor = \lfloor a_0, \ldots, a_m, b_0, \ldots, a_n \rfloor.$$

The operation will also be associative.

Theorem 8.35 *Concatenation is a representable function.*

Let $\bigotimes_{i<a} f(i) = f(0) \otimes f(1) \otimes \cdots \otimes f(a-1)$.

Theorem 8.36 *For a representable F, the mapping defined by*

$$(a, \overrightarrow{b}) \longmapsto \bigotimes_{i<a} F(i, \overrightarrow{b})$$

is representable.

These results about representable functions are more general but they are necessary to establish results about the representability dealing with Gödel numbering which we now state.

Theorem 8.37 *The set (unary relation) of Gödel numbers of variables is representable.*

Proof: Given our assignment of numbers to the variable symbols, it is clear that the set of Gödel numbers of variables is given by

$$\{a : \text{ there is } b \text{ less than } a \text{ such that } a = \lfloor 11 + 2b \rfloor = 11 + 2b\}$$

(recall that $\#\boldsymbol{v_i} = 11 + 2i$ and that $\lfloor a_0 \rfloor = a_0$). This relation will be representable by Theorem 8.16 and the fact that addition, multiplication, and any constant function are representable. ∎

Theorem 8.38 *The set of Gödel numbers of terms is representable.*

Proof: (sketch)

Let $\mathcal{T}$ be the set of terms in the formal language and then consider $K_{\mathcal{T}}$, the characteristic function of the set of terms. The terms were constructed via induction and so the characteristic function will refer to itself in the conditions for when the characteristic function takes on 1. The idea is to apply Theorem 8.32, and use a function g which eliminates direct reference to the characteristic function. The goal will then be to show that the function g will be representable (the results stated above will guarantee this) in which case the characteristic function will be representable and hence the set of terms by Theorem 8.17. ∎

Theorem 8.39 *The set of Gödel numbers of atomic formulas is representable.*

Theorem 8.40 *The set of Gödel numbers of wffs is representable.*

The proof for wffs is in the same spirit as that used for terms.

Theorem 8.41 *There is a representable function Sb (Sb is supposed to abbreviate "substitution") such that for a term or formula* $\boldsymbol{\alpha}$*, variable* $\boldsymbol{x}$*, and term* $\boldsymbol{t}$*,*

$$Sb(\#\boldsymbol{\alpha}, \#\boldsymbol{x}, \#\boldsymbol{t}) = \#\boldsymbol{\alpha}_{\boldsymbol{x}|\boldsymbol{t}}$$

Proof: (sketch) The function *Sb* can be defined in cases. That the function is representable follows the same type of argument as that employed with the Gödel numbering for terms. ∎

Theorem 8.42 *The function whose value at* n *is* $\#(\boldsymbol{S}^n\mathbf{0})$ *is representable*

Theorem 8.43 *There is a representable relation* Fr *such that for a term or formula* $\boldsymbol{\varphi}$ *and a variable* $\boldsymbol{x}$*,*

$$(\#\boldsymbol{\varphi}, \#\boldsymbol{x}) \in Fr \text{ iff } \boldsymbol{x} \text{ occurs free in } \boldsymbol{\varphi}.$$

Theorem 8.44 *The set of Gödel numbers of sentences is representable.*

Theorem 8.45 *There is a representable relation* Sbl *such that for a formula* $\boldsymbol{\varphi}$*, variable* $\boldsymbol{x}$*, and term* $\boldsymbol{t}$*,*

$$(\#\boldsymbol{\varphi}, \#\boldsymbol{x}, \#\boldsymbol{t}) \in Sbl \text{ iff } t \text{ is substitutable for } \boldsymbol{x} \text{ in } \boldsymbol{\varphi}.$$

Theorem 8.46 *The relation* Gen*, where* $(a, b) \in Gen$ *if and only if* a *is the Gödel number of a formula and* b *is the Gödel number of a generalization of that formula, is representable.*

Theorem 8.47 *The set of Gödel numbers of logical axioms is representable.*

Theorem 8.48 *For a finite set of formulas* S*,*

$$\{\#D : D \text{ is a deduction from } S\}$$

is representable.

Theorem 8.49 *A relation is recursive if and only if it is representable in the theory Cn* A*.*

Proof: That a relation representable in the theory Cn A is recursive follows from the definition of recursive. Suppose now that we have a recursive relation. A relation is recursive if there is a consistent finite set of sentences S

such that some formula $\boldsymbol{\varphi}$ represents R in Cn S. Suppose R is a k-ary relation. Then,

$$(a_1, \ldots, a_k) \in R \text{ implies } S \vdash \boldsymbol{\varphi}(\boldsymbol{S}^{a_1}\mathbf{0}, \ldots, \boldsymbol{S}^{a_k}\mathbf{0}) \text{ and}$$

$$(a_1, \ldots, a_k) \notin R \text{ implies } S \vdash (\neg\boldsymbol{\varphi}(\boldsymbol{S}^{a_1}\mathbf{0}, \ldots, \boldsymbol{S}^{a_k}\mathbf{0})).$$

