A Philosophy Student's Introduction to Metalogic

for Advanced Undergraduate and Beginning Graduate Students

John N. Martin

Department of Philosophy
University of Cincinnati

john.martin@uc.edu

Revised March 9, 2012

Table of Contents

Introduction

A. Topics

This text is an introduction to logical theory for advanced undergraduate and beginning graduate students in philosophy. Like most texts at this level it centers on the metatheory of first-order logic. The treatment includes the standard Gentzen natural deduction system, Tarski-style model theoretic semantics, and a Henkin-style completeness proof. The text, however, covers a selection of other topics as well. These were chosen with the needs of philosophy graduate students in mind, especially those planning to work in areas of philosophy that concern language, the natural sciences, and philosophical psychology.

Regardless of specialty, all philosophy students should know the standard theory of first-order logic, the *lingua franca* of technical research today. Here one learns about the difference between syntactic and semantic ideas, inductive definitions and proofs, models, validity, completeness, and constructive proofs. This core material is presented as a unit in Chapter 2. Students who will work with computation theory, which is especially common in the philosophies of language and mind, require a good grasp of effective process. The topic is covered twice, first historically in Chapter 1 as part Gödel's incompleteness proof, and again more systematically in Chapter 3. Covered there are the definition of *effective process,* Church's thesis, logic programming with Prolog as an example, and first-order undecidablity proven by Herbrand's methods. Special care is given to explaining Prolog in the language of first-order logic and for motivating the resolution proof technique within Herbrand's model theory. Chapter 4 introduces students to three areas of logic with broad application in philosophy: many-valued, modal, and intensional logic. Here philosophical issues are discussed and general methods introduced for assessing the logical merits of standard alternatives.

B. The History of Logic

It is the author's opinion that students cannot understand core ideas in modern logic, especially proof theory, model theory, and computation, without knowing something about their historical development. Accordingly the text begins with an account of the emergence of formal logic in the nineteenth and early 20th centuries. Chapter 1 recounts the process as stages: non-Euclidean geometry, the formal axiomatizations of mathematical theories like Peano's arithmetic, Frege's *Grundgesetze*, logicism, Russell's paradox, and Gödel's incompleteness proof. The point of the review is to bring students themselves to the mind-set of logicians in the 1930's when logical positivism gripped the philosophical world and advanced logic was about to bloom. Not only will they see the motivation for the concepts like natural deduction, model theory, and non-classical logics, which are studied systematically in later chapters, they will sense the point of branches of logic not covered in this book, like alternative philosophies of mathematics, higher-order logic, and axiomatic set theory. When no conceptual issue is at stake, the history is simplified by replacing awkward early formulations by clearer versions discovered later. The original axioms for propositional, first-order logic and type theory, for example, are streamlined, and Gödel's proof is simplified by skipping over various definitions of calculable functions and by using Tarski's theorem. To aid students in reading the original sources, however, the text retains traditional notation and definitions when permitted by modern conventions.

C. Method

The text is designed as much to teach methodology as ideas, and the emphasis on method greatly affects the character of this text. The basic methodological goal is to bring advanced students to the point where they can *read* logic for themselves. Reading papers and books on philosophical logic is an acquired skill. To further logical literacy, the text is divided into two styles, a more formal mathematical presentation similar to that found in professional logic papers, and a more chatty informal style appropriate for teaching. The formal texts are separated-off in displayed boxes. These are intended to be sources of perplexity to the student, specimen documents to be taken as objects of study and deciphered. They are surrounded by informal text written with the purpose of helping in the task. The official logical theory, then, is in the boxes. There it is rigorous, short and expressed in the peculiar jargon of the field. Students who come to read these mini-documents with understanding will have learned a good deal of logic, having acquired some of the mathematical sophistication needed for reading other formal texts. The formal style is also written at a level that students may take as a model to imitate as they go on to write about formal ideas in their own work.

Method is addressed in several more specific ways. One is the use of definitions and deductive reasoning. In one way or another almost all formal work consists of deductions from definitions and assumptions. Reading logic is largely reading proofs and definitions. It is fair to say that the sort of proofs philosophy students will encounter when reading, or will write themselves, is itself a kind of code. It is written in mathematical English (or similar natural language) that, as professionals understand, may be translated if necessary into more formal statements in first-order logic and set theory. The definitions and proofs actually written down then are really recipes for recreating longer more complete versions of the text in the notation of formal logic and mathematics. Reading the superficial form of logicalese this way takes practice. Accordingly formal definitions displayed in the text are explicitly stated in naïve set theory and proofs are spelled out in the steps of first-order logic. The level of detail is somewhat more than usually found in professional papers but still succinct enough to be challenging. Definitions and proofs become briefer and more formal as the text progresses.

A major methodological feature of the text is the use of abstract algebra. It is the author's experience that philosophers often do not understand the point of logic, or other formal work, because they do not understand what it is to study structure for its own sake. But mathematical work is essentially the study of structures – some would extend the claim to natural science generally. It is best to make this orientation clear to students from the outset, and the best way to do so is to formulate issues using algebraic ideas. Accordingly notions of set, relation, function, structure, and morphism are introduced gradually in the early chapters, and become the working idiom in Chapter 4. By the book's end a philosophy student will have a good idea of what it is to study structure.

Perhaps the most important lesson on method for students who will pursue logic itself is to learn to critically evaluate competing metatheories. Since this text is introductory, such comparison is limited to several standard controversies. Chapter 1 recounts the failure of logicism. In Chapter 2 axiomatic methods are contrasted with natural deduction, and the intuitionistic criticisms of classical logic and set theory are set out. Chapter 3 reviews differences between definitions by abstraction and induction. Chapter 4 is most methodological in this sense. Families of many-valued, modal, and intensional logics are defined in global terms, and tools developed for comparing their logical properties.

D. Exercises

Unlike some logic texts in which exercises develop examples relevant to mathematics or advance the book's content by having the students draw out additional technical results, the exercises here are designed with the philosophy student in mind. They attempt to insure that students posses sufficient technical skills to work through the material, but emphasize a discursive understanding of concepts. The problems following each chapter are divided into three sets. The first (marked **Skills**) gives pencil and paper practice at the sort of derivations and symbol manipulations needed to work through proofs. The second (marked **Ideas**) asks probing questions designed to insure an understanding of technical ideas. The third (marked **Theory**) poses global questions that allow the student to make a record in his or her own words of the general strategy of major proofs and of the theoretical issues they address. Each set concludes with a section (marked **Method**) in which students are alerted to methodological points to be watched for in the chapter. Students are advised to read the exercises first, keeping them in mind while working through the chapter's text.

The author wishes to tank the students who have helped improve the text with recommendations and corrections, especially Jennifer Seitzer and Viorel Paslaru.

Chapter 1

The Beginning of First Order Logic

I. 19TH CENTURY AXIOMATICS AND LOGICISM

A. *A Priori* Knowledge and Axiom Systems Prior to 1800

It is a curiosity of history that when the rest of what we think of as modern -- literature, art, political institutions, and above all the empirical sciences like physics and biology -- were being born in the Renaissance and Enlightenment, logic was stagnating and even regressing. Especially formal and technical logic was seen by the scholars of the period as a dubious legacy from mediaeval scholasticism. It was kept alive in courses at the universities, but the universities themselves were in decline. They too were holdovers of the Middle Ages, fixed in their ways and conservative, still teaching in Latin and insisting on examinations in the format of the mediaeval disputation. No longer were university teachers the leaders of intellectual invention or progress. In the Renaissance the role of innovator passed to scholars outside the universities who were wealthy or who could obtain the patronage or the rich and powerful. Later, in the Enlightenment, universities were further eclipsed as research institutions by the various academies in letters and science, like the Académie des Sciences in France and the Royal Academy in England, which were established to support individual researchers largely outside university faculties.

The logic that was still part of the university curriculum consisted of superficial summaries of the simpler parts of Aristotle's syllogistic. It consisted of the simple theory of the syllogism that we meet in Lecture 3, and which became known as "school logic." This sort of logic was still being taught in American universities well into this century. Forgotten were the lively controversies connected with its discovery in the Middle Ages, forgotten were its extensions -- which we have not studied -- into the logic of necessity, possibility, and time, called *modal logic*, pioneered by Aristotle himself and pursued in the Middle Ages. Forgotten was sentential logic, first discovered by the ancient Stoics, rediscovered, and elaborated in the Middle Ages. Forgotten was the sophisticated mediaeval research into grammar and semantics. There was some original work done outside the universities,[1] but on the whole, the era was a

[1] It is true that the decline of logic was gradual and that some interesting work in formal logic was done in the 14th and 15th centuries. It is also true that the Renaissance saw the reawakening of interest in Aristotle's informal logic or "topics," practical rules of thumb to aid reasoners engaged in day-to-day debate or scientific inquiry. Neither the work in formal or informal logic however was of high quality. See E.J. Ashworth, "Traditional Logic," and Lisa Jardine, "Humanistic Logic," in Charles B. Schmitt et al., *The Cambridge History of Renaissance Philosophy* (Cambridge: Cambraidge Universsity Press, 1988). Two exceptions to the general rule are found in the work of the Rationalists. The Port Royal logicians developed a grammatical and semantic theory that anticipated some of Noam Chomsky's ideas in linguistics (see Noam Chomshy*, Cartesian Lingusitcs*), and Leibniz anticipated modern symbolic logic by inventing various symbolic

logical dark age. With little personal understanding of the subject, a consensus developed among the learned about what logic was and how it fitted into the rest of science. It was seen as an uncontroversial subject, and one that was completely understood. In the 18th century the great German philosopher Immanuel Kant described logic, which he identified with the simple syllogistic, as a science that was "achieved and complete." Though complete, formal logic was also thought to have only very limited practical application to the world of affairs or natural science. It was of interest only to philosophers who, though uninterested in doing original work in logic, nevertheless needed to fit this curious branch of knowledge into the grand scheme of things.

Though philosophers from 1450-1850 really knew less about logic then those of previous periods, when they did try to fit it in, they attributed to logic a rather exalted status. It was well known, even in ancient times, that there was something particularly obvious or transparent about logic, but until relatively recently philosophers did not identify this transparency with logic in particular. Until well into the 17th century, they tended to think of all knowledge, or at least that associated with the sciences, as certain. This innocence was lost once the natural sciences like physics and chemistry were seriously underway, and it became very clear how much work they involved. It proved very difficult to amass the needed information and make the right generalizations. In comparison, logic, together with its cousin mathematics, stood out as areas of learning that were safe and certain.

They asked why. Why were logic and mathematics special? Kant and earlier philosophers of the Enlightenment offered an explanation by making a distinction. There are, they said, two kinds of knowledge. The more common type is that gained through the senses. This includes knowledge of day to day facts about the world, as well as the generalizations about experience that make up the natural sciences. Its technical name is ***empirical*** or ***a posteriori*** knowledge. Though empirical knowledge is extremely useful, it is hard to accumulate. It requires observations, data collecting, and generalizations, all methods that are time consuming, costly, and prone to mistakes.

The second, rarer sort of knowledge is that associated with logic and mathematics. It is the opposite of empirical knowledge; that is, it is knowledge *not* based on experience, and it is called ***non-empirical*** or ***a priori*** knowledge. This is the sort of knowledge obtainable just by thinking. You can make its discoveries in your armchair, with your eyes closed. At one point in the 17th century when modern physics was just unfolding, rationalism, the school of philosophy then dominant, optimistically proclaimed that all knowledge could be obtained by thought alone. It quickly became evident, however, that the optimism was unfounded, and that relatively little knowledge is obtainable just through thinking.

What little there is, however, is important. Logic and mathematics appear to fall within this class. Such knowledge does seem to have a special feature: if it is true, it is not subject to the same sort of doubts as empirical knowledge.

languages for reasoning that featured both a clear semantics and techiniques for generating syntactic proofs (see C.I. Lewis, *Survey of Symbolic Logic*).

Indeed, it is only about matters of pure reason that we seem to be able to achieve certainty.

But what is it that is special about mathematics and logic, making it obvious and obtainable through reason alone? The answer is as old as philosophy. Logic and mathematics are based on *proofs*. Among all true propositions, there are some that are basic and so obviously true that they are self-evident. These are set down as axioms, and from them small but completely reliable steps of reasoning derive other truths. A chain of such reasoning can take you from an axiom that is obviously true to a rather remote theorem, which might not be evident at all. However, since the premises of the demonstration are certain, and each logical step is transparently correct, it follows that we know the conclusion too with certainty. Aristotle calls the truths derived this way ***apodictic***, which is usually translated into English as ***provable*** or ***demonstrative****.*

Probably the most consequential book every written for the scientific method was Euclid's *Elements*. These are thirteen books, complied from the work of Greek mathematicians in the generation of Aristotle's pupils, which lay out the theory of plane geometry in axiomatic form. This was the first treatise to use the axiomatic method. Your high-school geometry text was probably modeled on the *Elements*. Many of you will remember that it began by laying down three sorts of assumptions: ***definitions***[2] of basic terms, ***axioms*** stating self-evident truths of logic, and ***postulates*** containing "self-evident" geometric truths. It is the postulates that will be of interest to us here. Euclid employs just five:

Euclid's Postulates for Plane Geometry (*Elements*, 300 B.C.)

1. Any two points are contained in some line.
2. Any finite line is contained in some line not contained in any other line.
3. Any point and any line segment beginning with that point determine a circle with the point as its center and the line as its radius.
4. All right angles are equal.
5. (**Euclid's original version**.) If a straight line falling on two straight lines makes the interior angles on the same side less than two right angles, then the two straight lines, if produced indefinitely, meet on that side on which the angles are less than the two right angles.

He then proceeds to deduce the theorems of the subject by giving a proof for each. A proof consists of a series of a special sort: each step is either a theorem already established or follows from a previous step in the series by a self-evident application of logic. It is important to stress that from ancient times Euclid's

[2] In our discussion here we will need to employ only three of the terms Euclid defines: right angle, perpendicular line and parallel line. His none too clear explanations are:

- When a straight line set up on a straight line makes adjacent angles equal to one another, each of the equal angles is ***right*** and the strainght line standing on the other is called ***perpendicular*** to that on which it stands (definition 10).
- ***Parallel*** straight lines are straigth lines which, being in the same plane a being produced indefinitely in both directions, do not meet one another in either direction (definition 23).

results were viewed as certain. The definitions, axioms, and postulates were viewed as certain, and so were the steps of reasoning contained in his proofs. It follows that every provable theorem is established with certainty.

Geometry's axiomatic method captured the imagination of philosophers and scientists. It was viewed as a paradigm of good scientific method. Indeed, since Euclid the axiom system has been the preferred format for the presentation of mathematical results. When possible it has been applied in philosophy and the natural sciences. Let me give you two examples, one each from philosophy and physics.

The first is the axiom set laid down by Spinoza, a rationalist philosopher of the seventeenth century, in his treatise the *Ethics.* From the following seven principles he attempts to deduce all the important truths of philosophy, both natural and moral.

Spinoza (*Ethics*, 1670)

1. Everything which is, is either in itself or in another.
2. That which cannot be conceived through another must be conceived through itself.
3. From a given determinate cause an effect necessarily follows; and, on the other hand, if no determinate cause is given, it is impossible that an effect can follow.
4. The knowledge of an effect depends upon and involves the knowledge of the cause.
5. Those things that have nothing mutually in common with one another cannot through one another be mutually understood, that is to say, the conception of the one does not involve the conception of the other.
6. A true idea must agree with that which is the idea.
7. The essence of that thing which can be conceived as not existing does not involve existence.

The next example is the axiom set used by Isaac Newton in *Principia Mathematica*, the classical statement of the science of mechanics. These are the famous three Laws of Motion from which he deduces the body of theorems that constitute the truths of the subject. You may well have studied these laws in a non-axiomatic form in high school physics. [3]

[3] These laws are in Newton's original formulations. You may recall them as (1) a body at rest tends to remain at rest, and a body in motion tends to remain in motion, until acted upon by an external force, (2) f = ma, and (3) for every action, there is an equal and opposite reaction.

Newton's Three Laws of Motion (*Mathematical Principles of Natural Philosophy*, 1686)

1. Every body continues in its state of rest, or of uniform motion in a right line, unless it is compelled to change that state by forces impressed upon it.
2. The change of motion is proportional to the motive force impressed; and is made in the direction of the right line in which that force is impressed.
3. To every action there is always an equal reaction: or, the mutual actions of two bodies upon each other are always equal, and directed to contrary parts.

We have here a pretty picture of natural science. Some propositions are empirical and cannot be known with certainty, but among these some are fundamental and have the status of *laws*. We may adopt them as postulates in an axiom system and deduce from them by logical rigor an entire branch of empirical science. The whole is then as secure as its basic postulates. An example of such a natural science is Newton's mechanics.

Natural sciences are to be contrasted with logic and pure mathematics. In the latter some propositions are known *a priori* and are certain. These may be laid down as axioms, and an *a priori* science then deduced from them with the aid of logic. An example is supposed to be Euclid's plane geometry.

In 1790, Kant advanced a systematic philosophy that gripped the intellectual world, especially Europe. He elaborated on the special status of logic and mathematics. There is, he says, a deep reason why math and logic are transparent to reason. Their laws express the very forms of thought. The underlying nature of reality dictates that when we perceive something we organize the world in accordance with the rules of logic and mathematics. The details of his system are complex and subtle, but what concern us here today is the sort of logic and math he thought was the key to organizing reality. Kant was no logician or mathematician, and he adopts the non--specialists viewpoint. By logic he means syllogisms and by math he means Euclid.

Among philosophers and mathematicians Kant's views on the nature of logic and mathematics were widely accepted and extremely influential. They came into conflict, however, with a niggling doubt that had been troubling specialists in geometry since ancient times. Working out this doubt in the face of Kantian dogma was to precipitate the crisis in 19th century mathematics that gave birth to modern logic.

Let us look again at Euclid's axioms. All were supposed to be self-evident. They were so regarded from the ancient world onwards. From the five, however, the fifth stands out. It is by far the most verbose. As such it is less likely to express an instantly obvious truth. Indeed, even in ancient times it was viewed as odd. In the fifth century, for example, Proclus thought it could be derived from the other four. His proof, however, fails because it mistakenly assumes the very postulate it is trying to prove.[4] But the postulate remained

[4] See Glenn R. Morrow, trans., *Proclus, A Commentary on the First Book of Euclid's Elements* (Princeton: Princeton Unversity Press, 1970), ll. 371.10-373.2, pp. liv-lv, 290-291.

troubling and attempts to put in on sounder footing continued without much success, until the early 19th century.

Let us consider for a moment how we might prove that the fifth postulate follows from the other four. One obvious strategy would be a reduction to the absurd. Assume together with the first four postulates the hypothesis that the fifth postulate is false. If we can then deduce a contradiction, we know that if the first four are true the fifth cannot be false, and is therefore true. Attempts to deduce a contradiction along this line, however, failed. This failure lead naturally to entertaining the opposite hypothesis, namely that the negation of the fifth postulate might be *consistent* with the first four postulates. But what would this mean?

It is clear how to prove *inconsistency*. Just deduce a contradiction. But how do you prove consistency? This is a question in logical theory, the sort of question that had been moribund for centuries. The mathematicians interested in the issue, however, proposed an answer. One way to show that an axiom system is consistent is to show it is *possible*, i.e. provide an interpretation or situation in which all the axioms are true together. After all, if they are *in*consistent, they are never simultaneously true. Thus if they are consistent, there ought to be some possible situation in which they are all true together.

B. Non-Euclidean Geometry[5]

In the 19th century it is was shown that in fact the fifth postulate does not follow from the other four. An important step is reformulating the fifth postulate in a shorter but fully equivalent manner. John Playfair (1748-1819) proposed a revised version (also know in antiquity) stated in terms of parallel lines.

Playfair's Version of Euclid's Postulate 5

Given a line and a point not on that line, there is exactly one line through that point parallel to the given line.

It was discovered independently by Karl Friedrich Gauss (1777-1855), Johann Bolyai (1802-1860), and Nikolai Lobachevski (1793-1856) that a consistent geometry would result by substituting for this postulate one inconsistent with it. The replacement specifies that through a point not on a line there is more than one parallel to it.

[5] There are introductions at all levels to non-Euclidean geomentry. For one that is non-technical and fun to read I suggest Philip J. Davis and Reuben Hersh, *The Mathematical Experience* (Boston: Houghton Mifflin, 1981). On the relation of non-Euclidean geometry to logic, I recommend Howard DeLong, *A Profile of Mathematical Logic* (Reading, MA: Addison-Wesley, 1970). On non-Euclidean geometry as a logistic system see Raymond L. Wilder*, Introduction to Foundations of Mathematics*, 2nd ed. (N.Y.: Wiley, 1967).

Postulate 5 in Lobachevskian Geometry

Given a line and a point not on that line, there are at least two distinct lines through that point parallel to the given line.

This system, not surprisingly, has some novel theorems. For example, the sum of the angles formed by a line bisecting two parallels is in general less than that of two right angles. This geometry, unlike Euclid's, also has the property that the measure of the least angle formed by the intersection of a parallel to the perpendicular of a line varies directly with the distance of the intersection from the line.

Soon after this discovery a third variety of geometry was discovered by Bernhard Rienmann (1826-1866) who observed that the angles of a triangle may be greater than that of two right angles and that a line and a point might well determine no parallel.

Postulate 5 in Riemannian Geometry

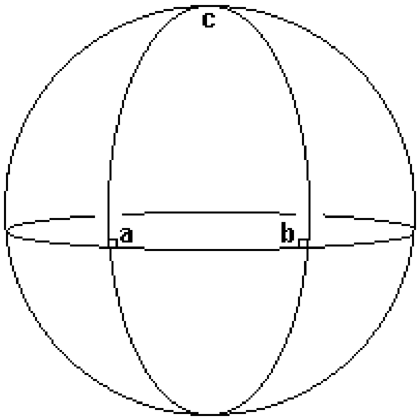
Given a line and a point not on that line, there is no line through that point parallel to the given line.