So, if $(a_1, \ldots, a_k) \in R$ there is deduction D of $\boldsymbol{\varphi}(\boldsymbol{S}^{a_1}\mathbf{0}, \ldots, \boldsymbol{S}^{a_k}\mathbf{0})$ from the set S. Thus $\#D$ is the set in the previous theorem, and the last number used to encode the deduction is $\#(\boldsymbol{\varphi}(\boldsymbol{S}^{a_1}\mathbf{0}, \ldots, \boldsymbol{S}^{a_k}\mathbf{0}))$. So, we may define the mapping

$$(a_1, \ldots, a_k) \longmapsto d \text{ where } d \text{ is the least sequence number } d \text{ such that the last}$$
$$\text{number in the sequence is } \#(\boldsymbol{\varphi}(\boldsymbol{S}^{a_1}\mathbf{0}, \ldots, \boldsymbol{S}^{a_k}\mathbf{0})).$$

We may do similarly for $(a_1, \ldots, a_k) \notin R$ and $\#((\neg\boldsymbol{\varphi}(\boldsymbol{S}^{a_1}\mathbf{0}, \ldots, \boldsymbol{S}^{a_k}\mathbf{0})))$. Since the Gödel numbers of deductions are representable in Cn A, R will be representable in Cn A. ∎

Corollary 8.49.1 *Any recursive relation is definable in* $\mathfrak{N}$.

Proof: This follows immediately from the fact that representability implies definability. ∎

Theorem 8.50 *Let S be a set of sentences. If $\#S$ is recursive, and Cn S is a complete theory, then $\#$Cn S is recursive.*

The proof for this theorem follows in the same spirit as the proof for the last theorem.

At this point, we have all of the representability results that we will need to prove Gödel's Incompleteness Theorem.

8.4 Gödel's Incompleteness Theorem

The linchpin for Gödel's Theorem is the following lemma.

Lemma 8.4.1 (Fixed-Point Lemma) *Given any formula* $\boldsymbol{\beta}$ *in which only* $\boldsymbol{v_1}$ *occurs free, we can find a sentence* $\boldsymbol{\sigma}$ *such that* $A \vdash \boldsymbol{\sigma} \leftrightarrow \boldsymbol{\beta}(\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0})$

This lemma says a lot. We are given the formula $\boldsymbol{\beta}$ which "expresses" some property. The idea is that we can find a sentence $\boldsymbol{\sigma}$ and compute its Gödel number $\#\boldsymbol{\sigma}$. Then we can indirectly represent $\boldsymbol{\sigma}$ in the formal language as $\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0}$. Then $\boldsymbol{\beta}(\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0})$ is true if and only if $\boldsymbol{\sigma}$ is true. Since $\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0}$ indirectly represents $\boldsymbol{\sigma}$, what the lemma indicates is that $\boldsymbol{\beta}$ is true of $\boldsymbol{\sigma}$ if and only if $\boldsymbol{\sigma}$ is true. So, we can think of $\boldsymbol{\sigma}$ indirectly asserting that $\boldsymbol{\beta}$ is true of itself.

Proof: Consider the function Sb as given in Theorem 8.41. For a fixed variable, this function becomes $Sb(\#\boldsymbol{\alpha}, \#\boldsymbol{t}) = \#(\boldsymbol{\alpha}(\boldsymbol{t}))$. Taking $\boldsymbol{t} = \boldsymbol{S}^n\mathbf{0}$, we have $Sb(\#\boldsymbol{\alpha}, \#\boldsymbol{S}^n\mathbf{0}) = \#(\boldsymbol{\alpha}(\boldsymbol{S}^n\mathbf{0}))$. Now, we have by Theorem 8.42, that the mapping $n \longmapsto \#(\boldsymbol{S}^n\mathbf{0})$ is a representable function. Hence, the function $g(\#\boldsymbol{\alpha}, n) = Sb(\#\alpha, \#(\boldsymbol{S}^n\mathbf{0})) = \#(\boldsymbol{\alpha}(\boldsymbol{S}^n\mathbf{0}))$ is functionally representable since it is a composition of representable functions. Thus, we are justified by the definition of functional representation in saying that there is a formula $\boldsymbol{\theta}$ in which $\boldsymbol{v_1}$,$\boldsymbol{v_2}$, and $\boldsymbol{v_3}$ occur free such that $\boldsymbol{\theta}$ (or equivalently $\boldsymbol{\theta}(\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3})$) functionally represents g.

Consider now the formula

$$\forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{v_1}, \boldsymbol{v_1}, \boldsymbol{v_3}) \rightarrow \boldsymbol{\beta}\ (\boldsymbol{v_3})]$$

which only has the variable $\boldsymbol{v_1}$ free. For shorthand, we will refer to this formula as $\boldsymbol{\delta}$. We claim that this formula defines a set such that $\#\boldsymbol{\alpha}$ ($\boldsymbol{\alpha}$ is a formula) is in the set if and only if $\#(\boldsymbol{\alpha}(\boldsymbol{S}^{\#\boldsymbol{\alpha}}\mathbf{0}))$ is in the set defined by $\boldsymbol{\beta}$ (which only has the variable $\boldsymbol{v_1}$ free).

For the details, assume $\#\boldsymbol{\alpha} \in \{n :\models_{\mathfrak{N}} \boldsymbol{\delta}(\boldsymbol{S}^n\mathbf{0})\}$ (the set defined by $\boldsymbol{\delta}$). Thus, $\models_{\mathfrak{N}} \boldsymbol{\delta}(\boldsymbol{S}^{\#\boldsymbol{\alpha}}\mathbf{0})$. Now,

$$\boldsymbol{\delta}(\boldsymbol{S}^{\#\boldsymbol{\alpha}}\mathbf{0}) = \forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{S}^{\#\boldsymbol{\alpha}}\mathbf{0}, \boldsymbol{S}^{\#\boldsymbol{\alpha}}\mathbf{0}, \boldsymbol{v_3}) \to \boldsymbol{\beta}\ (\boldsymbol{v_3})].$$

Since $\boldsymbol{\theta}(\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3})$ functionally represents the function g discussed earlier, by definition of functional representation (and since $\mathfrak{N}$ is a model of A), we must have

$$\models_{\mathfrak{N}} \forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{S}^{\#\boldsymbol{\alpha}}\mathbf{0}, \boldsymbol{S}^{\#\boldsymbol{\alpha}}\mathbf{0}, \boldsymbol{v_3}) \leftrightarrow \boldsymbol{v_3} \approx \boldsymbol{S}^{\#(\boldsymbol{\alpha}(\boldsymbol{S}^{\#\boldsymbol{\alpha}}\mathbf{0}))}\mathbf{0}]$$

where $g(\#\boldsymbol{\alpha}, \#\boldsymbol{\alpha}) = \#(\boldsymbol{\alpha}(\boldsymbol{S}^{\#\boldsymbol{\alpha}}\mathbf{0}))$. It will be clear then that

$$\models_{\mathfrak{N}} \boldsymbol{\delta}(\boldsymbol{S}^{\#\boldsymbol{\alpha}}\mathbf{0}) \text{ iff } \models_{\mathfrak{N}} \forall \boldsymbol{v_3}[\boldsymbol{v_3} \approx \boldsymbol{S}^{\#(\boldsymbol{\alpha}(\boldsymbol{S}^{\#\boldsymbol{\alpha}}\mathbf{0}))}\mathbf{0} \to \boldsymbol{\beta}(\boldsymbol{v_3})].$$

This statement implies that $\models_{\mathfrak{N}} \boldsymbol{\beta}(\boldsymbol{S}^{\#(\boldsymbol{\alpha}(\boldsymbol{S}^{\#\boldsymbol{\alpha}}\mathbf{0}))}\mathbf{0})$. By definition then, $\#(\boldsymbol{\alpha}(\boldsymbol{S}^{\#\boldsymbol{\alpha}}\mathbf{0})) \in \{n : \models_{\mathfrak{N}} \boldsymbol{\beta}(\boldsymbol{S}^n\mathbf{0})\}$. The chain of implications can be reversed to show the equivalence using the definition of satisfaction of a universal quantifier.

Given this information, we now define $\boldsymbol{\sigma}$. Let q be the Gödel number of

$$\forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{v_1}, \boldsymbol{v_1}, \boldsymbol{v_3}) \to \boldsymbol{\beta}\ (\boldsymbol{v_3})].$$

That is $\#\boldsymbol{\delta} = q$. Now let $\boldsymbol{\sigma}$ be the sentence

$$\forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{S}^q\mathbf{0}, \boldsymbol{S}^q\mathbf{0}, \boldsymbol{v_3}) \to \boldsymbol{\beta}\ (\boldsymbol{v_3})]$$

which is $\boldsymbol{\delta}(\boldsymbol{S}^q\mathbf{0})$. So,

$$\models_{\mathfrak{N}} \boldsymbol{\sigma} \text{ iff } \models_{\mathfrak{N}} \boldsymbol{\delta}(\boldsymbol{S}^q\mathbf{0}) \text{ iff}$$

$$\models_{\mathfrak{N}} \boldsymbol{\delta}(\boldsymbol{S}^{\#\boldsymbol{\delta}}\mathbf{0}) \text{ (since } \#\boldsymbol{\delta} = q) \text{ iff}$$

$\#\boldsymbol{\delta}$ is in the set determined by $\boldsymbol{\delta}$. By the above discussion, this statement is so if and only if $\#(\boldsymbol{\delta}(\boldsymbol{S}^{\#\boldsymbol{\delta}}\mathbf{0})$ is in the set defined by $\boldsymbol{\beta}$, or equivalently if

$\#(\boldsymbol{\delta}(\boldsymbol{S}^q\mathbf{0}))$ is in the set defined by $\boldsymbol{\beta}$. Since $\boldsymbol{\delta}(\boldsymbol{S}^q\mathbf{0}) = \boldsymbol{\sigma}$, our last statement is equivalent to saying that $\#\boldsymbol{\sigma}$ is in the set defined by $\boldsymbol{\beta}$. Hence, $\models_{\mathfrak{N}} \boldsymbol{\sigma}$ if and only if $\#\boldsymbol{\sigma}$ is in the set defined by $\boldsymbol{\beta}$. This discussion gives us a notion of what $\boldsymbol{\sigma}$ asserts in $\mathfrak{N}$, namely, $\boldsymbol{\sigma}$ asserts that its own Gödel number is the set defined by $\boldsymbol{\beta}$. Equivalently, $\models_{\mathfrak{N}} [\boldsymbol{\sigma} \leftrightarrow \boldsymbol{\beta}(\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0})]$. However, we wish to demonstrate that

$$A \vdash [\boldsymbol{\sigma} \leftrightarrow \boldsymbol{\beta}(\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0})],$$

and what we have demonstrated thus far is that

$$\text{Th}\ \mathfrak{N} \vdash [\boldsymbol{\sigma} \leftrightarrow \boldsymbol{\beta}(\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0})].$$