By way of exploring Rienmann's geometry let us see how it is an alternative to Euclid's. I said that both non-Euclidean versions of the fifth postulate were consistent with the first four postulates, and that the way you show this consistency is to construct a situation in which the first four postulates are true together with the new version of the fifth. Let us do so for Rienmann's version. Let us construct a model consisting of the points on the surface of a sphere, and let us identify a line as a great circles on the sphere's surface (i.e. a circle which centers on the center of the sphere). It is easy to see that the first four postulates are true. Any two points on the sphere's surface fall on some great circle, confirming postulate 1. Any finite arch of a great circle is contained in some great circle that is itself not a portion of another circle, satisfying postulate 2. Any arch of a great circle from a point on the surface determines a circle (this circle would not normally be a "line," i.e. a great circle) on the sphere with that arch as its radius, verifying postulate 3. Finally, all right angles are equal, postulate 4.

It is also true that Rienmann's postulate 5 is true. For consider any great circle on the sphere and any point off the circle. Now imagine a second a great circle passing through that point. This circle will interact the original circle, and hence is not parallel to it. Hence a "line" and a point determine no parallel.

It is also easy to see that the sum of the angles in a triangle is in general greater than that of two right angles. For example, consider the equator of the sphere given in the figure below. Consider in addition the sphere's "north pole" point **c**. Clearly **c** is not on the equator. Hence any great circle that passes through c will also intersect the equator. Indeed, any such circle will be a line of longitude of the sphere, forming a right angle with the equator. Now consider two points **a** and **b** on the equator, and the lines of longitude passing through them.

Notice also that **cab** and **cba** form right angles to the equator. Hence the sum of the angles of the triangle **cab** will be greater than that of two right angles.



By providing a model of the axioms, our discussion has proven the following result:

Theorem. Rienmannian Geometry is consistent.

The discussion also shows that Proclus and others were wrong in thinking that the first four postulates are sufficient for determining a Euclidean world. The fifth postulate is necessary as well.

It is hard to overstate the shock that resulted from the discovery of non-Euclidean geometry. No longer was geometry a paradigm case of *a priori* knowledge. Indeed the question of which geometry was the right one, i.e. which was true in the actual world, became an open question. Gauss and others started actually measuring the sum of the angles of large terrestrial triangles to see it they equaled 90 degrees. He found that within the margin of error of his measuring devices Euclid's geometry seemed to be confirmed. In the 20th century, however, it was a version of Rienmann's geometry that was incorporated into Einstein's theory of relativity. Geometry, in short, lost its status as *a priori* knowledge, and doubt was sown about the rest of mathematics. If geometry was not known *a priori*, then perhaps other branches of mathematics were not either.

C. Logistical (Axiom) Systems

One immediate consequence of non-standard geometry was a new interest in the properties of axiom systems. Mathematicians as a group became more sensitive to the difficulties of doing proofs. Beginning in the 19th century there was a widespread and quite significant elevation in the standard of rigor in mathematics generally. Proofs and definitions became clearer and more detailed, reaching a standard of precision that has been maintained to the present day. Although it had been assumed that most mathematical subjects were in principle axiomatizable, the details had never been worked out. In the mid-19th century programs started to axiomatize the various branches of the subject. Moreover, some mathematicians decided to make the steps of logical reasoning crystal clear. To do so they proposed introducing a special symbolic notation for the key non-mathematical words from English, German and other natural languages that were used to bind mathematical expressions into sentences and proofs. It took some decades for the symbolization to reach a standard form, but a consensus developed on the key ideas that needed to be symbolized and on the appropriate logical rules governing steps of reasoning.

In period from 1830 to 1930 the very notion of an axiom system and of its key properties became clearer and better defined. Like many important ideas in science, the ideas themselves are not difficult to grasp once they are formulated clearly, but arriving at their definitions nevertheless was the result of many years of puzzling over the nature of proofs.

Let us pause now to state the definitions, which are now standard. Our first goal is to define what it would be for a system, which we shall call ***S***, to be an axiom system. We begin by inventing a set of symbols to use in the notation of ***S***, and agree upon conventions or grammar rules for writing formulas, known as ***sentences***, in that notation. Let us use the $\mathbf{L}_{\boldsymbol{S}}$ to stand for the set of grammatical (well-formed) strings of symbols, called ***well-formed expressions*** of the system ***S***. Included in this set are the ***sentences*** of the system. Sometimes $\mathbf{L}_{\boldsymbol{S}}$ is called the ***language*** of ***S***.

We can now define the notion of an axiom system. It consists of a set of theorems deduced by logical rules from a set of axioms. In addition there is usually a set of definitions that allows for the abbreviation in more familiar terms of longer expressions in the syntax. In the twentieth century such systems are called ***logistic*** because they are logically explicit, making each step of reasoning perfectly clear.

Logistic Systems

An ***axiom*** (***logistic***) ***system S*** for sentences $\mathbf{L}_{\mathbf{S}}$ is **<$\mathbf{Ax}_{\mathbf{S}}$,$\mathbf{R}_{\mathbf{S}}$,$\mathbf{Th}_{\mathbf{S}}$,$\mathbf{D}_{\mathbf{S}}$,$\mathbf{DTh}_{\mathbf{S}}$> s.t.**

1. a set $\mathbf{Ax}_{\mathbf{S}}$ of sentences from $\mathbf{L}_{\mathbf{S}}$ called ***axioms***,
2. a set $\mathbf{R}_{\mathbf{S}}$ of logical rules on $\mathbf{L}_{\mathbf{S}}$ called ***rules of inference***,
3. the set $\mathbf{Th}_{\mathbf{S}}$ of ***theorems in primitive notation*** is defined inductively as follows:
   a. if P is in $\mathbf{Ax}_{\mathbf{S}}$, then P is in $\mathbf{Th}_{\mathbf{S}}$,
   b. if $P_1,...,P_n$ are in $\mathbf{Th}_{\mathbf{S}}$, and Q follows from $P_1,...,P_n$ by some rule in $\mathbf{R}_{\mathbf{S}}$, then Q is in $\mathbf{Th}_{\mathbf{S}}$
   c. nothing else is in $\mathbf{Th}_{\mathbf{S}}$;
4. a set of $\mathbf{D}_{\mathbf{S}}$ of ***abbreviative definitions*** of sentences of $\mathbf{L}_{\mathbf{S}}$ the form $t{=}t'$ or $P\leftrightarrow Q$**.**
5. the set $\mathbf{DTh}_{\mathbf{S}}$ of ***theorems in defined notation*** is defined inductively as follows:
   a. if P is in $\mathbf{Ax}_{\mathbf{S}}$ or $\mathbf{D}_{\mathbf{S}}$, then P is in $\mathbf{DTh}_{\mathbf{S}}$,
   b. if $P_1,...,P_n$ are in $\mathbf{DTh}_{\mathbf{S}}$, and Q follows from $P_1,...,P_n$ by some rule in $\mathbf{R}_{\mathbf{S}}$, then Q is in $\mathbf{DTh}_{\mathbf{S}}$
   c. nothing else is in $\mathbf{DTh}_{\mathbf{S}}$.

The definitions introduce new symbols and notation into the system, and allow its formulas to express ideas not obviously expressed by the symbols of the system's primitive notation used in $\mathbf{Ax}_{\mathbf{S}}$ and $\mathbf{R}_{\mathbf{S}}$. In this case the abbreviating expressions (those that appear $\mathbf{DTh}_{\mathbf{S}}$ but not in $\mathbf{Th}_{\mathbf{S}}$) are not really part of the language and then do not require semantic interpretation. For this reason these definitions are called both "abbreviative" and "eliminative."

When a definition functions as a kind of explaination what is being explained is the meaning of a defined expression that has an earlier history in and some sort of established meaning. Because the meaning of the new notation is intended to be "captured" by the definitions, the definition should provide an "intuitively acceptable" paraphrases of the defined expression that match thes earlier usage, its "standard use" outside and prior to the system. For example, It should match its use in earlier logic theory, the history of philosophy, or ordinary language.

One advantage of eliminative definitions in an axiom systems is that the defined ideas are in an important sense "explained" or "reduced to" the concepts that occur in the primitive notation of the axioms. The definitions provide explanations because all the system's theorems that are stated in "defined notation" follow logically as theorems from the axioms formulated in primitive notation. Thus, there is a sense in which the definitions in $\mathbf{D}_{\mathbf{S}}$ fuction as a set of supplementary axioms to thse in $\mathbf{Ax}_{\mathbf{S}}$ that set forth the meaning of the expressions they define. It was by means of definitions like these that Frege and Russell attempted to "reduce" arithmetic to logic. In their axiom systems matheimatical notation is introduced by eliminative definitions that are formulated using only logical symbols, and the truths of mathematics follow as theorems from axioms formulated purely in logical notation. When the defined terms are replaced by their *definienda* (when they are “cashed out’), they disappear and the theorems are stated solely in the primitive notation of the system’s axioms and rules.

Two additional points need to be made about eliminative definitions. The first is that the entire string of signs that is defined, the definitions *definiendum*, must be regarded strictly speakaing as a syntactic unit. Because it is merely an abbreviation for its *definiens*, any orthographic detail within the definiendum is purely accitdental. It there only as a guide to identifying its *definens*. For example, we shall shortly meet an eliminative definition of set abstract notation:

$$Q[\{x|P[x]\}] \quad \leftrightarrow_{\mathrm{def}} \quad \exists \mathrm{A}(\forall \mathrm{x}(\mathrm{x}{\in}A \leftrightarrow P[\mathrm{x}])\wedge\forall B(\forall \mathrm{x}(\mathrm{x}{\in}A \leftrightarrow P[\mathrm{x}])\rightarrow B{=}A)\wedge \mathrm{Q}[A]$$

Here $\{x|P[x]\}$ looks like a set name. It looks like the a singular term rougly equivalent to the English, “the set of all things of which $P(x)$ *is true.*” But that appearance is deceptive. In reality all the definition provides as an orthographic substitue, namely $Q[\{x|P[x]\}]$, that stands in place of the longer *definendum* $\exists \mathrm{A}(\forall \mathrm{x}(\mathrm{x}{\in}A \leftrightarrow P[\mathrm{x}])\wedge\forall B(\forall \mathrm{x}(\mathrm{x}{\in}A \leftrightarrow P[\mathrm{x}])\rightarrow B{=}A)\wedge Q[A]$. In this *definiendum* there is no singular term at all standing for that set. There are only variable, predicates and the various expressions in the formula $Q[x]$. In $Q[\{x|P[x]\}]$, the symbol string $\{x|P[x]\}$ is no more a referring expression than the part]}. Likewise, in *Principia Mathematica* Russell’s famously introduced notation for the expression called a *definite description*, in logical notation $1\mathrm{x}(P[x])$, which is read in English “the one and only *x* of which $P[x]$ is true.” This expression in English clearly has stand alone meaning and is a referring expression, as in the indentity statement *Venus is the star that appears first in the evening*. But in Russell’s definition it occurs only as part of a longer string and as such has no independent meaning:

$$Q[1x(P[x])] \quad \leftrightarrow_{\mathrm{def}} \quad \exists x(P[x])\wedge\forall y(P[y])\rightarrow y{=}x)\wedge P[y]$$

Russell made use of this technique to such a degree that the axiom system of *Principia* contains no singular terms at all – neither constants nor singular terms made up of factors. In his philosophical work he suggested that proper names in English should likewise be understood via eliminative definitions that quantified only over sense-data.

A second point to make about the parts of eliminative definitions is that although strictly speaking these orthographic parts are not part of the primitive language of the system and therefore do not require a semantic interpretation, if they the definitions are intendeded to provide genuine analyses of the the part, the definition has to conform to preanalytic usage. If part of that usage is to treat the orthographic part as a referring expression, then its usage within the system has to be consistent with that usage. A situation like this arrises in *Principia Mathematica*. In Russell and Whitehead’s axiomn system all the operators for the familiar operations in arithemetic, for example, though the symbols for addition and multiplication are introduced as orthographic parts of longer expressions that are defined in eliminative definitions. Thus, strictly speaking, in formulas like 2+3=7, + is not a functor that stands for the addition operation, and 2, 3, and 7 are not constants that stand for numbers. Nevertheless, for the system to be plausible the axioms must be consistent with an interpretation that treats them as referring expressions and assigns to them their traditional referents. This traditional assignment of referents to the operators and constants of arthememetic is called the *standard interpretation of arithemetic*. It is important because it is used by Gödel in his famous proof that Principia is incomplete. What Gödel assumes can be stated in more general terms. If an axiom system introduces expressions by eliminative definion, then any interpretation of its primitive terms must be extendable to an interpretation of its defined expressions that conforms with their preanalytic usage (i.e. it must conform to the "standard interpretation" of those terms – if there is one.)

We now tintroduce some standard notions and notation for speaking about the parts of an axiom system:

Definitions

We use the special notation $\vdash_{\boldsymbol{S}} P$ to mean that *P is a theorem of **S*** (or equivalently, that *P is in* $\mathbf{Th}_{\boldsymbol{S}}$ or in $\mathbf{DTh}_{\boldsymbol{S}}$.[6] (It will be clear from the context whether we are speaking of $\mathbf{Th}_{\boldsymbol{S}}$ or in $\mathbf{DTh}_{\boldsymbol{S}}$.)

Likewise, *not*$\vdash_{\boldsymbol{S}} P$ means *P is not a theorem of **S***

Let *P* be a sentence and *X* a set of sentences. Then the finite series $P_1,...,P_n$ of sentences (often written as a series of lines down the page) is called a ***deduction*** or ***proof*** of *P* from ***premises*** in *X,* iff the last sentence P_n in the series is *P*, and for each P_i of the series (for i=1,...,n), meets one of three conditions:

i. P_i is an axiom (in $\mathbf{Ax}_{\boldsymbol{S}}$, or in either $\mathbf{Ax}_{\boldsymbol{S}}$ or $\mathbf{D}_{\boldsymbol{S}}$)
ii. P_i is a premise (is in *X*), or
iii. P_i follows by some rule in $\mathbf{R}_{\boldsymbol{S}}$ from earlier members of the series.

[6] The symbol $\vdash$ as it is now used has two meanings. The first is the one being explained here and is "the following is a theorem," as in "$\vdash P\wedge{\sim}P$" which says that $P\wedge{\sim}P$ is a theorem. The symbol was first used by Frege (in the *Begriffsschrift,* 1879), and he viewed it a combination of two symbols: "|", meaning *it is asserted as a theorem that*, and the horizontal "—" meaning *it is true that*. Hence we get the reading *it is a theorem that*. Its second usage is related. In this sense $\vdash$ is placed between a group of premises on its left and a conclusion on its right, and means *there is a proof from the premises to the conclusion,* for example "$\{P\to Q,P\}\vdash Q$" means *there is a proof from P→Q and P to Q.* The two usages are related because in certain rules systems (called *natural deduction* systems -- there is an example below) all the theorems can be deduced in proofs in which all premises are systematically disgarded until at the end of the process the proof has no premises at all. This only happens in the special case in which the sentence proven is logically necessary, and the premiseless proof corresponds to the intuition that a *theorem of logic* is always true, no matter what. Let $\varnothing$ be the empty set. Then one way to say *P* is a theorem of logic is $\varnothing\vdash P$. Thus $\varnothing\vdash P$is equivalent to what we said in the original notation by $\vdash P$. The second notation is more general (and preferred in modern logic) because theoremhood is a special case of having a proof from premises, namely that case in which the premise set is empty.

We say *P* is ***(finitely) deducible*** from *X* in ***S*** (which we abbreviate as $X\vdash_S P$) iff there is some deduction of P from X in ***S***.[7] A deduction in which all the lines are either axioms or follow from previous lines by one of the rules is called a ***proof*** and any sentence deducible from the axioms alone is said to be ***provable*** in ***S***.

Clearly the notions of provable sentence and theoremhood coincide.

Theorem. The following are equivalent:
P is a theorem of ***S*** (written $\vdash_S P$).
There is some deduction of *P* in ***S*** from $\mathbf{Ax}_S$
(or from $\mathbf{Ax}_S$ and $\mathbf{D}_S$).
P is provable in ***S*** (written $\mathbf{Ax}_S\vdash_S P$).

Some axiom systems are successful and others are not. Success turns out to be a complex phenomenon, and a variety of concepts are needed to explain it. These we shall call the ***properties*** of an axiom system. Some of these are definable in completely syntactic terms because they are properties that are concerned solely with the way symbols are arranged on a page.

Others however have to do with whether sentences are true. Truth is a difficult notion and one that we will be returning to repeatedly in these pages. A few remarks about it are required at this point. Normally when we say a sentences is *true* this is short for saying it is true in what philosophers call "the actual world", the world around us that we are talking about. But in general there are other worlds as well, so-called "possible worlds", that we could be talking about. For clarity then we should indicate what world we are describing and treat *truth* as a property of sentences relative to a world. We shall adopt the standard notation $\models$ for the metalinguistic truth-predicate, and let $\mathfrak{A}$ stand for a "model" or "possible world." The way language works is that relative to a world $\mathfrak{A}$ the basic descriptive vocabulary is assigned referents. Facts about how these referents relate to one another then determine the truth-value in $\mathfrak{A}$ of sentences formed from that vocabulary. Of special interest is what is called "the standard interpretation" of the language. If there is one, we shall refer to this interpretation as the model of world $\mathfrak{S}$. This is the assignment in which basic vocabulary stands for its usual referents in the actual world and sentences formed from it have the truth-values that they have in the actual world given their standard referents.

[7] In some of the theoretical material of later lectures we want to allow for the possiblility that the set of premises be countable but infinite. To cover such cases the notion of "syntactic deducibility" we introduce a symbol $\large\vdash$ for arguments in which there are only a finite number of premises, and distinguish it from $\vdash$ which makes no specific requirment about how many premises there are. Accordingly, later we say *X* ***is finitely deducible from*** *P* ***in S***.(in symbols $X\large\vdash_S P$) iff there is some finite subset *Y* of *X* such that $Y\vdash_S P$.

Definition

$\mathfrak{A} \models P$ means *P is true in world (model) $\mathfrak{A}$.*
$\mathfrak{S} \models P$ means *P is true in standard interpretation $\mathfrak{S}$.*

As a matter of convention, when we say *P is true* “without any qualification” (*simpliciter*) what we really mean is *true in* $\mathfrak{S}$ or in *the actual world.*

Definitions. Properties of Axiom Systems

Syntactic Properties

Consistency:

If $\mathbf{L}_{S}$ employs ~ and ∧, we say ***S*** is ***syntactically consistent*** iff
no contradiction is a theorem: for all P in $\mathbf{L}_{S}$, not $\vdash_{S} P \wedge \sim P$,

A more general definition applicable to all $\mathbf{L}_{S}$ is the following:
S is ***syntactically consistent*** iff for some P in $\mathbf{L}_{S}$, not $\vdash_{S} P$.

Independence:

A sentence P is ***independent*** in ***S*** iff neither P nor $\sim P$ is a theorem of ***S*** : not $\vdash_{S} P$ and not $\vdash_{S} \sim P$.

Semantic Properties

Semantic Consistency:

S is ***semantically consistent*** or ***satisfiable***

iff there is some situation in which all the theorems of ***S*** are true simultaneously

iff for some $\mathfrak{A}$, for any theorem P of ***S***, $\mathfrak{A} \vDash P$

Independence:

A sentence P is ***semantically independent*** in ***S***

Iff there is some situation in which the axioms of ***S*** and P are true, and another situation in which the axioms and $\sim P$ are true the theorems of ***S*** could be true together.

iff for some $\mathfrak{A}$, $\mathfrak{A} \vDash P$ and for any theorem Q of ***S***, $\mathfrak{A} \vDash Q$, and
for some $\mathfrak{A}'$, $\mathfrak{A}' \vDash \sim P$ and for any theorem Q of ***S***, $\mathfrak{A} \vDash Q$.

An axiom system ***S*** is ***categorical***

iff there is only one model (or, more precisely, only isomorphic models) in which it is true

iff for any $\mathfrak{A}$ and $\mathfrak{A}'$, if for any theorem P of ***S***, $\mathfrak{A} \vDash P$ and $\mathfrak{A}' \vDash P$, then $\mathfrak{A}$ is isomorphic to $\mathfrak{A}'$.

Properties Relating Syntax and Semantics. Let $\mathfrak{S}$ be the standard interpretation. An axiom system ***S*** is ***sound***

iff all its theorems are true

iff for any P in $\mathbf{L}_{S}$, if $\vdash_{S} P$, then P is true

iff for any P in $\mathbf{L}_{S}$, if $\vdash_{S} P$, then $\mathbf{S} \vDash P$

An axiom system ***S*** is ***complete (relative to the standard model*** $\mathfrak{S}$***)***

iff every *true* sentence in the language $\mathbf{L}_{S}$ is provable as a theorem

iff if P is true, then $\vdash_{S} P$

iff if $\mathfrak{S} \vDash P$, then $\vdash_{S} P$

Conceptual Adequacy of an Axiom System

The axioms of ***S*** are conceptually adequate iff they are conform to the traditional scientific usage.

The definitions of ***S*** are conceptually adequate iff

1. it is not possible to prove any new theorem with the definitions that could not be proved without them and
2. the *definiens* of each definition is synonymous to its *definiendum* according to traditional scientific usage.

Ideally a system would contain only true sentences and it would leave out no true sentence. That is, it would be both sound and complete. If it were, it would automatically be consistent in both senses. Sometimes before we know whether it is sound or complete, we can determine whether the other properties hold: whether it is consistent in one sense or the other, or both, whether any of the proposed axioms are independent, and whether it is categorical. It turns out that sound and complete systems are difficult to construct, sometimes even impossible, and that categorical systems are very rare.

Let us conclude by using these ideas to summarize what was discovered in the 19th century about non-Euclidean geometry:

Facts about Non-Euclidean Geometry

- The negation of the fifth postulate is consistent with the truth of the first four.
- The fifth postulate is independent of the first four.
- The first four postulates are not categorical.
- It is an empirical question which geometry is sound (i.e. true).

D. Symbolic Logic

At this point in the lecture, I want to pause for a moment in our historical survey to introduce the standard notation and inference rules of modern logic. Without these little bits of basic vocabulary it is really impossible to talk about modern logic. Let us pause then to master eight logical symbols.[8]

[8] See Appendix I for alternative symbols are sometimes used in stead of these.

The Logical (Sycategorematic) Signs) of First-Order Logic:

Symbol:	English Translation:	Name of the Logical Operation:
~	*it is not the case that*	sentence negation
∧	*and*	conjunction
∨	*or*	disjunction
→	*if .... then ....*	the conditional
↔	*if, and only if*	the biconditional
∀	*for all*	the universal quantifier
∃	*for some*	the existential quantifier
=	*is identical to*	identity

Though there are huge number of valid inference rules that are more or less useful, we shall only need to appeal in the course of these lectures to a handful. We combine these symbols with letters to form phrases and sentences. The letters we shall use are all from parts of speech familiar from ordinary grammar (e.g. nouns and verbs) but we shall use different typefaces for each of the various parts of speech we shall employ. Which typeface is associated with a given part of speech is now a matter of convention, and the customary assignments are given in the table below.