Since $\boldsymbol{\theta}(\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3})$ functionally represents g, we have by the definition of functional representation

$$A \vdash \forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{S}^q\mathbf{0}, \boldsymbol{S}^q\mathbf{0}, \boldsymbol{v_3}) \leftrightarrow \boldsymbol{v_3} \approx \boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0}]$$

(note that $\#\boldsymbol{\sigma} = \#(\boldsymbol{\delta}(\boldsymbol{S}^q\mathbf{0}))$). Thus, we must also have

$$A \vdash \boldsymbol{\theta}(\boldsymbol{S}^q\mathbf{0}, \boldsymbol{S}^q\mathbf{0}, \boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0}).$$

It is clear that

$$\boldsymbol{\sigma} = \forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{S}^q\mathbf{0}, \boldsymbol{S}^q\mathbf{0}, \boldsymbol{v_3}) \rightarrow \boldsymbol{\beta}\ (\boldsymbol{v_3})] \vdash \boldsymbol{\theta}(\boldsymbol{S}^q\mathbf{0}, \boldsymbol{S}^q\mathbf{0}, \boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0}) \rightarrow \boldsymbol{\beta}\ (\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0})$$

by our substitution group of logical axioms. So, $A \cup \{\boldsymbol{\sigma}\} \vdash \boldsymbol{\beta}(\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0})$. By the Deduction theorem (Theorem 5.13), this last statement is so if and only if

$$A \vdash [\boldsymbol{\sigma} \rightarrow \boldsymbol{\beta}(\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0})] \qquad (*)$$

To show the other direction of the formal equivalence, we wish to demonstrate that

$$\{\forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{S}^q\mathbf{0}, \boldsymbol{S}^q\mathbf{0}, \boldsymbol{v_3}) \leftrightarrow \boldsymbol{v_3} \approx \boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0}], \boldsymbol{\beta}(\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0})\} \vdash \boldsymbol{\sigma}$$

(remember that $\boldsymbol{\sigma} = \forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{S^q0}, \boldsymbol{S^q0}, \boldsymbol{v_3}) \rightarrow \boldsymbol{\beta}\ (\boldsymbol{v_3})])$. We do this by assuming that we have a structure $\mathfrak{A}$ that satisfies the set

$$\{\forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{S^q0}, \boldsymbol{S^q0}, \boldsymbol{v_3}) \leftrightarrow \boldsymbol{v_3} \approx \boldsymbol{S^{\#\sigma}0}], \boldsymbol{\beta}(\boldsymbol{S^{\#\sigma}0})\}$$

and then showing that $\models_{\mathfrak{A}} \forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{S^q0}, \boldsymbol{S^q0}, \boldsymbol{v_3}) \rightarrow \boldsymbol{\beta}\ (\boldsymbol{v_3})]$. Our result will follow by the Completeness Theorem. Note that

$$\models_{\mathfrak{A}} \forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{S^q0}, \boldsymbol{S^q0}, \boldsymbol{v_3}) \rightarrow \boldsymbol{\beta}\ (\boldsymbol{v_3})] \text{ iff}$$

$$\text{for each fixed } u \in \mathfrak{A}(\forall) \models_{\mathfrak{A}} [\boldsymbol{\theta}(\boldsymbol{S^q0}, \boldsymbol{S^q0}, \boldsymbol{v_3}) \rightarrow \boldsymbol{\beta}\ (\boldsymbol{v_3})][[u]] \text{ iff}$$

$$\text{for each fixed } u \in \mathfrak{A}(\forall) \text{ either } \not\models_{\mathfrak{A}} \boldsymbol{\theta}(\boldsymbol{S^q0}, \boldsymbol{S^q0}, \boldsymbol{v_3})[[u]] \text{ or } \models_{\mathfrak{A}} \boldsymbol{\beta}(\boldsymbol{v_3})[[u]].$$

There are two possibilities; either $u = \mathfrak{A}(\boldsymbol{S^{\#\sigma}0})$ or $u \neq \mathfrak{A}(\boldsymbol{S^{\#\sigma}0})$. In the former case, $\models_{\mathfrak{A}} \boldsymbol{\theta}(\boldsymbol{S^q0}, \boldsymbol{S^q0}, \boldsymbol{v_3})[[u]]$ since

$$\models_{\mathfrak{A}} \forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{S^q0}, \boldsymbol{S^q0}, \boldsymbol{v_3}) \leftrightarrow \boldsymbol{v_3} \approx \boldsymbol{S^{\#\sigma}0}]$$

by assumption. However, in this case $\models_{\mathfrak{A}} \boldsymbol{\beta}(\boldsymbol{v_3})[[u]]$ by assumption. In the latter case

$\not\models_{\mathfrak{A}} \boldsymbol{\theta}(\boldsymbol{S^q0}, \boldsymbol{S^q0}, \boldsymbol{v_3})[[u]]$ by the same reasoning. Thus, we can say that for each fixed $u \in \mathfrak{A}(\forall)$ either

$$\not\models_{\mathfrak{A}} \boldsymbol{\theta}(\boldsymbol{S^q0}, \boldsymbol{S^q0}, \boldsymbol{v_3})[[u]] \text{ or } \models_{\mathfrak{A}} \boldsymbol{\beta}(\boldsymbol{v_3})[[u]].$$

Therefore,

$$\models_{\mathfrak{A}} \forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{S^q0}, \boldsymbol{S^q0}, \boldsymbol{v_3}) \rightarrow \boldsymbol{\beta}\ (\boldsymbol{v_3})],$$

and our result follows. Thus, since

$$A \vdash \forall \boldsymbol{v_3}[\boldsymbol{\theta}(\boldsymbol{S^q0}, \boldsymbol{S^q0}, \boldsymbol{v_3}) \leftrightarrow \boldsymbol{v_3} \approx \boldsymbol{S^{\#\sigma}0}],$$

$$A \cup \{\boldsymbol{\beta}(\boldsymbol{S^{\#\sigma}0})\} \vdash \boldsymbol{\sigma}.$$

By the Deduction Theorem,

$$A \vdash [\boldsymbol{\beta}(\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0}) \rightarrow \boldsymbol{\sigma}] \qquad (**)$$

Since we have both $(*)$ and $(**)$, We have

$$A \vdash [\boldsymbol{\sigma} \leftrightarrow \boldsymbol{\beta}(\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0})],$$

and the lemma is proven. ∎

Theorem 8.51 (Tarski Undefinability Theorem) *The set* $\#Th\ \mathfrak{N}$ *is not definable in* $\mathfrak{N}$.

This theorem will rely on the special argument by contradiction that we discussed in the overview of Gödel's Incompleteness Theorem.

Proof: Suppose by way of contradiction that there is a formula $\boldsymbol{\beta}$ that defines the set #Th $\mathfrak{N}$. Then $\boldsymbol{\beta}$ would only have one free variable since #Th $\mathfrak{N}$ will be a unary relation (a unary relation is the same as a set). We can assume without loss of generality that the variable $\boldsymbol{v_1}$ occurs free in $\boldsymbol{\beta}$ since if not we can find an alphabetic variant of $\boldsymbol{\beta}$ in which $\boldsymbol{v_1}$ does occur free by Theorem 5.22.

Applying the Fixed Point Lemma to the formula $(\neg\boldsymbol{\beta})$, which also only has the variable $\boldsymbol{v_1}$ occurring free, there is a sentence $\boldsymbol{\sigma}$ such that

$$A \vdash [\boldsymbol{\sigma} \leftrightarrow (\neg\boldsymbol{\beta})(\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0})].$$

It is clear that

$$(\neg\boldsymbol{\beta})(\boldsymbol{S}^{n}\mathbf{0}) = (\neg\boldsymbol{\beta}(\boldsymbol{S}^{n}\mathbf{0})),$$

so that we have

$$A \vdash [\boldsymbol{\sigma} \leftrightarrow (\neg\boldsymbol{\beta}(\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0}))].$$

Since $\mathfrak{N}$ is a model of A, we have

$$\models_{\mathfrak{N}} [\boldsymbol{\sigma} \leftrightarrow (\neg \boldsymbol{\beta}(\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0}))].$$

Thus,

$$\models_{\mathfrak{N}} \boldsymbol{\sigma} \text{ iff } \not\models_{\mathfrak{N}} \boldsymbol{\beta}(\boldsymbol{S}^{\#\boldsymbol{\sigma}}\mathbf{0}).$$

Since $\boldsymbol{\beta}$ is supposed to define #Th $\mathfrak{N}$, then our last statement is equivalent to

$$\models_{\mathfrak{N}} \boldsymbol{\sigma} \text{ iff } \#\boldsymbol{\sigma} \notin \#\text{Th } \mathfrak{N}.$$

So, $\boldsymbol{\sigma}$ asserts in $\mathfrak{N}$ that its own Gödel number is not in the set of all Gödel numbers for the sentences true in number theory. We can begin to see how $\boldsymbol{\sigma}$ is self contradictory. If $\boldsymbol{\sigma}$ is true in $\mathfrak{N}$, then by definition $\boldsymbol{\sigma} \in$ Th $\mathfrak{N}$, and hence $\#\boldsymbol{\sigma} \in \#$Th $\mathfrak{N}$, a contradiction. However, if $\boldsymbol{\sigma}$ is false in $\mathfrak{N}$, $\boldsymbol{\sigma} \notin$ Th $\mathfrak{N}$, and hence $\#\boldsymbol{\sigma} \notin \#$Th $\mathfrak{N}$. This is so if and only if $\models_{\mathfrak{N}} \boldsymbol{\sigma}$, that is, if and only if $\boldsymbol{\sigma}$ is true in $\mathfrak{N}$, a contradiction. Therefore, what we initially supposed to be true (that $\boldsymbol{\beta}$ defines the set #Th $\mathfrak{N}$) is false, and there is no such formula $\boldsymbol{\beta}$. Thus, the set #Th $\mathfrak{N}$ is undefinable in $\mathfrak{N}$. ∎