The Descriptive (Categrematic) Expressions of First-Order Logic:

Typeface:	Part of Speech:
A,B,C	simple sentence
P, Q, R	sentences (both simple and complex)
x, y, z	variables (pronouns in English)
a,b,c,d	constants (proper nouns in English)
F, G, H	predicates (common nouns, adjectives, verb phrases in English)
f,g,h	operators (names of functions or operations, usually on numbers)
P[*x*]	a sentence *P* that contains what is called a ***free occurrence*** of the variable *x*, i.e. an occurrence of *x* that is not contained within a fragment of *P* that begins with *x* or ∃*x*. *P*[*x*] is called an ***open sentence***.
P[*y*]	the sentence that results from *P*[*x*] of replacing some or all of the free occurrences of *x* by free occurrences of *y*
P[*t*]	the sentence that results from *P*[*x*] by replacing some of the free occurrences of *x* by a term *t*. A ***term*** is any proper name, variable or function value.

We may now combine the eight logical symbols with various typefaces to state what in the Middle Ages were called ***consequentiae*** and are today called consequences derived by ***rules of inference****.* These are rules that if followed will always yield a valid, logically acceptable, argument. Stating rules is sometimes tricky. Logicians in the middle Ages became quite apt.

Examples of Mediaeval *Consequentiae*

From the truth of the antecedent the truth of the consequent follows.

From the opposite of the antecedent the opposite of the consequence follows.

In modern logic, a conventionalized form, however, has developed for stating logical rules

How to Write Inference Rules in Logic:

Deduction Rules

We use the display

$$\frac{P_1 \quad .... \quad P_n}{Q}$$

to mean:

From the sentences P_1,...,P_n given as previously proven lines in an argument or as previously proven theorems in an axiom system, it is permissible to infer the sentence Q as the next line in the argument or as an additional theorem in the axiom system.

It will also be convenient to introduce a shorthand notation for logical equivalence. Two sentences are logically equivalent when they are true in exactly the same cases. Equivalents not only are logical consequences from each other, a sentence when it occurs as a part of a longer sentence may be replaced by its equivalent without altering the truth-value of the whole. For this reason equivalents are said to be *substitutable.* We introduce the abbreviation $\Leftrightarrow$ to indicate that the sentences flanking it are logically equivalent:

Substitution Rules

We use the display

$$P \Leftrightarrow Q$$

to mean:

Let R of a line in an argument or a theorem in an axiom system, and let P occur as part of R. It is then permissible to infer a new line or theorem by replacing some occurrences of P by Q in R. Similarly, Let R of a line in an argument or a theorem in an axiom system, and let Q occur as part of R. It is then permissible to infer a new line or theorem by replacing some occurrences of Q by P in R. (Thus $\Leftrightarrow$ means *is substitutable anywhere for*.)

The rules of classical logic that we shall refer to may be summarized using this notation. In this text we shall actually use only a small handful valid rules, but it is useful at this point to meet examples of complete rules sets. We shall consider two. Let us assume that the language for which the rules are formulated consists of a set **L$_{SL}$** of sentences (of "sentence logic") that are made up out of atomic (simple) sentences *A,B*, and *C* by means of parentheses and the connectives $\sim,\wedge,\vee,\rightarrow$, and $\leftrightarrow$.

The first set, which was invented by the American logician Irving Copi,[9] is easy to use and very popular in introductory undergraduate courses. Its terminology is widely used.

Copi's Nineteen Rules for Sentential Logic

Deduction Rules

Conjunction

P
$\underline{Q\quad\quad}$
$\therefore P\wedge Q$

Simplification

$\underline{P\wedge Q}\qquad \underline{P\wedge Q}$
$\therefore P\qquad\quad \therefore Q$

Addition

$\underline{\;P\quad\quad}$
$\therefore P\vee Q$

Disjunctive Syllogism

$P\vee Q\qquad P\vee Q$
$\underline{\sim P}\qquad\; \underline{\;\sim Q}$
$\therefore Q\qquad\quad \therefore P$

Constructive Dilemma

$P\vee Q$
$\underline{(P\rightarrow R)\wedge(Q\rightarrow S)}$
$\therefore R\vee S$

Modus ponens

$P\rightarrow Q$
$\underline{\quad P}$
$\therefore Q$

Modus Tollens

$P\rightarrow Q$
$\underline{\sim Q\quad\;}$
$\therefore \sim P$

Hypothetical Syllogism

$P\rightarrow Q$
$\underline{Q\rightarrow R}$
$\therefore P\rightarrow R$

Substitution Rules

Association

$P\wedge (Q\wedge R) \Leftrightarrow (P\wedge Q)\wedge R$
$P\vee(Q\vee R) \Leftrightarrow (P\vee Q)\vee R$

Commutation

$P\wedge Q \Leftrightarrow Q\wedge P$
$P\vee Q \Leftrightarrow Q\vee P$

DeMorgan's Laws

$\sim(P\wedge Q) \Leftrightarrow \sim P\vee\sim Q$
$\sim(P\vee Q) \Leftrightarrow \sim P\wedge\sim Q$

Double Negation

$\sim\sim P \Leftrightarrow P$

Implication

$P\rightarrow Q \Leftrightarrow \sim P\vee Q$

Transposition or **Contraposition**

$P\rightarrow Q \Leftrightarrow \sim Q\rightarrow\sim P$

Tautology

$P\wedge P \Leftrightarrow P\vee P \Leftrightarrow P$

Exportation

$(P\wedge Q)\rightarrow R \Leftrightarrow P\rightarrow(Q\rightarrow R)$

Equivalence

$P\leftrightarrow Q \Leftrightarrow (P\rightarrow Q)\wedge Q\rightarrow P) \Leftrightarrow (P\wedge Q) \vee(\sim P\wedge\sim Q)$

[9] Though compete in the sense that it is adequate for deducing all valid arguments in sentential logic, Copi's rule set has the curious feature that it is not possible to demonstrate tautologies using just his rules. For that reason it is sometimes called "logic without tautologies." See Irving Copi, *Symbolic Logic*; Leo Simmons, "Logic Without Tautologies," *Notre Dame Journal of Formal Logic*. Tautologies are captured if Copi's rule Exportation is replaced with the "Axiom" of Excluded Middle, *viz.*the rule that it is always correct to enter as a line in an argument or proof a sentence of the form $P\vee\sim P$.

Notice that the rules *modus ponens* and *modus tollens* are the mediaeval consequences cited earlier but formulated here in modern notation.

The second rule set, first proposed by Gerhard Gentzen and called somewhat misleadingly “natural deduction”, it is of more theoretical interest and is the one most commonly learned in more advanced logic courses. (It is the set used in Chapter 2 in the completeness proof of first-order logic.) Instead of postulating a set of “logical truths” as axioms from which other “logical truths” are deduced, the system characterizes what steps or reasoning are valid. These rules are broken down by the logical connectives. For each connective the system provides two rules. One rule explains how to introduce a line in an argument that contains the connective, and the other rule tells how to deduce a new line without the connective from earlier lines containing it. Since the rules explain how to “introduce” and “eliminate” connectives as a process of reasoning, they are said to explain how to “use” the connectives, and to explain their “meaning.” These theoretical claims for the system will be discussed in Chapter 2.

For the rule set to meet the somewhat contrived form of having exactly one introduction rule and one elimination rule for each connective, the special “contradiction” or “falsehood” symbol $\perp$ needs to be introduced. It is treated as a kind of degenerate connective itself and given its own its introduction and elimination rules. Intuitively all we need know about $\perp$ is that it is a sentence that is always false, a kind of fixed contradiction.

Natural Deduction Rules for Sentential Logic

Introduction Rules:

Contradiction Introduction

P

$\underline{\sim P}$

$\therefore \bot$

Reduction to the Absurd

If $\dfrac{A_1,\ldots,A_n,\sim P}{\therefore Q\wedge\sim Q}$ then $\dfrac{A_1,\ldots,A_n}{\therefore P}$

Conjunction

P

$\underline{Q}$

$\therefore P\wedge Q$

Addition

$\underline{P}$

$\therefore P\vee Q$

Modus ponens

$P\rightarrow Q$

$\underline{P}$

$\therefore Q$

Elimination Rules:

Ex Falso Quodlibet[10]

$\underline{\bot}$

$\therefore P$

Double Negation

$\underline{\sim\sim P}$

$\therefore P$

Simplification

$\dfrac{P\wedge Q}{\therefore P}$ $\dfrac{P\wedge Q}{\therefore Q}$

Constructive Dilemma

$P\vee Q$

$\underline{(P\rightarrow R)\wedge(Q\rightarrow S)}$

$\therefore R\vee S$

Conditional Proof

If $\dfrac{A_1,\ldots,A_n,P}{\therefore Q}$ then $\dfrac{A_1,\ldots,A_n}{\therefore P\rightarrow Q}$

These rules are usually justified in the semantics by referring to the meaning of the connectives. This meaning is explained in terms of the way in which the connectives contribute to the truth of the whole sentence, and is usually set forth in tabular form in what are called ***truth-tables***. The truth-tables, given below, explain how the truth-values (T for *true*, or F for *false*) for a complex sentence made up using a connective are determined from the truth-values of its parts. A negated sentence, for example, is true if its part is false, and vice versa. A conjunction is true iff both its conjuncts are. A disjunction is true iff at least one of its dijuncts are. A conditional is false in only one case: when the antecedent is true and the consequent false. A biconditional is true iff its parts have the same truth-values.

[10] From a falsehood (here a contradiction) infer anything whatever.

The Truth Tables for the Sentential Connectives

P	$\sim P$
T	F
F	T

P	Q	$P\wedge Q$	$P\vee Q$	$P\rightarrow Q$	$P\leftrightarrow Q$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	T	T	T	F
F	F	F	F	T	T

Given the truth-tables it is fairly easy to explain why the standard inference rules are valid: if the lines given as premises of the rule are true, then the truth-tables insure that the conclusion of the rule is true also.

Example. Theorem. Modus ponens is valid.
Proof. Suppose both premises P and $P\rightarrow Q$ is true. But given the truth-table for $P\rightarrow Q$ the only case in which $P\rightarrow Q$ and P is true is the case in which Q is also true. Therefore, Q must be true. QED

The sentential connectives and their rules, however, capture only part of the logical syntax of language. Many important arguments turn on the fact that sentences are written in subject-predicate form and employ the concepts of *identity*, *all* and *some*. For these arguments we much enrich the syntax to what is called ***first-order logic*** (also know as ***predicate logic*** or ***quantification theory***).

In this richer syntax we allow for expressions that stand for individual things in the world. These are called ***terms*** and they fall into three types: ***constants*** (proper names), ***variables*** (pronouns), or ***functor-values*** (names of the value of a function). We also employ names for classes and relations called ***predicates***. We combine terms and predicate to form ***simple*** (***atomic***) ***sentences***. One of the relational predicates is the identity sign =. We also allow that the universal quantifier $\forall$ (read *for all*) and the existential quantifier $\exists$ (read *for some*) can be combined with a variable and be put at the front of a sentence. The set $\mathbf{L}_{\mathbf{FOL=}}$ of all sentences of ***first-order logic with identity*** is that made up of constants, variables, functors, and predicates by means of $\sim,\wedge,\vee,\rightarrow,\leftrightarrow,=,\forall$, and $\exists$.

Examples of Sentences in First-Order Logic

First-Order Logic	Mathematical English	Regular English
$\forall xFx$	*For all x, x is F*	*Everything is F*
$\exists yLya$	*For some y, y bears L to a*	*Something bears L to a*
$\forall z(Fz\rightarrow Gz)$	*For all z, if z is F, then it is G*	*All F are G*
$\exists x(Fx\wedge Hx)$	*For some x, x is F and it is G*	*Some F are H*
$\forall x\exists yf(x)=y$	*For all x, and some y, f(x)=y*	*Everything has some f-value*

For arguments using first-order logic we shall need only six additional inference rules, which for purposes of reference we state now.

Natural Deduction Rules for First-Order Logic

Introduction Rules:

Universal Instantiation

Where c replaces some *x*:

$$\begin{array}{c}\underline{\forall xP[x]}\\ P[c]\end{array}$$

Existential Instantiation

If c is a name of convenience that is later dropped in the proof, and c replaces all *x*:

$$\begin{array}{c}\underline{\exists xP[x]}\\ P[\mathrm{c}]\end{array}$$

Law of Identity

The following is always true and a theorem:

$$\forall \mathrm{x}(x=x)$$

Elimination Rules:

Universal Generalization

If c is typical of everything:

$$\begin{array}{c}\underline{P[c]}\\ \forall \mathrm{x}P[x]\end{array}$$

Existential Generalization

If c replaces some *x*:

$$\begin{array}{c}\underline{P[c]}\\ \exists xP[x]\end{array}$$

Substitutivity of Identity

If *y* replaces some *x*:

$$\begin{array}{c}\underline{t=t' \quad P[t]}\\ P[t']\end{array}$$

E. Application of Symbolic Logic to Arithmetic

Perhaps the first successful attempt in the 19th century to apply to mathematics the logistic method was the axiomatization of arithmetic advanced by Guiseppe Peano.[11] We all know the elementary truths of arithmetic. We may identify them with the truths of equality, inequality, addition, subtraction, multiplication, and division that hold among the positive whole numbers including zero. Logicians traditionally call {0,1,2,3,...} the set of ***natural numbers***, and abbreviate its name to **Nn**. Peano was able to deduce the common truths about **Nn** from five axioms formulated in a notation of symbolic logic. His symbolism uses just three undefined primitive terms:

- the one-place predicate (class name) **Nn** that stands for the set of natural numbers **Nn**,
- the operator ***S*** that stands for the successor operation (x+1=y) on **Nn**, and
- the two-place predicate ("transitive verb") ∈ that stands for the membership relation between elements and sets.[12]

[11] Richard Dedekin has previously formulated the axioms in 1888.

[12] ∈ is the Greek letter epsilon, the initial letter of the Greek copula εἶναι, the verb *to be*.

The Primitive (Unefined) Terms of Peano's of Axiom System P:

Primitive Symbol:	English Translation:	Mathematical Idea:
0	*the number 0*	0
Nn	*the set of natural numbers*	{0,1,2,...}
$\boldsymbol{S}(x)=y$	*the successor of x is y*	the successor relation
$\in$	*is a member of*	set membership

Combining these terms with the symbols of symbolic logic Peano formulated an axiom system.

The version of the system we shall construct we shall call **P**. First we make up the set $\mathbf{L_P}$ of sentences in a formal language. These will be all sentences of symbolic logic that we can make up from the primitive terms **Nn**, ***S***, and $\in$, using the usual expressions of symbolic logic: the sentential connectives $\sim$, $\wedge$,$\vee$,$\rightarrow$, and $\leftrightarrow$; the quantifiers $\forall$ and $\exists$; the identity symbol =; and variables *x,y,z,* etc.

To define the axiom system itself we must specify a set of axioms, rules and definition. We shall use Peano's five axioms and the logical rules given earlier.

Peano's Axiom (Logistic) System for Arithmetic, the System P (from *Arithmetices Principia*, 1889)

1. The Postulates (Axioms) for the System P :

	Formulation in English:	Formulation in Logical Notation:
1.	0 is a natural number.	$\vdash_{\mathbf{P}} 0\in\mathrm{Nn}$
2.	Natural numbers are closed under successor.	$\vdash_{\mathbf{P}} \forall x[x\in\mathrm{Nn} \rightarrow\boldsymbol{S}(x)\in\mathrm{Nn})]$
3.	0 is the successor of no natural number.	$\vdash_{\mathbf{P}} \forall x[x\in\mathrm{Nn} \rightarrow\sim\boldsymbol{S}(x)=0)]$
4.	If the successors of two natural numbers are the same, so are those numbers.	$\vdash_{\mathbf{P}} \forall x\forall y([\boldsymbol{S}(x)=\boldsymbol{S}(y)]\rightarrow x=y)$
5.	**Mathematical Induction**. If 0 has a property (is in ***A***) and if a natural number has that property (is in ***A***) only if its successor does also, then all natural numbers have that property (are in ***A***).	$\vdash_{\mathbf{P}}\{0\in\boldsymbol{A}\wedge x\forall y([x\in\mathrm{Nn}\wedge y\in\mathrm{Nn}\wedge x\in\boldsymbol{A}\wedge\boldsymbol{S}(x)=y]\rightarrow y\in\boldsymbol{A})\}$ $\rightarrow\forall x(x\in\mathrm{Nn} \rightarrow x\in\boldsymbol{A})$

2. The Inference Rules for the System P :

The rules of logic stated earlier.

Given these axioms and elementary inference rules Peano was able to define other notions of arithmetic -- addition, multiplication, etc. -- and deduce the ordinary computational truths.

3. Defined Expressions in P. Some Definitions:

Expression Defined:	Definition:	English Translation:
1	***S***(0)	*one*
2	***S***(1)	*two*
3	***S***(2)	*three*
4	***S***(3), etc.	*four*
For any *n* and *m*, *n*+*m*	*n*+0=*n* and *n*+***S***(*m*)=***S***(*n*+*m*)	*plus*

4. Theorems in P. Some simple theorems:

Theorem	$\vdash_P$ 1+0=0	
Proof	$\vdash_P$ S(0)+0=S(0)	def of +
	$\vdash_P$ 1+0=1	def of 1
Theroem	$\vdash_P$ 0+1=1	
Proof	$\vdash_P$ 0+S(0)=S(0+0)	def of +
	$\vdash_P$ 0+S(0)=0	def of +
	$\vdash_P$ 0+1=1	def of 1
Theorem	$\vdash_P$ 1+1=2	
Proof	$\vdash_P$ S(1)=S(1)	axiom of =
	$\vdash_P$ S(1+0)=S(1)	sub of = from previous theorem
	$\vdash_P$ 1+S(0)=S(1)	def of +
	$\vdash_P$ 1+1=2	defs of 1 and 2

Thus it appeared, for a time, that arithmetic was axiomatizable.

F. Application of Symbolic Logic to Set Theory

Another subject that was axiomatized and was of great importance to logic and mathematics was set theory. In the second half of the 19th century the German mathematician Georg Cantor's undertook a project to explain the infinite.[13] In his long career he managed not only to solve many perplexing problems about the infinite that had puzzled thinkers for centuries, but also to make startling new discoveries and pose even deeper questions.

In his work Cantor found that before he could explain the infinite he had to first explain the notion of a "collection" or what we now call a ***set*** or ***class***, because, he reasoned, it is sets that are either finite or infinite depending on "how many" things they contained. But the notion of set is so simple and basic it is hard to see how it might be defined in even more basic terms. The best

[13] A full discussion of Cantor's project is to be found in Michael Hallett, *Cantorian Set Theory and Limitations of Size* (Oxford: Clarendon Press, 1984)

Cantor could do was to say that sets are unities formed by bring together within them all objects that possess a property in common. (In his private writings he speculates that the agent or underlying reality required to organize objects into collections and to sustain the distinct reality of the collection itself above and beyond its elements is God.) He puts the idea as follows, using the concept of "objects following a law" for what we would today refer to as objects exhibiting the defining property of a set:

> By a 'manifold' or 'set' I understand ... every totality of definite elements which can be united to a whole through a law. (1883b, p. 204, n. 1)

Though Cantor himself did not develop his definition formally, by the turn of the century others were translating it into the notation of symbolic logic and using it as an axiom in rigorous deductive statements of Cantor's theory. In 1903 the British philosopher and logician Bertrand Russell explained Cantor's ideas in what has come to be the standard way.[14] To do so he employees some special notation for talking about sets which I shall adapt today in their slightly revised modern form.

First we need a way to talk about the properties that an object must have in order to be in a set. This is the "law" defining the set that Cantor speaks of in his informal characterization above. Suppose for example we want to make up the set of all and only the things that are red. Russell's idea is to do so by using the sentence schema *x is red*. A sentence schema of this sort, namely one that would be a complete sentence if its *x* were replaced by a proper name, was called by Russell a ***propositional function*** and is called today an ***open sentence***. The desired set then is the class of all *x* such that the open sentence *x is red* is true of *x*. Indeed, according to Russell, "A class may be defined as all the terms satisfying some propositional function." Let us use the notation $P[x]$ and $Q[x]$ to represent "propositional functions" (open sentences) containing the variable *x*. We then represent the set of all *x* that satisfy $P[x]$ by the notation $\{x|P[x]\}$, called a *set abstract*. Likewise the set of all things satisfying $Q[x]$ is $\{x|Q[x]\}$.

Notice that the abstract $\{x|P[x]\}$ is the name of a set. That is, it is a referring expression of a certain sort. Together with the predicate $\in$ it provides a new way to translate into symbols an ordinary subject-predicate sentence like *Socrates is a human*. Let *R* be a logical predicate that translates the English adjective *rational* and *A* a predicate that translates *animal*. Assuming Aristotle's definition, $\{x|Rx\wedge Ax\}$ is then a name for the set of humans. Let *s* be a constant (proper name) that translates *Socrates*. Then *Socrates is a human* would be rendered in set theoretic notation by $s\in\{x|Rx\wedge Ax\}$. Here *s* takes the role of an Aristotelian subject term, $\in$ that of the copula, and $\{x|Rx\wedge Ax\}$ that of an Aristotelian predicate. Accordingly, sets may serve as the referents of predicates in traditional subject-predicate propositions, and therefore count as universals

[14] Bertrand Russell, *Principles of Mathematics* [1903] (N.Y: Norton), Chapter II, §§ 23 & 24.

according to Abelard's semantic definition. To the extend that mathematics is committed to set theory it is committed to traditional philosophical realism. Happily there is no danger of the sort of paradox that concerned Boethius. Let p be a constant that translates *Plato*. Both $s\in\{x|Rx\wedge Ax\}$ and $p\in\{x|Rx\wedge Ax\}$ may be true without $\{x|Rx\wedge Ax\}$ entering into the composition of either s or p. (As we shall see shortly, the paradoxes of set theory lie elsewhere.)