Corollary 8.51.1 *#Th $\mathfrak{N}$ is not recursive.*

Proof: By Corollary 8.49.1, if a unary relation (a set) is recursive, then it is definable in $\mathfrak{N}$. However $\#Th\mathfrak{N}$ is not definable in $\mathfrak{N}$, so neither can it be recursive. ∎

Remember that by Church's Thesis, recursion is the mathematically precise formulation of decidability. So, the above corollary asserts that number theory is not a decidable set, for if a sentence is true in $\mathfrak{N}$, its Gödel number will be in Th $\mathfrak{N}$ and vice versa. So, there is no effective procedure such that for every formal sentence $\boldsymbol{\sigma}$, the effective procedure will be able to decide whether

$\boldsymbol{\sigma}$ or its negation are in Th $\mathfrak{N}$. There is no computer program that, given *any* formal first-order sentence, will be able to definitely decide whether that sentence or its negation belongs in number theory.

And now, for the main result of this thesis.

Theorem 8.52 (Gödel's Incompleteness Theorem (Restated)) *If* $S \subseteq \textit{Th}\ \mathfrak{N}$ *and* $\#S$ *is recursive, then Cn* S *is not a complete theory.*

The reader should at this point re-examine the first statement of Gödel's Theorem to see how the concepts stated there relate to the concepts used in this restatement.

Proof: Since $S \subseteq \text{Th}\ \mathfrak{N}$, then Cn $S \subseteq \text{Th}\ \mathfrak{N}$ by the definitions of each of these theories. Suppose by way of contradiction that Cn S is a complete theory. By Theorem 7.21 (since Th $\mathfrak{N}$ is satisfied by the structure $\mathfrak{N}$), we have Cn $S = \text{Th}\ \mathfrak{N}$. However, if Cn S is complete, then by Theorem 8.50, $\#\text{Cn}\ S$ is recursive. This statement would say that $\#\text{Th}\ \mathfrak{N}$ is recursive, which by the corallary to the Tarski Undefinability Theorem is not the case. Therefore Cn S cannot be complete. ∎

Kurt Gödel proved a version of the incompleteness theorem (different from the one stated above) in 1931 a year after he proved a version of the Completeness Theorem for first-order logic for his doctoral dissertation in 1930. By Church's Thesis, recursiveness is equivalent to decidability. So, $\#S$ being recursive means that it is decidable. Clearly, if $\#S$ is decidable then S itself will also be decidable since we can decode all of the Gödel numbers back into the formulas in S. So, if we have a decidable set of sentences true in number theory, a set of sentences where a computer program could say "yes" or "no" to each formal sentence being in the set, the set of all consequences from the set cannot completely describe all true statements of number theory. There will always be a sentence true about numbers that cannot be proven (deduced) from such a decidable set.

Recall the definition of an axiomatizable theory.

Definition 8.53 *A theory T is* **axiomatizable** *if there is a decidable set of sentences Σ such that $T = Cn\ \Sigma$.*

By Gödel's Incompleteness Theorem, there can be no first-order axiomatization of number theory. For a set of axioms, we as reasoners *decide* what axioms to include. There is no set of *first-order* axioms that we can develop that can completely describe number theory. That is, there is no set of first-order axioms such that we could ever prove *every* statement true of number theory.

Consider again, Goldbach's Conjecture: every even integer greater than 2 is the sum of two primes. This statement of number theory is unproven to date using a "normal" set of first-order axioms. Given a normal set of first-order axioms one might assume, we know by Gödel's theorem that there are true statements of number theory that are not provable from this assumed set of axioms. Goldbach's Conjecture could very well be one of these unprovable statements. Or, it could be that Goldbach's Conjecture is false in number theory, and we have as yet to find a counterexample. We do not know whether it is true or false because no one has ever supplied a proof for it. There is great *evidence* that Goldbach's Conjecture is true, and so we could certainly add Goldbach's Conjecture to our set of axioms. It is then, of course, it is provable. But since we do not *know* whether it is in fact true in number theory, our set of axioms could be inconsistent. Given the modern axiomatic mathematical method, only if we as reasoners stumble upon a proof or counterexample (a first-order proof/counterexample or otherwise), will we be able to determine whether Goldbach's Conjecture is in fact true of number theory or not. The bait that keeps mathematicians fishing for a proof is the tantalizing tautology that it *must* be either true or false in first-order number theory (a complete theory).

Having proved Gödel's Incompleteness Theorem as the main goal of this thesis, we briefly and incompletely consider some of the philosophical consequences of the theorem in the next (and concluding) chapter.

Chapter 9

Conclusion

Almost since Gödel's announcement of his incompleteness theorem in Königsberg in 1930, the theorem has been used (and abused) to assert astounding things. It is not difficult to see why. The theorem has direct bearing on the philosophy of mathematics and goes right to the heart of meta-mathematical questions. How can we know mathematical truth? Is mathematics realism or formalism? How do we know whether proofs exist or do not exist? Can we prove every mathematically true statement? All of these questions are at the heart of the project of mathematics.

Gödel's Theorem stretches even beyond mathematics to metaphysics and deeper human questions. How do we know what we know? What is the human mind? Is there objective reality? Although, a complete philosophical discussion of the implications of Gödel's Theorem is beyond the scope of this thesis, we talk briefly about some (possible) derivatives and non-derivatives of the theorem.

Note in the discussion at the end of the last chapter the careful insertion of the term *first-order*. Gödel's Incompleteness Theorem is a meta-theorem about the formal mathematical structure called "first-order logic." In this thesis, we developed both sentential logic and first-order logic as mathematical models of humanity's deductive thought processes. Mathematical models are powerful because of their ability to compactly describe and indicate structure in the real world. For example, quadratic functions are useful to model the physical path of projectiles, and using reasoned results about quadratic functions can give us nice indications about what should happen in the real world. Thus, if I throw a ball, I can mathematically model its height t seconds after I throw it with a quadratic function $h(t)$. Given what I know about quadratic functions, I could make predictions about how high the ball will travel and at what time into its flight it will attain this height. I can predict when it will hit the ground again or reach any particular height.

However, if I go and actually measure the quantities after my predictions, I am liable to be close but not exactly in line with my predictions. Why is this? Because my model makes simplifying assumptions. Reality is complex. Perhaps I neglected the effect of air resistance when developing my quadratic function. My model is only as good as its underlying assumptions. If I include the effect of air resistance, my model will be more complex and more adequate in its reflection of the structure of the real world, but by no means will it be perfect.

This situation is analogous to mathematically modelling logic itself. Our first model was sentential logic, and we were able to prove some nice results such as Compactness. However, we saw the inadequacy of sentential logic to express properties of sets. We then developed first-order logic as a more complex model of logic, able to approximate our deductive thought processes more adequately. With a more expressive model we were able to prove more powerful results such as first-order Soundness (a proved statement will always be true), Completeness (any statement that is true given a set of assumptions can be proved from that set of assumptions), and Compactness (a statement provable from a set of assumptions will always be provable from a finite set of assumptions). We were also able to prove Gödel's Incompleteness Theorem, which is a statement about our first-order model of deduction (we were also able to indicate some incompleteness results for sentential logic as well (see Section 3.3)).

Now, as any good, consistent, mathematical model will do, Gödel's Incompleteness Theorem gives us some indication of what is going on with the reality of logic. However, we must be careful in drawing conclusions beyond what the theorem actually states. The theorem is very specific in that it is a statement about *first-order* logic and in that it deals with a specific theory (number-theory). (Gödel's Theorem does also have some corollaries that have implications beyond number theory, but these are beyond the scope of this thesis.) So, the conclusions that we draw from Gödel's Theorem will be accurate in their predictions insofar as they follow the statement of the theorem with all of its inherent assumptions. Let us start with a very bad application of Gödel's Theorem.

"Godel's Incompleteness Theorem demonstrates that it is impossible for the Bible to be both true and complete."

The author searched on the internet via the Google search engine using the search phrase "Godel's incompleteness theorem the bible" and pulled this phrase from the first result page, [6]. The intent was to pull the phrase from the website not to either support or rebut the argument present on that page. There is no intent to create a strawman argument.

First, Gödel's Theorem deals with a formal language that must be able so support the axioms of A. The Bible does not use such a formal language nor is its intent to have its language support the structure of number theory. Gödel's theorem talks about the truth about statements of number theory, and the incompleteness of certain subtheories of number theory. However, the above statement appears to talk about the truth of the assertions that the Bible does make and to the Bible having a complete description of reality in some context. The primary assertions of the Bible are not statements about number theory at all. Gödel's Theorem simply does not apply. There are numerous examples of such blanket and erroneous applications of Gödel's Theorem (see [3]).

As we have seen, Gödel's Theorem does not assert the incompleteness of everything about which statements of completeness and incompleteness can be made. It states the incompleteness of a *particular* mathematical theory. Even the term "completeness" has a very specific defined mathematical meaning. As, we have seen, there are examples of complete theories (such as the theory of algebraically closed fields of characteristic zero).

Now, for a few statements to which Gödel's Theorem could be reasonably be expected to apply.

"Gödel's Incompleteness Theorem indicates that there are first-order truths about number theory that computers will never be able to demonstrate to be true."

It is reasonable to apply Gödel's Theorem in this case. In fact, the person who took the most notice of Gödel's announcement at the 1930 Königsberg conference was John von Neumann, one of the fathers of computer science (see [4] for history; the internal architecture of most computers is known as Von

Neumann architecture). Fundamentally, computers operate on binary arithmetic, strings of 1's and 0's (or for us strings of 0's and S(0)'s). A computer programming language could definitely be considered a formal language that when compiled gets translated in the actual structure of number theory (the natural numbers are represented in binary notation). It is fairly clear that computers can be built that can support the structure of A, and in fact the logic that computers use is first-order logic. It is then completely valid to apply Gödel's Theorem to computability theory. Now, the arithmetic that the computer is programmed to use will of course involve only finitely many axioms. So, given the discussion at the end of the last chapter, the subtheory of the number theory that the computer is capable of expressing given the axioms programmed into it must be incomplete. That is, there are bona fide statements of number theory that can be expressed in the computer programming (formal language) such that these statements will be true in number theory but that the computer cannot determine whether they are true or false in the subtheory of number theory that it is operating under. The natural extension to this claim is by replacing the word "computers" with "human minds."