Russell makes the informal idea of set precise by using axioms. Then Cantor's sets, Russell says, obey two fundamental laws:

Russell's Axioms for Cantor's Naive Set Theory:

English Formulation:	**Logical Notation:**
1. Principle of Abstraction:	
Some set contains all and only the elements x such that P[x] is true.	$\exists A\forall x(x\in A \leftrightarrow P[x])$
2. Principle of Extensionality:	
Two sets are identical if they have the same members	$A=B\leftrightarrow\forall y(y\in A\leftrightarrow y\in B)$

Though these axioms are elegant and theoretical perspecuous, it is possible to dededuce more useful versions by introducing set abstract notation.

Definition of Set Abstract Notation

$$Q[\{x|P[x]\}] \quad \leftrightarrow_{\text{def}} \quad \exists A(\forall x(x\in A \leftrightarrow P[x])\wedge\forall B(\forall x(x\in A \leftrightarrow P[x])\rightarrow B=A)\wedge P[A]$$

$Q[\{x|P[x]\}]$ means "the open sentence $Q(x)$ is true of the one and only set that contains all and only the elements of which the open sentence $P(x)$ is true."

Theroems

Abstraction	$\exists A(A=\{x|P[x]\}$ and $\forall y(y\in\{x|P[x]\}\leftrightarrow P[y])$)
Extensionality	$\{x|P[x]\}=\{x|Q[x]\}\leftrightarrow \forall y(y\in\{x|P[x]\}\leftrightarrow y\in\{x|Q[x])$

.

If we combine these two principles with several definitions we obtain the axiom system based on Cantor's ideas known today as ***Naive Set Theory***. The definitions given below introduce the basic defined concepts: *subset, the empty set, set intersection, set union*, and *the set of subsets* (called *the power set*) of a set. We let A, B, C, etc. stand for sets.

Definitions of Elementary Sets, Relations on Sets, and Functions on Sets:

Notation:	English Translation:	Logical Definition:	Mathematical Terminology:
$x \neq y$	*x* is not identical to *y*	$\sim(x=y)$	***non-identity*** or ***inequality***
$x \notin \mathrm{A}$	*x* is not an element of set A	$\sim(x \in \mathrm{A})$	***non-membership***
$\mathrm{A} \subseteq \mathrm{B}$	*Everything in* A *is in* B	$\forall x(x \in \mathrm{A} \rightarrow x \in \mathrm{B})$	A is a ***subset*** of B
$\mathrm{A} \subset \mathrm{B}$	A⊆B *& some* B *is not in* A	$\mathrm{A} \subseteq \mathrm{B} \wedge \sim \mathrm{A}=\mathrm{B}$	A is a ***proper subset*** of B
∅ or Λ	*set containing nothing*	$\{x \mid x \neq x\}$	the "***empty set***"
V	*set containing everything*	$\{x \mid x=x\}$	the ***universal set***
$\mathrm{A} \cap \mathrm{B}$	*set of things in both* A *and* B	$\{x \mid x \in \mathrm{A} \wedge x \in \mathrm{B}\}$	the ***intersection*** of A and B
$\mathrm{A} \cup \mathrm{B}$	*set of things in either* A *or* B	$\{x \mid x \in \mathrm{A} \vee x \in \mathrm{B}\}$	the ***union*** of A and B
$\mathrm{A}-\mathrm{B}$	*set of things in A but not in B*	$\{x \mid x \in \mathrm{A} \wedge x \notin \mathrm{B}\}$	the ***relative complement*** of B in A
$-\mathrm{A}$	*set of things not in A*	$\{x \mid x \notin \mathrm{A}\}$	the ***complement*** of A
P(A)	*the set of subsets of* A	$\{\mathrm{B} \mid \mathrm{B} \subseteq \mathrm{A}\}$	the ***power set*** of A

These concepts have to do with sets and their elementary operations. Cantor's real objective, however, was to understand the infinite. If we now add to the ideas above just three more three definitions about the *size* of sets, we can derive some of Cantor's remarkable conclusions about the infinite.

First we need a definition of what it is for two sets to be of the *same size*. Cantor adopted the criterion that we be able to pair up each member of the first set with one and only one member of the second set, in what is called a 1 *to* 1 *correspondence.*

Next we need a definition of what it is for one set to be *bigger than* another. Clearly for one set to be smaller than a second, the first must be the same size as a proper subset of the second. But this condition is not enough since there are infinite sets that meet this condition but which are nevertheless of the same size. For example, the set of even whole numbers is the same size as the set of all whole numbers because we can put the two sets into 1 to 1 correspondence: 2 with 1, 4 with 2, 6 with 3, 8 with 4, etc. Hence one set is not bigger than the other. But the set of even whole numbers meets the condition that it is in 1 to 1 correspondence with a proper subset of the whole numbers, because it can be put into 1 to 1 correspondence to itself which is a proper subset of the whole numbers. Hence Cantor adds the extra condition that the relevant proper subset in question not be in 1 to 1 correspondence with the larger set. Thus A is bigger than B iff B is in 1 to 1 correspondence with some proper subset of A that is not in turn in 1 to 1 correspondence with A.

Lastly we need the notion of a set's being *infinite*. There are a number of ways the infinite might be defined, but Cantor proposes to use one property that seems to be true of all and only infinite sets. This property, moreover, is independent of the sort of entities that make up the set and does not rely on the existence of some standard system of counting or measurement. This is the property we have already seen exhibited by the set of whole numbers, namely that an infinite set can be put into 1 to 1 correspondence with one of its proper subsets.

Cantors Concepts of Set Size:

Notation:	Translation into English:	Logical Definition:
$A\approx B$	A *is the same size as* B	there is a 1-1 mapping from A onto B
$A<B$	B *is bigger than* A	for some C, $C\subset B$ and $A\approx C$, but $\sim(B\approx C)$
A is ***infinite***		for some B, $B\subset A$ and $A\approx B$

With these few notions Cantor was able to prove a remarkable result. For every set there is at least one set that is bigger than it, namely its power set. The proof in naive set theory is straightforward.[15]

Theorem (Cantor). For any set A, $A<\mathbf{P}(A)$

Proof. We show first that it is not the case that $A\approx\mathbf{P}(A)$. We do so by a reduction to the absurd. To begin the proof, we assume the opposite, that $A\approx \mathbf{P}(A)$. Then, there is a 1-1 mapping ***f*** from A onto **P**(A). Now consider the set:

$B = \{x|\ x\in A \wedge {\sim}x\in \boldsymbol{f}(x)\}$.

Clearly B is a subset of A. Hence, since ***f*** maps A onto **P**(A), there must be some *y* in A, such that $\boldsymbol{f}(y)=B$. Consider now two alternatives.

I. Suppose first that $y\in\boldsymbol{f}(y)$. Then, since $\boldsymbol{f}(y)=B$, we may substitute identities and obtain $y\in B$. But then by the definition of B, ${\sim}y\in\boldsymbol{f}(y)$. Hence, $y\in\boldsymbol{f}(y)\rightarrow{\sim}y\in\boldsymbol{f}(y)$.

[15] This result is similar to the theorem that the real numbers have a greater cardinality that the natural numbers, **Nn**<**R**. Assume for a *reductio* that there is a 1-1 mapping *f* from the decimal expansions of **R** to **Nn**. But if r_n is the decimal expansion of *r* paired with *n,* we can define a decimal expansion *d* not paired with any *n*. For any *n,* let $r_{n,m}$ be the *m*-th natural number in the decimal expansion of r_n. Define *d* by looking along the "diagonal" of the ordered decimal expansions: i.e. *d* is defined: for any *n,* $d_n= r_{n,n}+1$. It follows that for any r_n , $d\neq r_n$. But because *d* is a decimal expansion, it corresponds to some real number, and that number is not in the range of *f.* But this contradicts our asssumtion. QED.

If instead of pairing natural numbers with decimal expansions of reals, one pairs elements of A with the ordered 0's and 1's of the range of the charactersistic functions of sets in **P**(A), as similar "diagonal proof" can be fashioned for A<**P**(A).

That one infinite set can be larger than another has been knows since ancient times. Philoponus, the first Christain philosophy to espouse Neoplatonism, rejected the pagan view that the cosmos had no beginning in time. It takes Saturn 12 years to complete a revolution of the earth (or, as we now know, of the sun) and Jupiter 30 years. Clearly Saturn has made more revolutions around the earth than Jupiter. But if time had no beginning, then the number of times each has revolved the earth is infinite, and hence one infinite number would be larger than another, a conclusion thought to be absurd. The argument was known to Arabic philosophers like al Ghazali and underlies Aquinas' cosmological argument for the existence of God, which rejects an infinite regress in causes into the past as absurd.

Galileo and Leibniz argued similarly against an "actual infinite." Clearly, the number of points on the hypotenuse AC of a right triangle ABC is greater than the number of points on its side AB. But if, in fact, the sides were infinitely divisible, then the points on the hypotenuse AC could be mapped 1-1 to those on its side AB by simply dropping a line parallel to BC from each point on AC to a point on AB. Thus one infinitiy would be both larger than and the same size as another. Because this implication is contradictory, there could not be, they reasoned, an actual infinite. See Richard Soroabji, *Time, Creation and the Continuum* (London: Duckworth, 1983), and Samule Levey, "Leibniz on Mathematics," *Philosophical Review* 107 (1998), pp. 49-96.

II. Suppose the opposite, alternative, namely that $\sim y\in \boldsymbol{f}(y)$. Now, since $y\in$A by hypothesis, *y* meets the conditions for membership in B, briefly $y\in$B. Then, since $\boldsymbol{f}(y)$=B, by Substitutivity of identity, $y\in\boldsymbol{f}(y)$. Hence, $\sim y\in\boldsymbol{f}(y)\rightarrow y\in\boldsymbol{f}(y)$.
By I and II, it follows that $y\in\boldsymbol{f}(y) \leftrightarrow \sim y\in\boldsymbol{f}(y)$. But this is a contradiction. Hence the original hypothesis is false, and we have established what we set out to prove, namely it is not the case that A≈**P**(A). There remain two possibilities: either **P**(A)<A or A<**P**(A). However, we can apply the argument above to any B⊆A, showing that it is not the same size as **P**(A). Hence we may generalize that for all B⊆A, ~[B≈**P**(A)]. But logically, this fact entails that there no proper subset B of A such that B≈**P**(A). We have therefore eliminated the possibility that **P**(A)<A. It follows that the only remaining alternative must be true, namely that A<**P**(A). **QED**

This proof, like most proofs in modern mathematics, can in principle be recast entirely into the notation of symbolic logic so that no words from English remain. It is then possible to spell out the proof in detail citing for each step the relevant rule of inference that is applied. Like most mathematicians, however, I shall leave the translation into symbols and the detailed derivation for you to do as an exercise.

G. Reduction of Arithmetic to Logic and Set Theory

When Peano had axiomatized arithmetic and Cantor had worked out the notion of set, the stage was set for a remarkable synthesis. The German logician Gottleb Frege was the first to see that the two theories could be combined by means of symbolic logic into a single axiom system. In the last decade of the 19th century, Frege published an important work in which he deduced as theorems Peano's postulates for arithmetic from a handful of more basic axioms from logic and set theory. On the basis of his technical accomplishment, he advanced a hypothesis about the nature of mathematics generally. Mathematics, he suggested, was a part of logic. This thesis, known as ***logicism***, is rich in implications for mathematics, logic, and philosophy.

For the mathematician logicism explains what mathematics is all about, and what its methods should be. Math turns out to consist of the working out of reason's implications. Its method is the production of axiom systems that, in principle at least, could be formulated in symbolic logic. Non-Euclidean geometry then proves to have been a misleading storm in a teacup. Whatever the peculiarities of geometry, arithmetic, the heart of mathematics, remains groundable in *a priori* truths of reason.

For philosophy logicism breathes new life into a species of rationalism. There still seems to be an important branch of science, namely mathematics, which consists of working out the implications of the self-evident principles of pure thought.

For logic logicism is the supreme validation. Logic becomes the science of pure *a priori* reason. Logic provides the symbolic language, reasoning patterns, and axiomatic method applicable to all the sciences, and for non-empirical mathematics it provides in addition its basic truths.

To show how brief and elegant Frege's sort of theory can be, I will now provide a statement of a basic axiom set sufficient for his purposes. The system will be called **F** (for Fregean Arithmetic). We begin by specifying the set of sentences $\mathbf{L}_{\mathbf{F}}$ of the system. Only two primitive symbols are necessary beyond those of logic, and these two concern sets: the set membership symbol $\in$ and the set abstract $\{x|P[x]\}$. (Here $P[x]$ is an open sentence P containing the variable x.)

Primitive Symbols of Fregean Arithmetic (the System F):

Primitive Symbol:	English Translation:	Symbol Name & Idea:	Example:	Translation of the Example in English:
$\in$	*is a member of*	set membership	$x\in A$	*x is a member of A*
$\{x\|P[x]\}$	*set of x such that* $P[x]$	set abstract	$\{x\|\exists y(x=2y)\}$	*the even numbers*

The set $\mathbf{L}_{\mathbf{F}}$ of sentences in the formal language will be all sentences of symbolic logic that we can make up from the primitive terms $\in$ and $\{x|P[x]\}$ by means of the usual expressions of symbolic logic: the sentential connectives $\sim$, $\wedge$,$\vee$,$\rightarrow$, and $\leftrightarrow$; the quantifiers $\forall$ and $\exists$; the identity symbol =; and variables *x,y,z,* etc.

To define the axiom system we must now specify a set of axioms, rules and definitions. In place of Peano's axioms using primitive ideas from arithmetic, Frege uses axioms from logic and set theory. These may be divided into three sorts.

The Three Kinds of Axioms for the Axiom System F

- Axioms for sentence logic (which was then called the *propositional calculus*)
- Axioms for predicate logic and identity, (then called the *predicate calculus* or *quantification theory* and now called *first-order logic*)
- Axioms for set theory

Since Frege's original work in the 1890's the required axiom set has been reduced and simplified.[16] In place of Frege's original five axioms of the propositional calculus, I shall use a three axiom simplification first proposed by the Polish logician Jan Lukasiewicz in 1930. Frege's original axioms for the quantifiers were reduced and stated in a rigorously logistic system by David Hilbert and Wilhelm Ackermann in 1922. The three axiom version used here is

[16] Gottlob Frege, *Grundgesetze der Arithmetik*, vol. I (1893), vol. II (1903) (Jena: Verlag Hermann Pohle). (A partial translation is avialable in Montgonery Furth, *The Basic Laws of Arithmetic* (Berkeley: Univ. of California Press, 1964). Jan Lukasiewicz and A. Tarski, "Untersuchugen über den Aussagenlalkül," *C. R. Soc. Sci. Varsovie* 23 (1930). David Hilbert and Wilhem Ackermann, *Mathematical Logic* [1928] (N.Y.: Chelsea, 1950). W.V.O.Quine, *Mathematical Logic* [First ed.,1940] (N.Y.:Harper, revised ed.1951; Bertrand Russell, *op. cit.*

due to W.V.O.Quine (1940). For naive set theory I shall use Russell's two axioms of 1903. Strictly speaking the axioms are called ***axiom schemata*** because each schema validate a set of axioms, namely the set of all sentences that have the same form as the schema. (More precisely, an axiom is an instance of a schema obtained as the result of uniformly replacing in the schema all occurrences of non-logical letters by descriptive expressions of the appropriate grammatical type.) One reason I have chosen this particular set of axioms is that it needs only the one rule of inference, *modus ponens*.

The System F for Arithmetic. (Modeled on Frege's, *Grundgesetze der Arithmetik*, 1893,1903):

1. The inference rules of F. $\mathbf{R_F}$ contains just one rule:

If $\vdash_F P$ and $\vdash_F P\rightarrow \mathrm{Q}$, then $\vdash_F Q$ (*modus ponens*)

2. The Axioms of F. The set $\mathbf{Ax_F}$ of axioms consist of all sentences of the following forms:

Axioms of the Propositional Calculus (Sentence Logic) (Lukasiewicz , 1930)

1. $\vdash_F P\rightarrow(Q\rightarrow P)$
2. $\vdash_F(P\rightarrow(Q\rightarrow R))\rightarrow((P\rightarrow Q)\rightarrow(P\rightarrow R))$
3. $\vdash_F({\sim}P\rightarrow{\sim}Q)\rightarrow(Q\rightarrow P)$

Axioms of First-Order with Identity (Quine, 1940)

4. $\vdash_F \forall x(P\rightarrow Q)\rightarrow(\forall xP\rightarrow\forall xQ)$
5. $\vdash_F P\rightarrow\forall xP$ where x is not free in P
6. $\vdash_F\forall xP[x]\rightarrow P[y]$ where $P[y]$ is like $P[x]$ except for containing free occurrences of y where $P[x]$ contains free occurrences of x
7. $\vdash_F\forall x(x=\mathrm{x})$
8. $\vdash_F\forall x\forall y(\mathrm{x}=y \wedge P[\mathrm{x}]) \rightarrow P[y])$ where $P[y]$ is like $P[x]$ except for containing free occurrences of y where $P[x]$ contains free occurrences of x

Axioms of (Naive) Set Theory (Russell's version of Frege, 1903)

9. $\vdash_F \exists A\forall \mathrm{x}(\mathrm{x}\in A \leftrightarrow P[\mathrm{x}])$
10. $\vdash_F A=B\leftrightarrow\forall y(y\in A\leftrightarrow y\in B)$

Definition (Set Abstract)

$Q[\{x|P[x]\}] \leftrightarrow_{\mathrm{df}} \exists \mathrm{A}(\forall \mathrm{x}(\mathrm{x}\in A \leftrightarrow P[\mathrm{x}])\wedge\forall B(\forall \mathrm{x}(\mathrm{x}\in A \leftrightarrow P[\mathrm{x}])\rightarrow B=A)\wedge P[A]$

Theorems (Abstraction for Set Abstracts)

$\vdash_F \exists A(A=\{x|P[x]\}$

$\vdash_F \forall y(y\in\{x|P[x]\}\leftrightarrow P[y])$

Theorem (Extensionality for Set Abstracts)

$\vdash_F \{x|P[x]\}=\{x|\mathrm{Q}[x]\}\leftrightarrow \forall y(y\in\{x|P[x]\}\leftrightarrow y\in\{x|\mathrm{Q}[x])$

Reduction of *n*-place Relations to Sets of *n*-tuples

Definitions

$<x,y> =_{\mathrm{df}} \{x, \{x,y\}\}$ (***ordered pair***)

$<x_1,\ldots,x_n,y>=_{\mathrm{df}} <<x_1,\ldots,x_n,>y>$ (**ordered *n*-tuple**)

Theorems (Propertes of Pairs and *n*-tuples**)**

$\vdash_{\mathbf{F}}$ $<x,y>=<y,x>$ iff $x=y$
$\vdash_{\mathbf{F}}$ $<x_1,\ldots,x_n>=<y_1,\ldots,y_n>$ iff $(x_1=y_1,\ldots,\&\ldots\&\ x_n=y_n)$

Theorems (Abstraction for Relations)
$\vdash_{\mathbf{F}}$ $\exists R\forall \mathrm{x}\forall y\ (<x,y>\in R \leftrightarrow P[x,y])$
$\vdash_{\mathbf{F}}$ $\exists R\forall y_1\ldots y_n(<y_1,\ldots,y_n>\in R \leftrightarrow P[y_1,\ldots,y_n])$
$\vdash_{\mathbf{F}}$ $\forall x_1 y(<x,y>\in\{<x,y>|P[x,y_n]\}\leftrightarrow P[x,y])$
$\vdash_{\mathbf{F}}$ $\forall y_1\ldots y_n(<y_1,\ldots,y_n>\in\{<x_1,\ldots,x_n>|P[x_1,\ldots,x_n]\}\leftrightarrow P[y_1,\ldots,y_n])$

Definitions

$\mathrm{AxB} =_{\mathrm{df}} \{<x,y>\mid x\in\mathrm{A}\wedge y\in\mathrm{B}\}$	***Cartesian product*** of A and B
$\mathrm{A}^2 =_{\mathrm{df}} \mathrm{AxA}$	***Cartesian product*** of A and A
$\mathrm{A}_1\mathrm{x}\ldots\mathrm{xA}_{n+1} =_{\mathrm{df}} (\mathrm{A}_1\mathrm{x}\ldots\mathrm{xA}_n)\mathrm{xA}_n$	
$\mathrm{A}^n =_{\mathrm{df}} \mathrm{A}_1\mathrm{x}\ldots\mathrm{xA}_n$	***Cartesian product*** of $\mathrm{A}_1,\ldots,\mathrm{A}_n$
$\mathrm{V}^2 =_{\mathrm{df}} \mathrm{VxV}$	***The universal (binary) relation***

Theorems
$\vdash_{\mathbf{F}}$ $\mathbf{P}(\mathrm{V}^2) = \{R \mid R\subseteq \mathrm{V}^2\}$ the set of 2-place relations
$\vdash_{\mathbf{F}}$ $\mathbf{P}(\mathrm{V}^n) = \{R \mid R\subseteq \mathrm{V}^n\}$ the set of *n*-place relations

Theorems (Extensionality for Relations)
$\vdash_{\mathbf{F}}$ If $\mathrm{R,R'}\in\mathrm{P}(\mathrm{V}^2)$, then $\mathrm{R=R'} \leftrightarrow \forall xy(<x,y>\in\mathrm{R} \leftrightarrow <x,y>\in\mathrm{R'})$
$\vdash_{\mathbf{F}}$ $\{<x,y>|P[x,y]\}=\{<x,y>|Q[x,y]\}\leftrightarrow\forall xy(\ P[x,y] \leftrightarrow Q[x,y])$

$\vdash_{\mathbf{F}}$ If $\mathrm{R,R'}\in\mathrm{P}(\mathrm{V}^n)$, then $\mathrm{R=R'} \leftrightarrow \forall x_1\ldots x_n(<x_1,\ldots,x_n>\in\mathrm{R} \leftrightarrow <x_1,\ldots,x_n>\in\mathrm{R'})$
$\vdash_{\mathbf{F}}$ $\{<x_1,\ldots,x_n>|P[x_1,\ldots,x_n]\}=\{<x_1,\ldots,x_n>|Q[x_1,\ldots,x_n]\}\leftrightarrow\forall x_1\ldots x_n(P[x_1,\ldots,x_n] \leftrightarrow Q[x_1,\ldots,x_n])$

If we now add several of the elementary definitions of arithmetical ideas, we can state some of the theorems provable within the system.