"Gödel's Incompleteness Theorem indicates that there are first-order truths about number theory that human minds will never be able to demonstrate to be true."

Now, the application of Gödel's Theorem is less clear because it is less clear what the human mind is as opposed to a computer. There is no doubt that our minds are adequate to support the structure of A and that if we are operating under a certain set of assumed axioms for number theory, the subtheory of number theory build off of those axioms will be fundamentally incomplete in a *first-order* logical setting, meaning that there will be first-order logical statements that will be true of number theory that human minds can never prove *with first-order logic.* However, note the careful insertion of the word "first-order." Gödel's Incompleteness Theorem is fundamentally a statement about the mathematical model of first-order logic. Unlike a computer, however, the mind is capable of logic beyond the first-order.

Refer back to Example 4.39. There, we discussed how every first-order statement of ordering that can be made about the rational numbers can be made about the real numbers and vice versa. The ordering structures for $\mathbb{Q}$ and $\mathbb{R}$ are elementary equivalent. However, in terms of ordering, the Completeness Axiom for the real numbers (every bounded above set has a least upper bound) describes the essential difference between the real numbers and the rational numbers. However, as noted in Example 4.39, this statement is a second-order logical statement. In second-order logic, we have the same symbols as in the first-order alphabet, but now we allow variable predicate and function symbols (Enderton, pp.268-269). This difference between first-order and second-order languages will allow for the ability to range not only over elements within a set, but also to range over sets themselves. So, we need a second-order language to express the Completeness Axiom for the real numbers. The point of this example is that we as reasoners have more at our disposal than just first-order logic. We can demonstrate a fundamental difference between the ordering structures for $\mathbb{Q}$ and $\mathbb{R}$ using homomorphisms. So, even though we as mathematicians working within the axiomatic method cannot hope to prove, using first-order logic, every statement true of number-theory by Gödel's Incompleteness Theorem, we do have more than first-order logic at our disposal to attempt to prove such statements, the proof then being a deduction of a higher-order logic rather than a deduction of first-order logic.

An example may clarify more fully what we mean.

Take a polynomial with integer coefficients. We know, given the Fundamental Theorem of Algebra, that this polynomial will have all of its roots in $\mathbb{C}$. The idea of this polynomial having a root can be expressed as a first-order statement. However, knowing that this polynomial will have all of its roots in $\mathbb{C}$ comes from the Fundamental Theorem of Algebra, whose proof relies on results of Complex Analysis, which relies on $\mathbb{C}$ being a complete (in the sense of the Completeness Axiom) space, which is a second-order concept. So, there are things that we can know at a first-order level which only come from our knowledge (coming from a proof) at a second-order level.

Does Gödel's Incompleteness Theorem give a statement about our inability as human reasoners to prove things? In a first-order logical way, but not in a logical way in general. It would indeed be interesting to see if there is an analogous second-order Incompleteness Theorem and would be an avenue for further research. One thing is clear from our discussion however. Given our current understanding of what a computer is, the ability of our minds to reason is greater than that of a computer. As many have indicated, Gödel's Incompleteness Theorem indicates that our minds are more than a computer.

Even though Gödel's Incompleteness Theorem does spark many interesting philosophical notions, these cannot be fully explored in this thesis. It will no doubt be discussed for generations to come even as the average working mathematician labors in his or her particular field completely unphased by Gödel's Incompleteness Theorem in his or her establishment of mathematical results. After all, Gödel's Theorem does not destroy the possibility of proofs and may even encourage us to continue to seek for mathematical truth that exists apart from our current set of assumed axioms.

Bibliography

[1] Enderton, Herbert B. *A Mathematical Introduction to Logic.* Academic Press Inc, San Diego, 1972

[2] Fraleigh, John B. *A First Course in Abstract Algebra* 3d, Addison-Wesley Publishing Company, Philippines, 1982

[3] Franzén, Torkel. *Gödel's Theorem–An Incomplete Guide to its Use and Abuse* A K Peters, Ltd, Wellesley, 2005

[4] Goldstein, Rebecca. *Incompleteness–The Proof and Paradox of Kurt Gödel* W. W. Norton & Company, Inc, New York, 2005

[5] Ross, Kenneth A. *Elementary Analysis: The Theory of Calculus* Springer-Verlag New York, Inc, 1980

[6] "35 Godel's Incompleteness Theorem (Atheism FAQ)." `www.stason.org/TULARC/religion/atheism/35-Godel-s-Incompleteness-Theorem-Atheism-FAQ.html#.Ubr6aZxc4_g`, Accessed 6/14/2013

VITA

Author: Christopher Mullins

Place of Birth: Spokane, Washington

Undergraduate Schools Attended:

Spokane Falls Community College

Gonzaga University

Degrees Awarded:

Associate of Arts, 2008,
Spokane Falls Community College

Bachelor of Science in Mathematics, 2011,
Gonzaga University

Honors and Awards:

Outstanding Graduate Student of MS Mathematics, 2013

Graduated Summa Cum Laude, Gonzaga University, 2011

Carsrud Award for Outstanding Senior Mathematics Student,
Gonzaga University, 2011

Inland Northwest Car Club Scholarship, 2010, 2011

Alumni Scholarship, Gonzaga University, 2009

Trustee Scholarship, Gonzaga University, 2008