3. Definitions within F.

Symbol:	English Translation:	Definition:
$\boldsymbol{S}(\mathrm{n})$	*the successor of* n	$n\cup\{n\}$
0	*zero*	$\varnothing$
1	*one*	$\boldsymbol{S}(0)$
2	*two*	$\boldsymbol{S}(1)$
3	*three*	$\boldsymbol{S}(2)$, etc.
Nn	*the natural numbers*	$\bigcap\{A\mid\ 0\in A\ \&\ (x\in A\rightarrow S(x)\in A)\}$
$n+m$	*the sum of* n *and* m	the element e of **Nn** such that there are some non-overlapping sets A and B such that A≈n, B≈*m*, e≈(A∪B)
$n\leq m$	n *is less than* m	$n\subseteq m$ (for *n* and *m* in **Nn**)

Given these axioms and definitions it is possible to prove as theorems Peano's postulates for arithmetic and from them in turn the truths of the simple arithmetic of the natural numbers.

Theorems in F.

Peano's Postulates (as stated earlier) are theorems of **F.**

$\vdash_{\mathbf{F}} 0\in\mathrm{Nn}$

$\vdash_{\mathbf{F}} \forall x[x\in\mathrm{Nn}\rightarrow\boldsymbol{S}(x)\in\mathrm{Nn})]$

$\vdash_{\mathbf{F}} \forall x[x\in\ \mathrm{Nn}\rightarrow\sim\boldsymbol{S}(x)=0)]$

$\vdash_{\mathbf{F}} \forall x\forall y([\boldsymbol{S}(x)=\boldsymbol{S}(y)]\rightarrow x=y)$

$\vdash_{\mathbf{F}}\{0\in\boldsymbol{A}.\wedge x\forall y[x\in\mathrm{Nn}\wedge y\in\mathrm{Nn}\wedge x\in\boldsymbol{A}\wedge\boldsymbol{S}(x)=y]\rightarrow y\in\boldsymbol{A})\}\rightarrow\forall x(x\in\mathrm{Nn}\rightarrow x\in\boldsymbol{A})$

Peano's Theorems are (therefore) also Theorems of **F.**

$\vdash_{\mathbf{F}} 1\leq 3$

$\vdash_{\mathbf{F}} 2+2=4$

Since most mathematicians would identify the soul of mathematics with arithmetic, the success of this derivation goes a long way towards showing that mathematics is "part of" logic, and that the methods of mathematics should be those of the axiomatic logician.

H. Logicism, Russell's Paradox, and *Principia*

Socrates, the patriarch of philosophy, began the discipline by asking questions of the following form: *What is X?* He applied the question to justice, courage, beauty, piety, knowledge, and being. Philosophers ever since have been asking the same thing in various ways. The more basic the topic, the more difficult is to answer. The question *What is knowledge?* generates the entire branch of philosophy known as epistemology. The philosophy of science falls within epistemology, and a special division of the philosophy of science is the philosophy of mathematics.

What is mathematics? This is a question in the Socratic tradition that will concern us during this lecture. It is the central question in the discipline known as *the foundation of mathematics*, a subject that overlaps mathematics, logic and philosophy. The question and its answers are intimately tied to formal logic. We shall see how, with the aid of logic, researchers in this century have discovered that mathematics is a very strange beast indeed.

One answer to the question *What is mathematics?* is called ***logicism***. This school asserts boldly that mathematics is, in a precise sense, part of logic. The truths of mathematics are supposed to be deducible as theorems from the laws of logic.

Logicism then adopts a basic position on the nature of mathematics. It asserts that mathematics is to be explained in terms of axiom systems. On this view, mathematical activity consists of nothing more than the formulation of axioms systems about mathematical topics like geometry and arithmetic. It follows that the appropriate scientific method for mathematics is nothing more than that of producing and appraising axiom systems. That is, a mathematician is an applied logician.

We saw that in the 19th Century the discovery of non-Euclidean geometry complicated the picture. Mathematical axioms like the parallel postulate could no longer be viewed as self-evident truths of reason. Rather, a distinction had to be made that acknowledged that mathematical axiomatization alone was not sufficient for guaranteeing truth-in-the-actual-world or the practical applicability of mathematical results. Mathematicians were forced to admit that empirical investigations were needed to supplement mathematics proper and that it is these extra-mathematical investigations that ultimately determined an axiom system's utility.

Having made this admission, however, mathematicians could still regarded their daily tasks as consisting of proving theorems from abstract principles. Their continuing success at axiomatizing various branches of mathematics lent credence to the overall picture. Peano axiomatized arithmetic. Frege pointed the way to an even grander synthesis. He based a single system on the axioms logic and set theory, and then deduced Peano's postulates as theorems. In the language of the time, he "reduced" arithmetic to logic and set theory.

As soon as this pretty landscape was sketched, however, it was devastated. Frege circulated his work to Russell prior to its publication, and Russell discovered that Frege's axioms harbored a contradiction. Frege, however, went ahead and published his work with a sad appendix sketching Russell's discovery. The blow was so severe that Frege writes that he even began to doubt the truth of arithmetic itself.

You can appreciate how deep the problem is. On the one hand, the axiom system is short, and each of its axioms states an obvious truth, either about logic or sets. On the other hand, a contradiction cannot be true. Where does the problem lie? A solution has occupied logic and mathematics well into this century, and we shall pursue in the next lecture. Let us begin by stating Russell's derivation of what is called ***Russell's Paradox***. The proof is short and simple. It consists of applying the principle that any definable set exists. Russell's strategy is to define the set $\{x|{\sim}(x{\in}x)\}$, called ***Russell's set***. It has the property that if something is in it, then it is not in it, and vice versa.

Theorem. Russell's Paradox. (Discovered by Russell and reported by Frege, *Grudgesetze*, Vol II., 1903):

The combined axioms of logic and naive set theory in the system **F** are inconsistent.

Proof

Axiom 9 reads:

1. $\vdash_{\mathbf{F}}\exists A\forall x(x\in A \leftrightarrow P[x])$

We existentially instantiate Line 1 for A using the name of convenience $\{x|P[x]\}$:

2. $\vdash_{\mathbf{F}}\forall y(y\in\{x|P[x]\}\leftrightarrow P[y])$

Since $P[x]$ is a schema that may be replaced by any open sentence with free-variable x, let us apply it to the case in which $P[x]$ is the formula $\sim(x\in x)$.
Then, $P[y]$ would be $\sim(y\in y)$, and Line 2 yields:

3. $\vdash_{\mathbf{F}}\forall y(y\in\{x|\sim(x\in x)\}\leftrightarrow\sim y\in y))$

Line 3 records the bare fact that an entity y is in Russell's set $\{x|\sim x\in x\}$ if, and only if, y has the defining property of the set, namely $\sim(y\in y)$. Moreover, Line 3 is true for all values of y. Since we may replace all occurrences of y in Line 3 with any entity, let us replace it with the entity named $\{x|\sim(x\in x)\}$.
That is, we obtain by Universal Instantiation from line 3 the instance:

4. $\vdash_{\mathbf{F}} \{x|\sim(x\in x)\} \in \{x|\sim(x\in x)\} \leftrightarrow \sim(\{x|\sim(x\in x)\} \in \{x|\sim(x\in x)\})$.

But Line 4 is contradictory, being of the form $P\leftrightarrow\sim P$. **QED**

This discovery caused a crisis. On the one hand, Frege's axioms were viewed as rightly capturing the ideas of logic and set theory. If these ideas are incoherent, then arithmetic may be incoherent as well. On the other hand, if the axioms fail to capture the ideas properly, then a better, more accurate set of axioms should be sought. Russell chose the second alternative and set about seeking a better axiomatization. Thus began a major ten year collaboration with the philosopher-mathematician Alfred North Whitehead. In 1910 Russell and Whitehead published the first volume of *Principia Mathematica*, a complex axiomatic effort to rescue set theory from paradoxes and to deduce from it the theory not only of the natural numbers but of the real numbers as well.[17] Their solution to the paradoxes is to posit a structural organization on the universe of sets that assigned each set to a rank, called its ***type***. It requires that entities of type τ form sets of the next succeeding type $\boldsymbol{\tau+1}$. Conversely, sets of type $\boldsymbol{\tau+1}$ are composed only of entities from type τ. This restriction is introduced into Frege's axioms for set theory by attaching to each variable a superscript

[17] Alfred North Whitehead and Bertrand Russell, *Principia Mathematica*, vols. I-III (Cambridge: Cambridge University Press, 1910-1911).

indicating its type. The axioms are then revised so that the only theorems of set membership that can be deduced have $\in$ flanked by an expression on the left that is of one type lower than the expression on its right. Russell called his version of set theory the ***theory of types***. We shall call the new axiom system by its traditional name, **PM** (for *Principia Mathematica*).[18] The language $\mathbf{L_{PM}}$ of the new set theory is identical to that of $\mathbf{L_F}$ except that all variables have a type superscript. The axiom system is likewise the same except that variables have type superscripts and type-sensitive axioms for set theory replace Frege's two unrestricted axioms. The revised system is described below.

The Axiom System PM, **the Simple Theory of Types.** (Modeled on Whitehead and Russell, 1910)

1. The Axioms of PM. The set $\mathbf{Ax_{PM}}$ is the same that of $\mathbf{Ax_F}$ except that in axiom schemata 4-8, x is replaced by x^τ and y by y^τ. In addition axiom schemata 9 and 10 are replaced by:

9*. $\vdash_{\mathbf{PM}} \forall y^\tau (y^\tau \in \{x^\tau | P[x^\tau]\}^{\tau+1} \leftrightarrow P[y^\tau])$, for any $P[x^\tau]$
10*. $\vdash_{\mathbf{PM}} \forall x^{\tau+1} \forall y^{\tau+1} [x^{\tau+1} = y^{\tau+1} \leftrightarrow \forall z^\tau (z^\tau \in x^{\tau+1} \leftrightarrow z^\tau \in y^{\tau+1})]$

2. The Inference Rules of PM. The set $\mathbf{R_{PM}}$ is the same as that of **F**, containing just *modus ponens*.

Is the new theory **PM** consistent? Can we deduce a contradiction within it? Proving an axiom system consistent is hard to do in principle. We can prove it inconsistent if we can discover that a contradiction follows as a theorem. It was in this manner that Russell showed that Frege's system was inconsistent. However, **PM** was designed so that neither Russell's paradox, nor any other contradiction known to be provable in **F**, could be proven in **PM**. Thus it appears that **PM** is in fact consistent. Indeed, in the eighty years since the publication of the theory, nobody has found a contradiction. All the axioms appear to be true and its inference rule valid. Hence **PM** appears to be sound. Moreover, within **PM** it is possible to prove as theorems the basic laws of the branches of arithmetic that it endeavors to encompass. Thus it appeared to be complete as well. In the period 1910-1930, it appeared that the research enterprise known as logicism might well have succeeded.[19]

[18] For those familiar with the basic notation of logic a good exposition of Russell and Whitehead's theory is Irving M. Copi, *The Theory of Logical Types* (London: Routledge & Kegan Paul, 1971). A more rigorous statement of the axioms is given in Chapter 2.

[19] An entraining introduction to logicism and the logica personalities concerned is the graphic novel Apostolos Doxiadis and Christos H. Papadimitriou, *Logicomix* (N. Y.: Bloombury, 2009).

II. GÖDEL'S INCOMPLETENESS PROOF

A. Kurt Gödel

In 1931 Kurt Gödel, a young Austrian logician, produced a mathematical proof of the falsity of logicism. The proof is famous. Though less well known to the general public, it ranks with items like the theory of relativity as one of the intellectual feats of the century. What I shall do in the time remaining is sketch the highlights of this achievement. Though the full proof is long and technical, its general strategy is quite accessible.

In a nutshell, what Gödel showed is that any axiom system powerful enough to prove Peano's axioms for arithmetic would leave out at least one truth of arithmetic. Let us begin by being very clear about what arithmetic is.

The Language Arithmetic. Arithmetic is written in a language. In this regard arithmetic is like any other science. The sciences investigate the world and record what they discover in sentences. These sentences are all written in some language. What is somewhat unusual about arithmetic is that its language is mathematical. It consists of the symbols. The use of symbols is dictated by the unusual subject matter of arithmetic. For our purposes here we may identify arithmetic with the study of the natural numbers (the positive integers plus 0) and the operations of addition and multiplication. The subject matter is precise and restricted. The notation of symbolic logic was designed first by Frege and later by Russell and Whitehead in *Principia Mathematica* to talk about arithmetic. In his original paper Gödel takes the axiom system of *Principia* as his reference and refers to its language. He assumes that the part of the language that constitutes "arithmetic", *i.e.* the part which states theorems that uses numerals, the arithmetic operations and relations, has a "standard interpretation." That is, he assumes that the numerals like '2' stand for the number two, and that the operator ***S*** stands for the successor operation, that **+** stands for the addition operation, etc. Thus he understands the set of theorems of the axiom system to be the closure under modus ponens of the set of axioms and definitions. We shall call this axiom system in ***PM*** and its language $\mathbf{L}_{\boldsymbol{PM}}$. It is built up from a basic vocabulary that may be divided into the following "parts of speech:"

The Parts of Speech of $\mathbf{L}_{PM}$

Descriptive Expressions	**Semantics : Referents in the Standard Interpretation**
Predicates: =, ∈, ≤, etc.	stand for sets and relations on numbers
Functors:[20] ***S***,**+**,**x**	stand for operations on numbers
Constants: 0,1,2,...,Nn	name particular numbers or structures composed of numbers
Variables: $x^{\tau},y^{\tau},z^{\tau},...,x^{\tau+1},y^{\tau+1},z^{\tau+1},...$	range over objects at the various levels in the hierarchy of set-types.
Logical Signs	
~,∧,∨,→,↔,∀,∃,(,)	interpreted by truth-tables and compositional rules

For simplicity we shall assume that by a sentence of arithmetic we mean a sentence written in the syntax of $\mathbf{L}_{PM}$.

Arithmetic Describes Facts in the Standard Interpretation and the Actual World. Gödel assumes that the language of arithmetic has a standard interpretation over entities in the actual world, among which are numbers, and that this makes some formulas true and others false. He moreover assumes that we by and large know which is which.

The Standard Interpretation of $\mathbf{L}_{PM}$.

By a ***standard interpretation of*** $\mathbf{L}_{PM}$ which we shall call $\mathfrak{S}$, we mean any interpretation over some set D (called the "domain") that contains as a subset the set of all natural numbers, and has an the interpretation $\mathfrak{S}$ that assigns referents to constants, predicates, and operators that meets the following conditions:

1. for any numeral *n* (which is a constant), $\mathfrak{S}(n)$ is the natural number conventionally assigned to *n*;
2. $\mathfrak{S}$(=) is the identity operation,
   $\mathfrak{S}$(∈) is the set membership relation,
   $\mathfrak{S}$(**Nn**) is the set of natural numbers,
   $\mathfrak{S}$($\underline{<}$) is the less than relation on natural numbers,
   $\mathfrak{S}$(***S***) is the successor operation on natural numbers,
   $\mathfrak{S}$(**+**) is the addition operation on natural numbers, and
   $\mathfrak{S}$(**x**) is the multiplication operation on natural numbers.

We say a sentence *P* ***is true in*** $\mathfrak{S}$, or $\mathfrak{S} \models P$, if *P* is determined to be true by the assignments $\mathfrak{S}$ over the domain.[21]

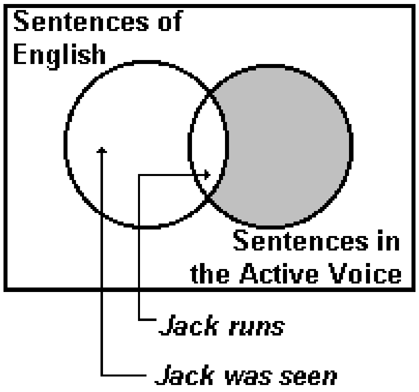
[20] Here ***S*** is the name for the successor function, i.e. if *x* stand for *n*, then **S**(*x*) stands for *n*+1, and $\mathcal{N}\!n$ is the name for the set of natural numbers {0,1,2,...}. In the axiomatic development of $\mathcal{PM}$ most of the vocabulary mentioned here (including ≤,⊆,S,**+**,**x**,0,1,2,..., $\mathcal{N}\!n$) is in fact introduced by abbreviative definitions and does not appear as part of the primitive name, predicates, or functors. All that is needed as primitives are ∈ and =. The Axiom of Abstraction may even be rewritten as explained in Chapter 2 so as to avoid the notation for set abstracts which is then also introduced by definition:

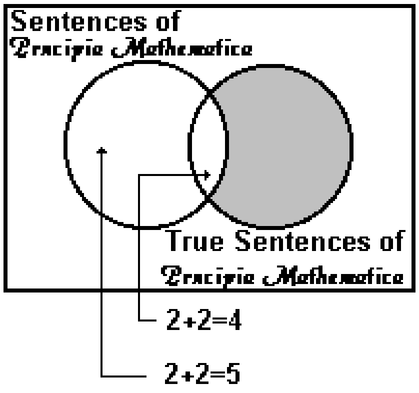
9**. $\vdash_{\mathcal{PM}} \exists y^{\tau+1} \forall y^{\tau} (y^{\tau} \in y^{\tau+1} \leftrightarrow P[y^{\tau}])$, for any $P[x^{\tau}]$.

[21] The notion of true relative to an interpretation is explained in detail in Part II.

Examples of Sentences in $\mathbf{L}_{PM}$:	**Truth-Value in the Standard Interpretation:**
$2\mathbf{+}2=4$	true
$2\mathbf{+}2=5$	false
$2\leq 3\mathbf{+}6$	true
$3\in\mathbf{Nn}$	true
$\forall x^{\tau}\ (x^{\tau}\leq \boldsymbol{S}(27)\rightarrow x^{\tau}\leq \boldsymbol{S}(28))$	true
$\exists x^{\tau}\ (x^{\tau}\mathbf{x}2=3)$	false

Contrasting Facts about Numbers and Sentences. Anything, of course, may be grouped into sets, and this includes the words and symbols of a given language. For example, the well-formed ("grammatical") sentences of English form a set. So too do the well-formed English sentences in the active voice. Indeed, the latter is a subset of the former. Likewise the sentences of $\mathbf{L}_{PM}$ may be grouped into sets. It is the set of all well-formed sentences of ***PM***. An important subset of sentences of ***PM*** is the set of sentences of ***PM*** true in the standard model of arithmetic, which is known less formally as the set of "arithmetic truths."




Sentences of Principia Mathematica
True Sentences of Principia Mathematica
2+2=4
2+2=5

These sets, moreover, are assumed to be part of the actual world. That is, one of the sets of things which exist in our world is the set of grammatical sentences of English, another is the set of English sentences in the active voice. The former contains the latter and a sentence describing that fact about sets would be true in the actual world. Likewise the set of sentences of $\mathbf{L}_{PM}$ exists in the actual world, and so too does the set of true sentences in $\mathbf{L}_{PM}$. A further sentence saying that the latter is a subset of the former would likewise be true in the actual world.

Examples of Sentences:	**Truth-Value in the Actual World:**
The set of English sentences in the active voice is a subset of the set of English sentences.	true
The set of English sentences is a subset of the set of English sentences in the active voice.	false
The set of truths in $\mathbf{L}_{PM}$ *is a subset of the set of sentences of* $\mathbf{L}_{PM}$ *.*	true
The set of sentences of $\mathbf{L}_{PM}$ *is a subset of the set of truths in* $\mathbf{L}_{PM}$ *.*	false

A major feature of Gödel technique is to talk simultaneously about numbers and about language, about what numbers have what properties and about which sets of sentences in $\mathbf{L}_{PM}$ are subsets of others. Thus, in what follows we shall be dealing not only with the truths of arithmetic (in the standard interpretation over the actual world), but also with truths about sets of sentences of $\mathbf{L}_{PM}$ (in the actual world). It will be important to keep straight at each point whether we are talking about numbers or about the language that talks about numbers.

Theorems and Truths: Completeness. There is a distinction between the truths of arithmetic (relative to the standard interpretation) and the set of theorems of *Principia Mathematica*. Frege, and later Russell and Whitehead, set out to axiomatize the truths of arithmetic. Like most of us, they felt they already had a good idea of what these truths are, and they set about devising an axiom system to capture them. If they were to succeed, then all the intuitive truths of arithmetic would follows from the axioms by the rules, and no truth would be left out. The set of theorems of the system would be identical to the set of arithmetic truths. In the language of logistic systems (*Lecture 6*), the axiom system would be ***complete***.

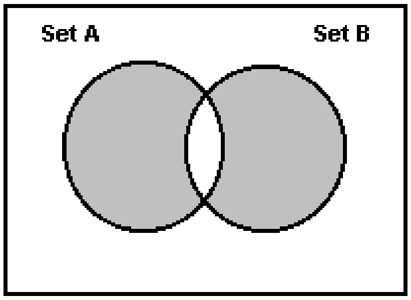
Definitions. Let ***S*** be a logistic (axiom) system for arithmetic and **S** the standard interpretation for arithmetic:

S is ***sound*** means that for any P, if $\vdash_{S} P$, then $\mathbf{S} \models P$

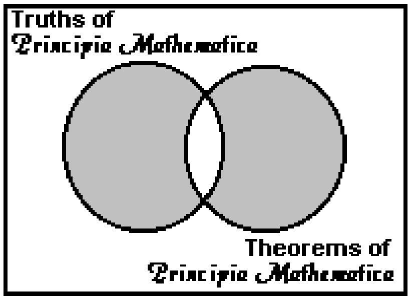
S is ***complete*** means that for any P, if $\mathbf{S} \models P$, then $\vdash_{S} P$

When the context is clear, logicians generally shorten ***sound and complete*** to just ***complete***. Using this convention, then, Frege, and Russell and Whitehead were endeavoring to produce a complete system for arithmetic. Gödel showed that any such attempt would fail.

The issue may be depicted clearly with Venn diagrams. The diagrams below show two worlds. In each there are two sets, and in each world the two sets are identical, because their areas outside the intersections are empty (shaded). The diagram on the right depicts what the actual world ***would be*** like if ***PM*** 's axiom system ***were*** successful.

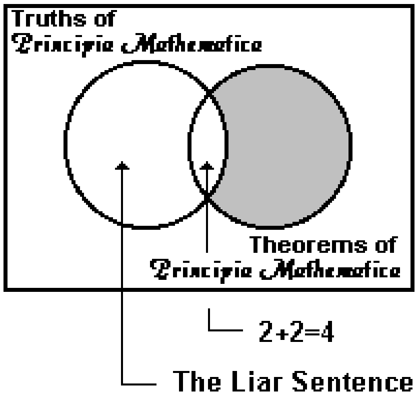


Sets A and B are identical



Completeness:
***P* is a truth of arithmetic iff $\vdash_{PM} P$**

What Gödel showed is that in fact neither the axiom system of ***PM***, nor any other axiom system, could be complete. Given any proposed axiom system (in $\mathbf{L}_{PM}$), he provided a general method that would produce some standard truth of arithmetic (in $\mathbf{L}_{PM}$) that was not a theorem of the system. This truth is stated in what is called the ***Liar Sentence***. He showed that with respect to ***PM*** (or any sound axiom system) that the standard model is rather depicted as follows:



Proof Strategy. Gödel's proof strategy turns on an obvious implication of completeness. Suppose for the sake of argument that the system ***PM*** were complete. Then, by definition, the set of truths written in the language $\mathbf{L}_{PM}$ ("the truths of arithmetic") and the set of theorems of ***PM*** would coincide. Now arithmetic is a very subtle and complex language in which quite complex thoughts can be expressed. Let us make a second supposition. Suppose that the language of arithmetic were rich enough to talk about the set of theorems itself. Suppose, in other words, that $\mathbf{L}_{PM}$ contains a predicate, let us call it $\mathscr{Th}$, that stands for the theorems of ***PM***. It would then follow that $\mathscr{Th}$ stood not only for the theorems of ***PM*** but also for the truths of $\mathbf{L}_{PM}$, because the completeness hypothesis guarantees that these two sets would be identical. It is at this point that a problem arises. It has been known since ancient times that the Liar Paradox follows in any language that contains its own truth predicate. Since, on our hypotheses, $\mathbf{L}_{PM}$ contains $\mathscr{Th}$ that stands for the set of truths written in $\mathbf{L}_{PM}$, it should be possible to prove some sort of liar paradox, and Gödel does so. Such is Gödel's proof. It is a reduction to the absurd of the completeness hypothesis.

Gödel built these insights into a proof. First of all he showed that it is, in fact, possible for arithmetic to talk about its own theorems. Because it is possible to introduce by definition into the language $\mathbf{L}_{PM}$ of *Principia Mathematica*, which contains among its primitive terms expressions like = and ∈, the standard descriptive terms of arithmetic, namely the predicates like and $\leq$ and $\geq$, constants (numerals) 0,1,2,…, and functors like + and ×, it is possible to formulate in $\mathbf{L}_{PM}$ all the formulas we would like to express about arithmetic. Indeed, it is possible to formulate all the propositions of arithmetic, both true and false. For example, the constants (numerals) 2, 4 and 5 can be introduced into the language of $\mathbf{L}_{PM}$ by definition and so can the functor +. Thus, both 2+2=4 and 2+2=5 are formulas of $\mathbf{L}_{PM}$. The former is a "truth of arithmetic" because it is true under the standard interpretation 𝔖 of $\mathbf{L}_{PM}$, i.e. 2+2=4 is true if + stands for the addition function, 2 for the number two, 4 for the number four. Likewise, 2+2=5 is false in the standard interpretations in which 5 stands for the number five. For this reason $\mathbf{L}_{PM}$ is called *the language of arithmetic*.

Now, from of the axioms of *Principia* it is possible to prove as theorems the basic axioms of Peano arithmetic and from these the principle results of arithmetic generally including real number theory.[22] That is, it is possible to prove within *Principia* most of what we think is true in arithmetic. Let us call **Th** all the theorems of Principia formulated in the language of arithmetic. That is **Th** is *the set of theorems of arithmetic*. Gödel showed that, in a sense, it is possible to formulate in *Principia* a predicate 𝒯𝒽 that in a sense names exactly this set.

The constants and predicate of language of arithmetic under its standard interpretation stand for numbers, relations of relations on numbers, and functions on numbers. They do not strictly speaking stands for syntactic entities like variables, constants, predicates, functors, formulas, or sets of formulas. **Th**, however, is a set of formulas. Gödel showed that although strictly speaking under 𝔖, the standard interpretation of $\mathbf{L}_{PM}$, the terms arithmetic do not stand for expressions in the language of $\mathbf{L}_{PM}$, there are in $\mathbf{L}_{PM}$ sets of terms that stand in a 1 to 1 relationship with to various parts of speech (types of expressions in $\mathbf{L}_{PM}$), including a set of numbers that stands in a 1-1 relation to the set of formulas of $\mathbf{L}_{PM}$. In addition, there are predicates that stand for sets and relations among these numbers. In particular he showed that:

- $\mathbf{L}_{PM}$ contains a set of terms that stand under 𝔖 for a set of numbers that stands in 1 to 1 relationship to the set **Th** of theorems of $\mathbf{L}_{PM}$ and
- $\mathbf{L}_{PM}$ contains a predicate, called 𝒯𝒽 (in italics), that stands under 𝔖 for these numbers.

It follows that if *n* is a numeral, then *n*∈𝒯𝒽 is a formula. Moreover, if *n* stands for the number under 𝔖 that stands in the 1 to 1 relation to the formula *P,* then *n*∈𝒯𝒽 is true under 𝔖 if and only if *P* is in **Th**. That is, for any formula *P,* there is a formula in $\mathbf{L}_{PM}$ that is true under 𝔖 if and only if *P* is a theorem of arithmetic. In short, whether a formula is a theorem can be expressed by a formula in $\mathbf{L}_{PM}$.

[22] On the linitations of the theory of real numbers provable in Principa, which makes use of the theory of types, see Irving M. Copi, *The Theory of Logical Types* (London: Routledge, 1971).

It follows that if *Principia* were complete, then truth would also be expressible because truth and theoremhood would coincide by completeness. In particular, if arithmetic were complete and the set **Th** of theorems coincided with the set of truths or arithmetic, then $\mathscr{Th}$ would be in effect a truth predicate because it would hold that $n\in\mathscr{Th}$ iff *P* is a truth of arithmetic, where *n* functions as a kind of indirect name of *P.* The strategy (of Tarski's proof of Gödel's result) is a *reductio*, to show that on the assumption that the elements of $\mathscr{Th}$ stand in a 1 to 1 to the set of truths of arithmetic, the liar paradox follows.

To prove that the language of ***PM*** does not contain its own proof predicate, it is necessary to employ a what is called the lair sentence. This sentence says of itself that it is not true. More formally, it possible to find a term *t* that replaces the variable *x* in the open sentence *x is not true* such that *t* has as its reference the Gödel number of the formula *t is not true.* The resulting formula *t is not true* is the *liar sentence*. To construct this sentence Tarski made use of a more general fact about the expressive power of the language of ***PM***: any open sentence can be appied to some term so that the resulting formula says of the formula that the open sentence is true of it. More formally, for any open sentence *P*[*x*] there is some term *t* that has as its referent the formula *P*[*t*]. This expressive property is called *self-predication*. Tarski simply applies this general fact about expressibility to the particular open sentence *x is not true.*

Most of the work of the proof consists in proving three ***lemmas*** (intermediate metatheorems from which the final result follows):

The Steps of Gödel's Proof

Lemma 1. Self-Predication. Any open sentence can be predicated of itself.

Lemma 2. Tarski's Theorem. No language that expresses its own syntax contains its own truth predicate.

Lemma 3. Expressibility of Theoremhood. Any sound axiom system for arithmetic, like ***PM***, contains a predicate $\mathscr{Th}$ that stands for the set of theorems of the system.

Both lemmas depend on the ability of arithmetic to, in a sense, talk about itself. Let us pause for a moment to ask what it means to talk about arithmetic. What is arithmetic anyway? It was essentially this question that Gödel had been investigating before he came upon his proof. He was doing basic research on a deep question in mathematics: *What is the nature of arithmetical calculation?* His proof of incompleteness was a by-product of this more basic work.

B. Strategy Part 1. Arithmetical Calculations and Recursive Functions.

In some sense we all know what arithmetical calculation is. We spend years as children learning the techniques of adding, subtracting, multiplying and dividing -- methods that apply in principle to any numbers, no matter how large. Moreover all examples of calculation share some common features. In the process, say, of multiplying, we begin with the *multiplicandum* and the *multiplicans*, and by a short series of prescribed steps we arrive at their product. The process proceeds in steps that are themselves simple and relatively foolproof. Such calculation procedures are called ***algorithms***. Gödel was one of the first to investigate their properties, and he succeeded in advancing the first adequate definition for the concept.

There are many ways that the phenomenon of mathematical calculation might be explained, but Gödel hit upon a particularly useful one. He decided to *define* the *set* of calculable arithmetic operations by actually constructing it bit by bit.

Definitions by Abstraction. Normally in philosophy or science we expect a set to be defined by providing a list of the necessary and sufficient conditions for membership within it. Such were Aristotle's essential definitions. He defined the species *human being* by *rational animal*, a paraphrase intended to provide a list of individually necessary and jointly sufficient conditions for membership in the species. Modern zoologists would probably offer a different characterization, but it would be essentially of the same sort. It would provide a more satisfactory list of necessary and sufficient conditions. In computer science what philosophers call definition by necessary and sufficient conditions is called an ***intensional definition***. In logic a definition of a set in terms of such conditions is often said to be ***by abstraction.***

Definition by Necessary and Sufficient Conditions (Abstraction)

A ***definition by necessary and sufficient conditions***, also called an ***intensional definition*** and ***definition by abstraction***, has the following form:

P iff Q
$\mathbf{A} = \mathbf{B}$
$\mathbf{A} = \{x|\ P[x]\}$

The term to the left is called the ***definiendum*** (Latin past participle for *the thing being defined*), and the expression on the right the ***definiens*** (Latin active participle for *the thing doing the defining*.) In addition the definition must meet at least the following conditions:

- The *definiendum* must be viewed as a grammatical unit, and the *definiens* must be a phrase of the same grammatical type (a sentence or a set name).
- The *definiens* and the *definiendum* must have the same meaning in ordinary language or scientific usage. (In philosophical jargon, they are said to have the same ***intention***).
- The *definiens* must not contain any expression defined in terms of the *definiendum*.

Note that strictly speaking the identity sign in $\mathbf{A} = \mathbf{B}$, $\mathbf{A} = \{x|\ P[x]\}$, and $\{x|\ P[x]\}= \{x|\ Q[x]\}$ may be replaced by the biconditional. We do so by replacing the identity assertion by a biconditional logically equivalent to it by means of the Axiom of Extensionality in set theory. The equivalents are as follows:

$\mathbf{A} = \mathbf{B}$ means $\forall x(x\in\mathbf{A} \leftrightarrow x\in\mathbf{B})$
$\mathbf{A} = \{x|\ P[x]\}$ means $\forall y(y\in\mathbf{A} \leftrightarrow y\in\{x|\ P[x]\})$, or equivalently

$$\forall y(y\in\mathbf{A} \leftrightarrow P[y])$$

$$\{x|\ P[x]\}=\{x|\ Q[x]\} \text{ means } \forall z(z\in\{x|\ P[x]\} \leftrightarrow z\in\{y|\ Q[y]\})\text{ , or equivalently}$$

$$\forall y(P[y]\leftrightarrow Q[y])$$

Hence all intensional definitions are at root biconditionals. Accordingly, the *definiendum* and the *definiens* are said to be ***intentionally equivalent***.

Definitions of this sort, however, are not without their problems. It is precisely a definition by abstraction that generates Russell's paradox. We define the set $\{x|{\sim}x\in x\}$. Since we can define it, it becomes a legitimate instance of the universal quantifier in the Frege's Principle of Abstraction, and we get the contradiction. Other contradictions are also generable in the same way.

One possible explanation of the fact that some definitions generate paradoxes is that the sets defined in these cases are very large. Indeed it can be shown that they are transfinite (in the sense explained in *Lecture 6*): they are infinite but not mapable 1 to 1 onto the natural numbers.

Inductive Definitions. There is a special subclass of intensional definitions, however, that is less problematic. It permits the definition of infinite sets and does not generate paradoxes if the sets used in the definition are themselves unproblematic. These definitions do not state categorically a list of necessary and sufficient conditions for membership but rather construct a set element by element.

The construction process begins by specifying some list of starting elements. These are put into the set first. Then some construction methods are stipulated. Each method explains a way of starting with some elements already in the set, and then using them to find some new element that is then added to the set. The process is called a ***definition by induction***, and the set is said to be obtained "by closing the set of initial elements under the operations of construction."

A set **A** is said to be ***inductively defined*** in terms a of a set **B** of basic elements and a set of rules **R** as follows:[23]

1. (Basis Clause.) All elements of **B** are in **A**.
2. (Inductive Clause.) If elements $e_1,...,e_m$ are all in **A**, and if some rule in **R** applied to $e_1,...,e_m$ designates the element e_n, then e_n is in **A**.
3. (Closure Clause) Nothing else is in **A**.

Gödel provided an inductive definition for the set of calculable mathematical operations. He called these calculations ***recursive functions***, but they are today know as the ***primitive recursive function***.[24] He used the format of an inductive definition. First he chose a small number of operations that were obviously "calculations" and put these into a "basis set." He then defined two ways in which (a finite number of) calculation operations might be combined into a single slightly more complex method of calculation. He then closed the basis set under the two methods of "construction."

[23] Inductive definitions are a *subvariety* of definitions by abstraction because the inductive form may be restated using set theory into an identity that does not mention the *definiendum* in the *definiens*:

$$\mathbf{A} = \bigcap\{x|\ \mathbf{B}\subseteq x \wedge \forall y_1...\forall y_n \forall r \forall z[(y_1\in x \wedge ... \wedge y_n\in x \wedge r\in\mathbf{R} \wedge \mathbf{R}[y_1...y_n z]) \rightarrow z\in x]\}$$

(Here $\bigcap\{x|\ P[x]\}$ is set theoretic notation for the set of objects in common among the sets satisfying *P*, i.e. $\{y|\forall z(P[z] \rightarrow y\in z)\}$.)

When the elements of the set are themselves sets, and there are an infinite number of these, new members of the set may be constructed by yet a further method called ***transfinite recursion***: if $\mathbf{C}\subseteq \mathbf{A}$, the $\bigcup\mathbf{C}\in\mathbf{A}$. In this book we shall be dealing only with countably infiite sets that are constructed without the use of transfinite recursion.

[24] Authors subsequent to Gödel now usually call these the ***primitive*** recursive functions to distinguish them from other broader definitions of recursion now current.

The Basic Primitive Recursive Functions

The Successor Function S	For any natural number x:	$\mathbf{S}(x) = x+1$
Constant Functions K_c	For a constant c:	$\mathbf{K}_c(x) = c$
Index Functions I_n	For the n-th position:	$\mathbf{I}_n(x_1,...,x_n,...,x_m)=x_n$

Rules for Specifying New Recursive Functions from Old

The Composition Rule:

Simple Version. If h and g are 1-place recursive functions, then the following defines a 1-place recursive function f:

$$f(x) = h(g(x))$$

General Version. If h is an m-place recursive function and $g_1,...,g_m$ are all n-place recursive functions, then the following defines an n-place recursive function f:

$$f(x_1,...,x_n) = h(g_1(x_1,...,x_n),...,g_m(x_1,...,x_n))$$

The Recursion Rule:

Simple Version. If c is a constant and g is a 2-place recursive function, then the following defines a 1-place recursive function f:

$$f(0) = c$$
$$f(x+1) = g(x,f(x))$$

General Version. If g is a n-place recursive function and h is a n+2 place recursive function, then the following defines a n+1 place recursive function f:

$$f(0,y_1,...,y_n) = \mathrm{g}(y_1,...,y_n);$$
$$f(x+1,y_1,...,y_n) = h(x,f(x,y_1,...,y_n),y_1,...,y_n)$$

Definition of Recursive Function. The set **PRF** of ***(primitive) recursive functions*** is the set defined by induction from the set of basic elements {$\boldsymbol{S}$,$\boldsymbol{K_c}$,$\boldsymbol{I_n}$} (for all constants c and positive integers n) and the rule set {**Composition**, **Recursion**}:

1. (Basis Clause.) {$\boldsymbol{S}$,$\boldsymbol{K_c}$,$\boldsymbol{I_n}$}$\subseteq$ **PRF**
2. (Inductive Clause.) if **R** is in {**Recursion**, **Composition**} and $\mathrm{f}_1,...,\mathrm{f}_m$ are all in **PRF** and g is definable from $\mathrm{f}_1,...,\mathrm{f}_m$ by **R**, then g is in **PRF**.
3. (Closure Clause.) Nothing else is in **PRF**.

Notice that by this definition a function is (primitive) recursive simply by virtue of being a member of the set **PRF**. A function may be in that set and we not know it. It may be in the set while we only possess a poor definition of it, one that is not calculable and which is not sufficient for showing it meets the defining conditions for being in **PRF**. If however we are to show it is in **PRF**, we must be able to show it is either a basic function or is one definable from basic functions by means of recursion or composition. These facts are relevant to the philosophical issue of whether the operations (a.k.a. functions) that the brain performs count as computable (recursive) functions. To show that a function the brain performs is computable one would have to have a definition of that function that met the empirical requirements (whatever they might be) sufficient for knowing that it does in fact characterize a brain process. In addition the definition would have to be such that we could prove it equivalent to a definition that met the membership conditions in **PRF**.

Examples of PRF's

Calculation Operations Defined by the Recursion Rule. The two-place ***addition operation*** **+** is defined by recursion in terms of the one-place successor operation ***S***:

$$x+0=x$$
$$x+\boldsymbol{S}(y)=\boldsymbol{S}(x+y)$$

The 2-place ***multiplication operation*** **x** is defined by recursion in terms of the two-place addition operation **+**:

$$x\mathbf{x}0=0$$
$$x\mathbf{x}\boldsymbol{S}(y)=(x\mathbf{x}y)+x$$

Calculation Operations Defined by the Composition Rule. Given the constants a and b, we construct the linear function *h* with slope *a* and *y*-intersect *b*, namely $y=ax+b$ as follows. (Note that we can *calculate y* from *a, b,* and *x*.)

Let **n** be the number named by the constant *c.* Then $\boldsymbol{K}_c$ is the constant function pairing any *x* with **n**. Further, let $\boldsymbol{I}_1$ be the index function that assigns any *x* to itself. Note that both $\boldsymbol{K}_c$ and $\boldsymbol{I}_1$ are basic recursive functions. Let *a* and *b* be constants (i.e. numerals). We define three functions *f*, *g*, and *h* by composition.

$f(x)=\boldsymbol{K}_a(x)\mathbf{x}\boldsymbol{I}_1(x)$ (In traditional notation: $f(x)=a\mathbf{x}x$)
$g(x)=\boldsymbol{K}_b(x)+\boldsymbol{I}_1(x)$ (In traditional notation: $g(x)=x+b$)
$h(x)=g(f(x))$ (In traditional notation: $h(x)=ax+b$.

Thus, by composition we have defined *h*, the equation for a line with slope *a* and y-intersect *b.*

Using recursive functions, Gödel was able to define a second important idea, that of a *decidable* set of numbers. Intuitively, a decidable set is one for which there is a testing procedure that allows us to "decide" whether a given element is contained in the set.

Definitions

A ***decision procedure*** or ***calculable characteristic function*** for a set **C** of natural numbers is a recursive function *f* defined for all natural numbers n such that *f*(n)=1 if n∈C and *f*(n)=0 if n∉C.

A set **C** is ***decidable*** iff there is some decision procedure for C.

Example of a Decidable Set: The Set of Prime Numbers

We define a finite testing procedure *f* : For any number *n*, divide all numbers less than *n*, one at a time, into *n*.

If none of these (other than 1 and n itself) divides without a remainder, then n is in the set of prime numbers, and set *f*(*n*)=1

Otherwise n is not in the set, and set *f*(*n*)=0

As we shall see in the Chapter 3, recursive functions and decidable sets are of tremendous importance theoretically. They underlie computer science and make computers possible. What we are interested in today, however, are Gödel's application of these ideas to the foundations of arithmetic.

C. Strategy Part 2. Gödel Numbering: Arithmetization of Syntax

What We Can Talk About in Arithmetic. Let us look at the symbolic resources of the syntax in $\mathbf{L}_{\boldsymbol{PM}}$. Though limited, it includes key logical and mathematical expressions. It includes the expressions of logic (the connectives, the quantifiers, the identity sign, and type superscripted variables) as well as expressions for set theory and the definable expressions of arithmetic. With these ideas alone it is possible to write formulas in $\mathbf{L}_{\boldsymbol{PM}}$ that name the key ideas used in calculation. We saw in the last lecture how to define ***Nn***, the name of the set of the natural numbers, and how to give a name (called a ***constant*** or ***numeral***) to each individual natural number. For example, the numeral 0 that names the number zero is a primitive symbol of the syntax. Using the primitive symbol ***S*** that stands for the successor operation, it is then possible to construct a name for the number one, namely ***S***(0). Likewise two has the name ***S***(***S***(0)), etc. We also saw how to define the operations for addition and multiplication.

Intuitively, when we say an expression *E* in $\mathbf{L}_{\boldsymbol{PM}}$ "expresses" or "stands for" a relation of arithmetic, we mean that the sentence made by predicating *E* of a series of numerals is true just when the numbers in fact have the relation *E* expresses. To make this idea clearer, let us establish some notation in the meta- and object languages to distinguish between a number and the numeral that stands for it. Let us use bold face letters to stand for natural numbers. For example, we say that **n** is a natural number. Note that numbers are part of our world and hence **n** is one of the many entities that enters into the facts that make up our world. The sentences of $\mathbf{L}_{\boldsymbol{PM}}$ make assertions about these facts and talk about **n**. The ones that assert facts that actually obtain are true. Now, to talk about **n**, we switch to language. Let us establish the convention that the boldfaced underlined letter $\underline{\mathbf{n}}$ is a numeral (constant) in the syntax of $\mathbf{L}_{PM}$ that stands for the number **n** in the actual world, and which we have just named by '**n**' in the metalanguage. That is, $\underline{\mathbf{n}}$ is a expression in language, specifically an expression of $\mathbf{L}_{PM}$, that names the number **n** which is an object in the actual world. With this convention we can explain more clearly what it is for an expression in $\mathbf{L}_{PM}$ to express or stand for a relation in the world. Let us adopt the notation that $P(x_1,...,x_k)$ is a sentence P containing free variables $x_1,...,x_k$.

Definition

If **R** is an n-place relation on the natural numbers, then **R** is said to be ***expressible*** in $\mathbf{L}_{PM}$ iff there is some sentence $P(x_1,...,x_k)$ of $\mathbf{L}_{PM}$ such that:

$\mathbf{n}_1,...,\mathbf{n}_k$ stand in the relation **R** in $\mathfrak{S}$ iff $\models_{\mathfrak{S}} P(\underline{\mathbf{n}}_1,...,\underline{\mathbf{n}}_k)$

(i.e. $\mathbf{n}_1,...,\mathbf{n}_k$ are elements in the domain of $\mathfrak{S}$ and **R** is a relation on that domain such that $<\mathbf{n}_1,...,\mathbf{n}_k>\in \mathbf{R}$ iff $<\mathfrak{S}(\underline{\mathbf{n}}_1),\ldots, \mathfrak{S}(\underline{\mathbf{n}}_1)>\in\mathfrak{S}(P)$)

If **C** is a subset of the natural numbers, then **C** is said to be ***expressible*** in $\mathbf{L}_{PM}$ if there is some sentence $P(x)$ of $\mathbf{L}_{PM}$ such that, for any n:

n$\in$**C** iff it is a truth of arithmetic that $P(\underline{\mathbf{n}})$.

The Arithmetization of Syntax. Gödel's work on recursive functions suggested to him a way in which the notation of arithmetic might be used to talk about the syntax of the language of arithmetic itself. The process of using the language of $\mathbf{L}_{PM}$ to talk about the expression $\mathbf{L}_{PM}$ is known as ***the arithmetization of syntax***.

Suppose we could map a set of the numbers 1 to 1 with the set of the theorems of ***PM*** in such a way that whatever was true of theorems was reflected in a corresponding fact about their numerical representatives, and vice versa. The numbers could then serve as "proxies" for the theorems themselves. We could investigate the theorems by investigating their numerical proxies and then translating the facts discovered into facts about the theorems themselves.

Such correspondences are often used in mathematics. Indeed, it is a common methodological practice in logic and mathematics to investigate the properties of one structure by discovering those of another structure isomorphic to it.

Definition

A mapping h the elements of structure $\mathbf{S}_1$ onto those of structure $\mathbf{S}_2$ is called an **isomorphism** (and $h(x)$ in $\mathbf{S}_2$ is called the ***proxy*** of x in $\mathbf{S}_1$) iff

1. h is a 1 to 1 mapping, and

2. each relation R on elements of $\mathbf{S}_1$ is paired with a relation R_h of $\mathbf{S_2}$ in such a way that
$x_1,\ldots,x_n$ stand in relation R iff $h(x_1),\ldots,h(x_n)$ stand in relation R_h.

It follows straight from the definition that the properties of proxies correspond to facts about the structure they represent.

Theorem. If there is an isomorphism *h* from structure $\mathbf{S}_1$ onto structure $\mathbf{S}_2$, then:

$x_1,\ldots,x_n$ stands in relation R in $\mathbf{S}_1$ iff $h(x_1),\ldots,h(x_n)$ stand in relation R_h in $\mathbf{S}_2$.

Gödel made proxies of elements in the syntax of $\mathbf{L}_{PM}$ from numbers. The process is called the ***arithmetization of syntax***. He was then able to investigate the properties of these numbers with the assurance that they directly reflected properties of the syntax itself. He actually made numerical proxies for three different sets (structures) within syntax:

- the primitive signs used to make up expressions;
- the strings made up from these signs;
- the ordered sequences (or series) of strings.

He provided number proxies for each element of these three sets.

It is important to recall the difference between a sign and a string of signs. In $\mathbf{L}_{PM}$ the (***primitive***) ***signs*** are:

$\in$, =, $\sim$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\forall$, $\exists$, (,), $x^{\mathbf{n}}_m$ (for each n and m).

These are put together (from left to right) in any combination to make up a ***string***. For example, these sign may be used to make up the strings:

$$\exists x^{\mathbf{6}}_4(x^{\mathbf{6}}_4=x^{\mathbf{6}}_4)$$
$$)x^{345}_2\sim\sim\leftrightarrow\in$$

The first happens to be a well-formed sentence of $\mathbf{L}_{PM}$ and the second is not. Both however are strings. A ***sequence*** (also called a ***series***) of strings is an ordered list of strings. For example, the display of two examples above is composed of two strings. It is a series of length two. Some series of strings are important, especially those series of sentences that amount to proofs.

From the viewpoint of the general strategy of Gödel it is not important to know the details of how he assigned number proxies to each of these sets, and there are in fact many different ways to do so. What is important, however, is that he was able to assign a number substitute for each basic sign of $\mathbf{L}_{PM}$, for each string of these signs, and for each series of such strings. Because it is well known and of some historical interest, Gödel's original method of enumeration is given below.

Gödel Numbering. Before stating Gödel's enumeration process, we must correct a fault that we let pass in the exposition of the basic signs of $\mathbf{L}_{PM}$. So that the concepts definable in terms of them remain decidable, there must only be a finite number of basic signs. We must then figure out some way to construct the infinite number of variables of the form $\mathrm{x}^{\mathbf{n}}{}_{m}$ from a finite number of more basic signs.

One way to do so is to use just the two signs *x* and |. We simply construe the superscript and subscript on the variable $x^{\mathrm{n}}{}_{m}$ as short for stroke strings of the appropriate number, as follows:

Definition of $\mathbf{x}^{\mathbf{n}}{}_{\boldsymbol{m}}$:

$$\mathrm{x}^{\mathbf{n}}{}_{m} = (x\overbrace{|...|}^{n\text{ strokes}})\underbrace{|...|}_{\text{m strokes}}$$

For example, $x^4{}_8$ is (*x*||||)||||||||.

The primitive signs of $\mathbf{L}_{\boldsymbol{PM}}$ are ∈, =, ~, ∧, ∨, →, ↔, ∀, ∃, (,), *x*, and |. Each expression of the language then may be built up as a string from left to right of signs drawn from the finite vocabulary of thirteen items. For example, the string ∃(x||||)||||||||((x||||)||||||||=(x||||)||||||||) (which is $\exists x^4{}_8(x^4{}_8=x^4{}_8)$ in more familiar notation) is made up by stringing together the signs ∃, x, (,), |, and = in a left to right order.

Assigning numbers to each *sign* is easy. There are only thirteen of them, All we need do is assign one number to each. Assigning numbers to *strings*, however is more complex. We want a 1 to 1 mapping, and ideally one that allows us to recover the string in a simple way from the number that represents it.

To map a string onto its unique number, Gödel manufactured a number for the string from the signs that make it up. Each of the signs that compose the string has its number. In addition, it is possible to assign a number to each position in a finite string. It is an easy matter then to define some function that computes a number from these two sorts of information. Let us adopt the notation that at the *i*-th position is the sign E_i. Further, let us assume that the position *i* has been assigned the number n_i, and that the sign E_i itself has been assigned the numerical code m_i. We may now assign to "E_i at position *i*" the value of some number. Let us use some numerical function *f* of n_i and m_i. It does not matter what this function is, so let it be the operation of raising n_i to the power m_i, i.e. $n_i^{m_i}$. Having assigned numbers, one to each sign-position pair, we are ready to assign a number to the entire string of signs. We must compute this number from those numbers assigned to the various sting-position pairs. As before, it does not matter what function we choose. Let us use multiplication. Then, to the string $E_1...E_n$, we shall assign the number $n_1^{m_1}\mathbf{x}...\mathbf{x}n_n^{m_n}$. In this manner we may compute a unique number for each string.

The reverse direction, however, is trickier. Given a number, we would like to be able to recover the unique string that it represents. To achieve this recoverability Gödel made use of the basic fact of number theory that every number is factorable into primes in only one way.

The Fundamental Theorem of Arithmetic

For every positive integer *m* there is a unique finite series of primes $\boldsymbol{p}_1,...\boldsymbol{p}_n$, such that for each i=1,…,*n*, $1<\boldsymbol{p}_i\leq\boldsymbol{p}_{i+1}$) and there is a series of exponents $e_1,...,e_n$ such that

$$m = \boldsymbol{p}_1^{e_1}\mathbf{x}\ldots\mathbf{x}\ \boldsymbol{p}_n^{e_n}.$$

Let us make the following suppositions: that *m* is a string code uniquely factorable into $\boldsymbol{p}_1^{e_1}\mathbf{x}\ldots\mathbf{x}\ \boldsymbol{p}_n^{e_n}$, that each prime $\boldsymbol{p}_i$ indicates a position in the string, and each exponent encodes a basic sign. We could then recover the string itself. The left-most sign is that encoded by e_1, the sign to its right by e_2, etc. Gödel achieved recoverability, then, by encoding the positions by primes. We use the primes 2 and greater. In addition we assign a numerical code to each of the basic signs by a simple 1-1 stipulation. We refer to the code number of a basic sign *E* by the notation $\mathbf{bn}_E$. The Gödel number of the string $E_1....E_n$, which we are now ready to define formally, is abbreviated $\mathbf{n}_{E_1....E_n}$:

The Definition of Gödel Numbers

Gödel Numbering of Basic Signs:
The assignment $\mathbf{bn}_E$ of primes to basic signs *E*, called ***the Gödel number*** of *E*, is defined by this list:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
∈	=	~	∧	∨	→	↔	∀	∃	(	)	*x*	|	***S***

Gödel Numbering of Strings:

The ***Gödel number*** of the string $E_1....E_n$, briefly $\mathbf{n}_{E_1....E_n}$, is defined as follows, where $\boldsymbol{p}_1,...,\boldsymbol{p}_n$ are the first n primes greater than 1 as ordered by $\leq$:

$$\mathbf{n}_{E_1....E_n} = \boldsymbol{p}_1^{\mathbf{bn}_{E_1}}\ \mathbf{x}\ ...\ \mathbf{x}\ \boldsymbol{p}_n^{\mathbf{bn}_{E_n}}$$

Gödel Numbering of Sequences of Signs:

The ***Gödel number*** of the string $S_1....S_n$, briefly $\mathbf{sn}_{S_1....S_n}$, is defined as follows, where $\boldsymbol{p}_1,...,\boldsymbol{p}_n$ are the first n primes:

$$\mathbf{n}_{S_1....S_n} = \boldsymbol{p}_1^{\mathbf{n}_{S_1}}\ \mathbf{x}\ ...\ \mathbf{x}\ \boldsymbol{p}_n^{\mathbf{n}_{S_n}}$$

D. Proof: Part 1. Tarski's Theorem

Gödel knew that there is a problem with truth-predicates. Any self-referring language generates paradoxes that contains a predicate true of all and only the sentences in that language that are true. If all the relevant conditions were met in *PM* -- if *PM* were complete making the set of truths and the set of theorems identical, if its language $\mathbf{L}_{\boldsymbol{PM}}$ had Gödel numbers making it self-referring, and if it also had a predicate *Th* standing for the set of theorems -- then *Th* would be a truth-predicate in $\mathbf{L}_{\boldsymbol{PM}}$. It would stand for all and only the true sentences in *Th*, because it stands for the theorems and by hypothesis the theorems are the same as the true sentences. It should then be possible to prove a contradiction. This is result that Gödel established.[25] Before sketching the steps of the argument, however, we must look more deeply into why a language cannot contain its own truth-predicate. What sort of contradictions are derivable and how?

The Paradox of the Liar. Consider the famous liar sentence:

(L) *This sentence is not true.*

The sentence L has always caused headaches. Grammatically, it is a negated subject-predicate sentence. Consider its subject: *this sentence*. In the context of L the subject refers to a sentence, namely the sentence L itself. Consider next the sentence's predicate: *is true*. We know what it means. A sentence is true if what it says is the case. (This is the view called *the correspondence theory* of truth, which we met in earlier lectures on Greek and mediaeval logic.) We also know the meaning of *not*: it changes a true sentence into a false one and a false to a true one.

If the same sentence contains self-reference, a truth-predicate, and negation, a paradox follows. Consider the following line of reasoning. By the Law of Excluded Middle, the sentence L is either true or false. Suppose it is true. Then what it says must be the case. But what it says is that the sentence is not true. Suppose, on the other hand, that the sentence is false. Then what it says is not the case. Then it is not the case that it is not true. That is, it is true. In either case we get an absurdity.

A common reaction to the paradox is to blame the predicate *is true*. This was the diagnosis of the logician Alfred Tarski who studied the Liar Sentence in detail in the 1930's. He admitted that self-predication is not generally problematic. We frequently engage in self-reference in natural language. We can say, for example:

[25] In the presentation here that subsumes the incompleteness theorem under the Liar Paradox and Tarski's Theorem, I am following exposition of Raymond M. Smullyan*, Gödel's Incompleteness Theorem* (New York: Oxford University Press, 1992). Smullyan's account is both lucid and elegant. He succeeds in organizing a complex subject in a way that makes the key ideas stand out. In the historical note at the end of the lecture I summarize Gödel's original proof strategy that employs the weaker assumption of ω-consistency.

This sentence is in English.
This sentence consists of six words.
This sentence is in the passive voice.
This sentence contains a relative clause.

These sentences are perfectly comprehensible. We can not only understand them, but we can tell right away that the first two are true and the last two are false.

Let us adopt a convention from Tarski. When we underline a word or word group it losses its ordinary meaning and stands instead for the word or word group itself. That is, instead of standing for things in the world, underlined words stand for the symbols themselves. Consider the following sentences:

Socrates was a Greek philosopher.
Socrates was a Greek philosopher.
Socrates is a noun.
Socrates is a noun.

Of these the second and third are false, and the first and fourth are true. (In the Middle Ages a term with its usual referent is said to have ***personal supposition*** and to be term of ***first intention***. Those that stand for themselves have ***material supposition*** and are terms of ***second intention***. In modern logic, if a terms has its usual referent it is said to be ***used***. If it is standing for itself, it is said be ***mentioned***. In this terminology the subject term of L is of second intention and has material supposition. It mentions the very sentence that is being used.)

Tarski showed how to use this sort of convention to summarize a general feature of natural language:[26]

Theorem. Self-Predication in Natural Language

The result may be expressed in the following increasingly precise but equivalent ways:

- In natural language one can always fashion a sentence that makes a predicate *P* apply to that sentence itself by predicating *P* of a name we give to that sentence, for example *C*. That is we fashion the sentence *C is P* with the understanding that *C* is a name for the very sentence *C is P.*
- For any predicate *P* of natural language, there is some sentence *C is P* such that *C* stands for *C is P*:
- There is some sentence *C is P* such that it is true that *C* = *C is P*.

[26] In his original discussion Tarski puts quotation marks around an expression that stands for itself. Underlining is somewhat less confusing. The original papers, both technical, are Alfred Tarski, "The Concept of Truth in Formalized Languages" [1931], and "Foundations of the Calculus of Systems," [1935], reprinted in *Logic, Semantics, Metamathematics* (Oxford: Clarendon Press, 1956). For more readable accounts of Tarski's work in his own words see his papers "Truth and Proof," *Scientific American* (1969) 194, 63-77, and "The Semantic Conception of Truth," *Philosophy and Phenomenological Research* (1944) 4, 341-375.

"Intuitive" Proof. Let *P* be such a predicate. Then *This sentence is P* is an example of the required sort. In that sentence, *this sentence* stands for *This sentence is P,* and thus *this sentence* = $\underline{\textit{this sentence is P}}$ is true. **QED.**

Tarski was then able to show that natural language could not contain its own truth-predicate.

Tarski's Theorem for Natural Languages

There is no predicate *T* in natural language such that, for any sentence *P* in natural language, it is true that

$\underline{\textit{P}}$ *is T iff P*

Proof. The proof is by reduction to the absurd. Assume on the contrary that there is such a predicate *T* such that it is true that

(T) for any *P*, $\underline{\textit{P}}$ *is T* iff *P*

Consider the sentence L below:

(L) *This sentence is not T*

Let us apply T to L to deduce the following special case:

(TL) $\underline{\textit{This sentence is not T}}$ *is T iff this sentence is not T*

But notice that as a result of the meaning of *this sentence* in L, the following identity is true:

(I) *this sentence* = $\underline{\textit{this sentence is not T}}$

Thus by substituting the identity of I into TL we obtain:

(C) *This sentence is T iff this sentence is not T*

But C is a contradiction. Hence the original assumption T must be false. **QED.**

Applying the Liar to Arithmetic. Gödel was able to prove in ***PM*** an analogue to Tarski's Self-Predication Theorem. His technique is not to talk about the expressions themselves, but rather to talk about their Gödel numbers. These Gödel numbers are "proxies" for expressions in that they stand in 1 to 1 correspondence with them. We do so here by augmenting the syntax of $\mathbf{L}_{\boldsymbol{PM}}$ in two ways. First of all we adapt the underlining convention used above so that if

E is an expression of $\mathbf{L}_{\boldsymbol{PM}}$, we use its underlining $\underline{E}$ to stand for Gödel number of *E*. Second, we need a way to talk about the Gödel number of the results of substituting one expression for another. We do so as follows. If *a*, *b*, and *c* are expressions of $\mathbf{L}_{\boldsymbol{PM}}$, then we specify that the expression *[a(b/c)]* is a also a term in of $\mathbf{L}_{\boldsymbol{PM}}$. This term is going to stand for the Gödel number of the result of replacing occurrences of b by *c* in *a*. We interpret it as follows. Consider an occurrence of the term *b* in *[a(b/c)]* when it occurs in a expression E. We call that occurrence of *b* ***ephemeral*** if b is distinct from *c.* It is "ephemeral" in such cases because it has been replaced by *c* and therefore is "not really there" in E. Let $a\frac{b}{c}$ be the result of substituting the expression c simultaneously for all non-ephemeral occurrences of b in a. Then, *[a(b/c)]* is interpreted as standing for $a\frac{b}{c}$.[27]

Augmentation of the Standard Interpretation S of $\mathbf{L}_{\boldsymbol{PM}}$,

1. If *E* is an expression of $\mathbf{L}_{\boldsymbol{PM}}$, $\mathfrak{S}(\underline{E})$ is the Gödel number of *E*,
2. If *a*, *b*, and *c* are expressions of $\mathbf{L}_{\boldsymbol{PM}}$, then $\mathfrak{S}$(*[a(b/c)]*) is the Gödel number of $a\frac{b}{c}$

Gödel showed that in $\mathbf{L}_{\boldsymbol{PM}}$ there is self-predication in the following sense: for any open sentence, there is some numeral in the language that can be used to make a self-referential sentence out of that open sentence. The numeral stands for the very number that is the Gödel number of the sentence when the numeral occupies the position of the variable.

[27] In his original proof, Gödel does not introduce new terms to stand for $a\frac{b}{c}$. He rather assigns a Gödel number to $a\frac{b}{c}$ and used its numeral as we are using *[a(b/c)]*. For expository purposes the current procedure is simpler and presents no problem in principle because, as we shall see in Chapter 2, a first-order syntax may always be expanded to include a denumerable number of new terms (i.e. put in 1-1 correspondence to the natural numbers) and an interpretation (in this case the "standard interpretation" $\mathfrak{I}$ relative to $\mathfrak{S}$ of arithemetic in $\mathbf{L}_{\mathcal{PM}}$) extended to assign referents to them.

Theorem. Self-Predication in *PM*. For any open sentence $P(x)$ of $\mathbf{L}_{PM}$, there is some sentence $P(\underline{\mathbf{n}}_J)$ of $\mathbf{L}_{PM}$ such that $\underline{\mathbf{n}}_J$ "stands for" $P(\underline{\mathbf{n}}_J)$ in the sense that it is a truth of arithmetic that $\underline{\mathbf{n}}_J = \underline{\mathbf{n}}_{P(\underline{\mathbf{n}}_J)}$.

Proof. The proof consists of describing (an ingenious) way to make up the required $P(\underline{\mathbf{n}}_J)$ from $P(x)$. Let $P(x)$ be an open sentence of $\mathbf{L}_{PM}$ and v a variable. Then it is a fact of substitution that:

(i) $P[P[v(v/v)](v/P[v(v/v)])]= P[v(v/v)]\frac{v}{P[v(v/v)]}$

Moreover, by the augmentation (2) of $\mathfrak{I}$ above,

(ii) $\mathfrak{I}(\ [P[v(v/v)](v/P[v(v/v)])]\ )$ = the Gödel number of $P[v(v/v)]\frac{v}{P[v(v/v)]}$

Further, by (1) of the augmentation of $\mathfrak{I}$ above,

(iii) $\mathfrak{I}(\underline{\mathbf{n}}_{P[P[v(v/v)](v/P[v(v/v)])]}\ )$ = the Gödel number of $P[P[v(v/v)](v/P[v(v/v)])]$

Hence, by substituting the identity of (i) into (iii),

(iv) $\mathfrak{I}(\underline{\mathbf{n}}_{P[P[v(v/v)](v/P[v(v/v)])]}\ )$ = the Gödel number of $P[v(v/v)]\frac{v}{P[v(v/v)]}$

By substituting the identity of (ii) into (iv),

(v) $\mathfrak{I}([P[v(v/v)](v/P[v(v/v)])]) = \mathfrak{I}(\underline{\mathbf{n}}_{P[P[v(v/v)](v/P[v(v/v)])]})$

Hence by the truth-conditions for =-statements in $\mathbf{L}_{PM}$, the following is a truth of arithmetic (.i.e. is true in $\mathfrak{I}$):

(vi) $\mathbf{S} \models [P[v(v/v)](v/P[v(v/v)])] = \underline{\mathbf{n}}_{P[P[v(v/v)](v/P[v(v/v)])]}$

We summarize the results so far: for $P(x)$, the name $\underline{\mathbf{n}}_{P[P[v(v/v)](v/P[v(v/v)])]}$ and the sentence $P(\underline{\mathbf{n}}_{P[P[v(v/v)](v/P[v(v/v)])]})$ are such that:

$$\mathfrak{I} \models [P[v(v/v)](v/P[v(v/v)])] = \underline{\mathbf{n}}_{P[P[v(v/v)](v/P[v(v/v)])]}$$

Hence by existential generalization from the particular case of $[P[v(v/v)](v/P[v(v/v)])]$, there is some numeral $\underline{\mathbf{n}}_J$ and sentence $P(\underline{\mathbf{n}}_J)$ such that

$$\mathfrak{I} \models \underline{\mathbf{n}}_J = \underline{\mathbf{n}}_{P(\underline{\mathbf{n}}_J)}..$$

QED.

Tarski's Theorem for *PM*

There is no open sentence $T(v)$ in $\mathbf{L}_{PM}$ such that, for any sentence P in $\mathbf{L}_{PM}$,

$$\mathfrak{S} \models T(\underline{\mathbf{n}}_P) \leftrightarrow P$$

Proof. The proof is by reduction to the absurd. Assume on the contrary that there is a predicate T such that the following is a truth of arithmetic:

(T) For any P, $\mathfrak{S} \models T(\underline{\mathbf{n}}_P) \leftrightarrow P$

Consider the open sentence $\sim T(x)$. We know by the self-predication theorem, that there exists a sentence J, namely,

(J) $\sim T(\underline{\mathbf{n}}_{\mathrm{J}})$

such that (regardless of whether it is true or false) the subject term of the sentence refers to the sentence itself. That is, it is a truth of arithmetic,

(I) $\mathfrak{S} \models \underline{\mathbf{n}}_{\mathrm{J}} = \underline{\mathbf{n}}_{\sim T(\underline{\mathbf{n}}_{\mathrm{J}})}$

Let us instantiate T, taking for P the special case J :

(TJ) $\mathfrak{S} \models T(\underline{\mathbf{n}}_{\sim T(\underline{\mathbf{n}}_{\mathrm{J}})}) \leftrightarrow \sim T(\underline{\mathbf{n}}_{\mathrm{J}})$

Then by substituting the identity of I into TJ, we obtain a truth of arithmetic:

(C) $\mathfrak{S} \models T(\underline{\mathbf{n}}_{\mathrm{J}}) \leftrightarrow \sim T(\underline{\mathbf{n}}_{\mathrm{J}})$

But C is a contradiction. Hence the original assumption must be false. **QED.**

E. Proof: Part 2. Expressibility of Theoremhood

Gödel was able to show that the notion of theoremhood is expressible in arithmetic. That is, he was able to show that the syntax of $\mathbf{L}_{PM}$ contains a predicate $\mathscr{Th}$ that stands for the set **Th**, the set of numerical proxies (Gödel numbers) for the true sentences of the syntax. Gödel's approach was to work out some very general conditions in which concepts are expressible in arithmetic and then show that the set of theorems meets these conditions.

What Gödel noticed was the relatively obvious fact, once you think about it, that much of arithmetic consists of talking about calculations, or in other words about recursive functions and decidable sets. He was able to show that $\mathbf{L}_{PM}$ is completely adequate to the task. Within its syntax it is possible to talk about every kind of arithmetical calculation. He showed that every recursive function is expressible.

Theorem. Recursive Expressibility in $\mathbf{L}_{PM}$.

Every recursive function and decidable set is expressible in $\mathbf{L}_{PM}$:

1. If **R** is an *k-1*-place recursive function on the natural numbers, there is some sentence $P(x_1,...,x_k)$ of $\mathbf{L}_{PM}$ such that, for all $\mathbf{n}_1,...,\mathbf{n}_k$:
   $\mathbf{n}_1,...,\mathbf{n}_k$ stand in the relation **R** iff $\mathfrak{S} \models P(\underline{\mathbf{n}}_1,...,\underline{\mathbf{n}}_k,)$;

2. If **C** is a decidable set of natural numbers, there is some sentence $P(x)$ of $\mathbf{L}_{PM}$ such that, for any **n**:
   $\mathbf{n} \in \mathbf{C}$ iff $\mathfrak{S} \models P(\underline{\mathbf{n}})$.

Proof Strategy. The proof consists of two parts:

- finding in arithmetic a way to name (define a predicate for) each of the basic recursive functions, and
- given the names of the functions to which the processes of recursion and composition are applied, finding a way in arithmetic to name the functions that result from the process.

It will follow that every recursive function "has a name." Since arithmetic contains notation for identity, set membership and the logical operators, Gödel was in fact able to manufacture these names. (The formulas involved are somewhat complex and technical, so we shall not present them here.) Since decidable sets are just special cases of recursive functions, every decidable set will be expressible as well.

Unfortunately Gödel was not able to use the criterion of decidability directly to show that theoremhood is expressible, because theoremhood in arithmetic is, in fact, not decidable. He was able, nevertheless, to make up a complex sentence that applied to theorems by talking about another set that is decidable. Since this more basic set is decidable, it is expressible, and he used its predicate to define $\mathscr{Th}$.

The key decidable idea is the relation *is-a-deduction-of.* Though there is no method for coming up with a proof, there is one for testing whether any series of sentences constitutes a proof. This situation which at first sight seems paradoxical in fact accords well with the experience of every logic student. Those of you who have actually tried to come up with proofs in logic classes are very familiar with the fact that it is hard to do. There is no easy way to invent a proof, and doing so is rather a matter of creativity than simple method. But once you have a candidate for a proof, it is a very easy matter to check whether it is in fact it makes any mistakes. Two basic facts underlie this situation. On the one hand, there is no algorithm for testing whether a sentence is theorem of arithmetic (or even of first-order logic[28]). On the other hand, it is possible to devise simple testing procedures that will decide whether any series of sentences is a proof in an axiom system.

Let us make this concrete for the case of arithmetic. We begin by focusing on the relation *is-a-deduction-of* that holds between a proof and the sentence it proves. This relation is decidable. Gödel shows that it is so indirectly by proving that a relation on Gödel numbers, which he called **Ded**, is decidable. If $\mathbf{n}_{\mathrm{D}}$ is the Gödel number of the deduction D, and $\mathbf{n}_{P}$ the Gödel number of the formula *P,* then the relational fact $\mathbf{Ded}(\mathbf{n}_{\mathrm{D}}, \mathbf{n}_{P})$ on numbers holds exaclty when D is a decution of *P* in ***PM***. In addition it is possible to show that **Ded** is decidable. It follows then by the expressiblity theorem that **Ded** is expressible by some open sentence in $\mathbf{L}_{\boldsymbol{PM}}$.

Definition. $\mathbf{Ded}(\mathbf{n}_{\mathrm{D}}, \mathbf{n}_{P})$ iff D is a deduction of *P* in ***PM.***

Theorem. Decidability of *Deducibility in PM*.
The relation **Ded** is decidable.

Proof Sketch. The test for whether D is a genuine deduction goes roughly as follows. We start by taking the Gödel number $\mathbf{n}_{\mathrm{D}}$ of D. From it we obtain the Gödel numbers, in order, of the sentences that make up D: $\mathbf{n}_{P_1},\ldots,\mathbf{n}_{P_k}$. For each $\mathbf{n}_{P_m}$ (for $m \leq k$), the sentence P_m must either be an axiom or follow from earlier sentences in the D by the rule *modus ponens*. If so, D is a genuine deduction of *P* and $\mathbf{n}_{\mathrm{D}}$ stands in the relation **Ded** to $\mathbf{n}_{P}$. If not, then it isn't. Whether *P* is or is not is entirely evident by inspection. Hence, **Ded** is decidable.

[28] Unlike arithmetic, first-order logic (the logic of the sentential connectives, ∀ and ∃) is complete in the special sense that its valid arguments can be inductively defined by transparent syntactic methods. However, there is no decision procedure for testing whether a sentence is a logical truth of first-order logic (and hence a theorem in the complete axiom system). This result, known as ***the undecidability of first-order logic***, is important and will be discussed in Part **III**. Here, however, it is relevant to note the following. Since all sciences include logic, the truths of logic are a subset of those of arithmetic. Moreover, if a subset is undecidable, so is the set itself. Thus, arithmetic must be undecidable.

The expressibility theorem tells us that since **Ded** is decidable, it is expressible in the syntax of $\mathbf{L}_{PM}$ by some open sentence, which we will arbitrarily decide to call $\mathscr{Ded}$. Thus, we write $\mathscr{Ded}(x,y)$ to "say" in $\mathbf{L}_{PM}$ that "*x is (the Godel number of) deduction of (a formula with Gödel number) y*," and we state the special case that D *is a deduction of P* by the sentence $\mathscr{Ded}(\underline{\mathbf{n}}_{\mathrm{D}}, \underline{\mathbf{n}}_{P})$.

Corollary. Expressibility of *Deducibility in* $\mathbf{L}_{PM}$**..**
There is some sentence $\mathscr{Ded}(x,y)$ of ***PM*** such that, for any $\mathbf{n}_{\mathrm{D}},\mathbf{n}_{P}$:

$$\mathbf{Ded}(\mathbf{n}_{\mathrm{D}},\mathbf{n}_{P}) \text{ iff } \mathfrak{S} \models \mathscr{Ded}(\underline{\mathbf{n}}_{\mathrm{D}}, \underline{\mathbf{n}}_{P}).$$

By employing the sentence $\mathscr{Ded}(x,y)$, we reach the objective of defining within $\mathbf{L}_{PM}$ a predicate $\mathscr{Th}$ that stands for **Th**, the set of Gödel numbers of the theorems of ***PM***. We use the sentence $\exists y\mathscr{Ded}(y,x)$, which is true for the value $\mathbf{n}_P$ of x exactly when $\mathbf{n}_P$ is the Gödel number of a theorem of ***PM***. In $\mathbf{L}_{PM}$ we may write $\exists y\mathscr{Ded}(y,x)$ to say "*x is(the Gödel number of) a proof*," which is just another way of saying *x is a theorem*. Thus, we define $\mathscr{Th}(y)$ as $\exists x\mathscr{Ded}(x,y)$.

Definition. Th is $\{\mathbf{n}_P | \vdash_{PM} P\}$, i.e. the set of Gödel numbers of theorems in ***PM***.

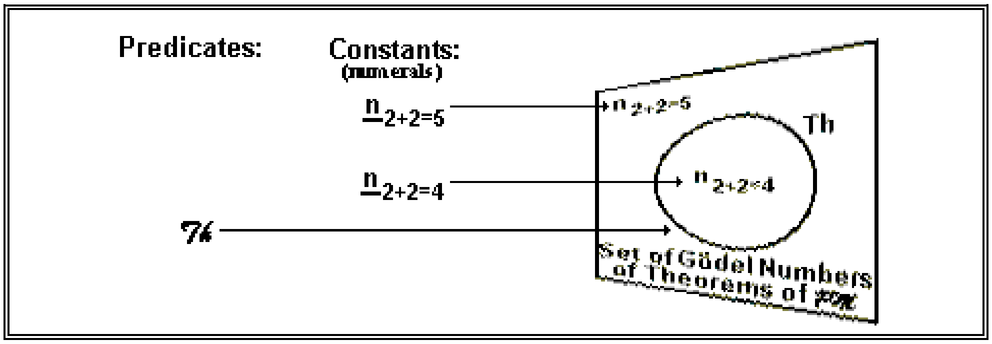
Theorem. Expressibility of *Theoremhood in* $\mathbf{L}_{PM}$**.**
The set **Th** is expressible in $\mathbf{L}_{PM}$:
there is some sentence $\mathscr{Th}(x)$ of $\mathbf{L}_{PM}$ such that, for any $\mathbf{n}_P$:

$$\mathbf{n}_P \in \mathbf{Th} \text{ iff } \mathfrak{S} \models \mathscr{Th}(\underline{\mathbf{n}}_P).$$

Proof Sketch. Since **Ded** is decidable, it is expressible in $\mathbf{L}_{PM}$ by some predicate, which we may call $\mathscr{Ded}$.

Moreover, since **Ded** is expressible by $\mathscr{Ded}$, the open sentence $\exists x\mathscr{Ded}(x,\underline{\mathbf{n}}_P)$ "says" (the equivalent in Gödel numbers of the fact) that it is true that *P* is proven in some deduction, or in other words $\vdash_{PM}P$. Hence $\exists x\mathscr{Ded}(x,\underline{\mathbf{n}}_P)$ is a truth of arithmetic exactly when $\mathbf{n}_P \in \mathbf{Th}$. Thus **Th** is expressible in $\mathbf{L}_{PM}$.

We may diagram the expressibility of **Th** by $\mathscr{Th}$ as follows:

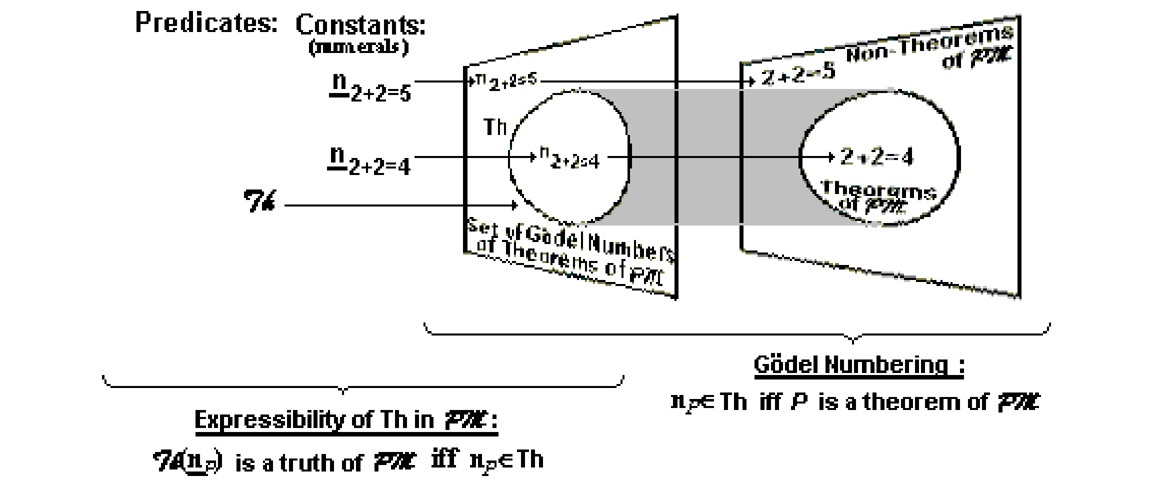


The Expressibility of Th in $\mathbf{L}_{PM}$: $\mathbf{n}_P\in$Th iff $\mathscr{Th}(\underline{\mathbf{n}}_P)$ is a truth in $\mathbf{L}_{PM}$

F. Proof of Incompleteness

Let us summarize the situation. The predicate $\mathscr{Th}$ is true of the set of numbers **Th** that stand as proxies for the theorems of *PM*, and we are using the notation $\mathscr{Th}(\underline{\mathbf{n}}_P)$ to say in $\mathbf{L}_{PM}$ that the sentence with Gödel number $\mathbf{n}_P$ is a theorem of *PM***.**

The full situation may be depicted in a diagram. The picture below illustrates first that numerals (constants) in $\mathbf{L}_{PM}$ stand for numbers which are in turn proxies for sentences, and second that predicates of $\mathbf{L}_{PM}$ stands for sets of numbers, which in turn are proxies for a set of expressions. The predicate $\mathscr{Th}$ in particular stands for the set of numbers that are proxies for the theorems of *PM*.



We have the predicate $\mathscr{Th}$ that we know expresses in $\mathbf{L}_{PM}$ the set of theorems of ***PM***, and via Gödel numbers we have the resources such that for any sentence *P* of $\mathbf{L}_{PM}$ there is some numeral in the language that is the name of the Gödel number of *P.* Indeed, the expressibility of **Th** by $\mathscr{Th}$ insures that we can say in $\mathbf{L}_{PM}$ that a sentence is a theorem of ***PM***. By negation then we can say that something is not a theorem.

One curious consequence of this ability is that some self-referring sentences must be true. In particular, the ***Liar Sentence*** which says of itself that it is not a theorem must be true.

Definition

By the ***Liar Sentence*** we mean the following sentence:

(J) $\sim\mathscr{Th}(\underline{\mathbf{n}}_J)$

in which J stands for the sentence itself, and which is such that we may formulate a truth of arithmetic:

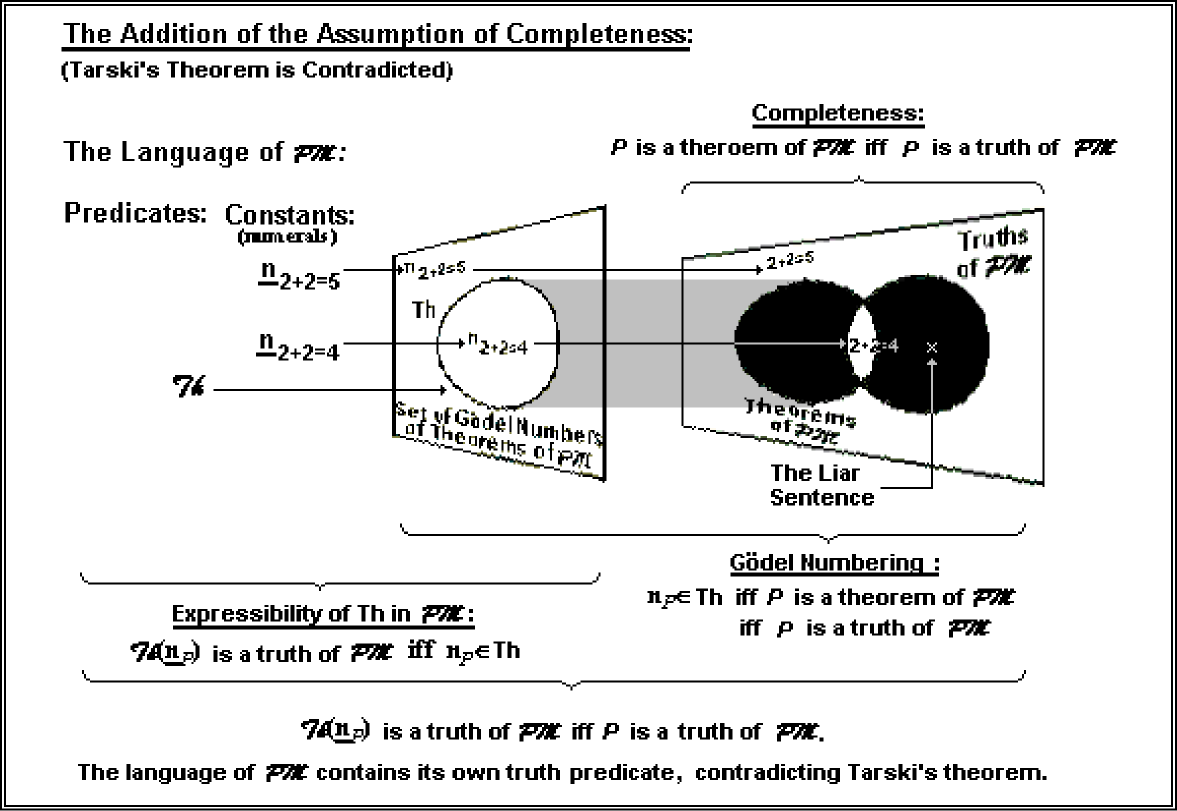
$$\Im \models \underline{\mathbf{n}}_J = \underline{\mathbf{n}}_{\sim \mathcal{Th}(\underline{\mathbf{n}}_J)}$$

What J says, namely that it itself is not a theorem, must be true -- if in fact the set of theorems of ***PM*** is consistent. Gödel was able to prove this fact.

Theorem. If ***PM*** is consistent, then not $\vdash_{PM} \sim\mathcal{Th}(\underline{\mathbf{n}}_J)$

Though we shall not need to appeal to this result in the proof of incompleteness set forth below, its derivation is part of Gödel's original proof strategy and it is outlined in the supplementary material at the end of the lecture.

Supposing we have proven the previous theorem. For the sake of argument let us now entertain the hypothesis (to be refuted) that *PM* is complete. Gödel saw that something then goes wrong. If *PM* were complete, there should be no true sentence that is not a theorem. But the Liar sentence is such a sentence. It is a true non-theorem. Completeness therefore must be a false hypothesis. Such is the strategy of Gödel's original proof. The contradictory picture it draws is the following:


The Addition of the Assumption of Completeness:
(Tarski's Theorem is Contradicted)
The Language of PM:
Predicates:
Constants:
(numerals)
n2+2=5
n2+2=4
Th
n2+2=5
Th
n2+2=4
Set of Gödel Numbers of Theorems of PM
Completeness:
P is a theroem of PM iff P is a truth of PM
2+2=5
Truths of PM
2+2=4
Theorems of PM
The Liar Sentence
Gödel Numbering :
nP∈ Th iff P is a theorem of PM
iff P is a truth of PM
Expressibility of Th in PM:
Th(nP) is a truth of PM iff nP∈ Th
Th(nP) is a truth of PM iff P is a truth of PM.
The language of PM contains its own truth predicate, contradicting Tarski's theorem.

The argument that Gödel uses to establish the properties of the Liar Sentence, which are somewhat technical, are sketched in the supplementary material appended at the end of the chapter. It is possible however to show incompleteness in a simpler way that does not draw upon the Liar sentence.

Incompleteness follows directly from the impossibility of a truth-predicate (Tarski's Theorem) and the expressibility of theoremhood. The results needed are summarized below.

Summary of Results

Lemma 1. Tarski's Theorem for *PM*

There is no open sentence $T(v)$ in $\mathbf{L}_{PM}$ such that,

for any sentence P in $\mathbf{L}_{PM}$, $\mathfrak{S} \models T(\underline{\mathbf{n}}_P) \leftrightarrow P$.

Lemma 2. Expressibility of *Theoremhood in* $\mathbf{L}_{PM}$.

The set **Th** is expressible in $\mathbf{L}_{PM}$:

there is some sentence $\mathscr{Th}(x)$ of $\mathbf{L}_{PM}$ such that, for any $\mathbf{n}_P$:

$\mathbf{n}_P \in \mathbf{Th}$ iff $\mathfrak{S} \models \mathscr{Th}(\underline{\mathbf{n}}_P)$.

The Incompleteness Theorem (Gödel).

PM is incomplete.

Proof (from Tarski's Theorem)

The proof strategy is to show that if ***PM*** were complete then Tarski's Theorem would be false. Since Tarski's Theorem is true, it would then follow that ***PM*** could not be complete. Let us assume completeness:

(Completeness) For any P in $\mathbf{L}_{PM}$, $\vdash_{PM} P$ iff it is a truth of arithmetic that P.

We claim then that the open sentence $\mathscr{Th}(x)$ meets the conditions for a truth-predicate for $\mathbf{L}_{PM}$, namely:

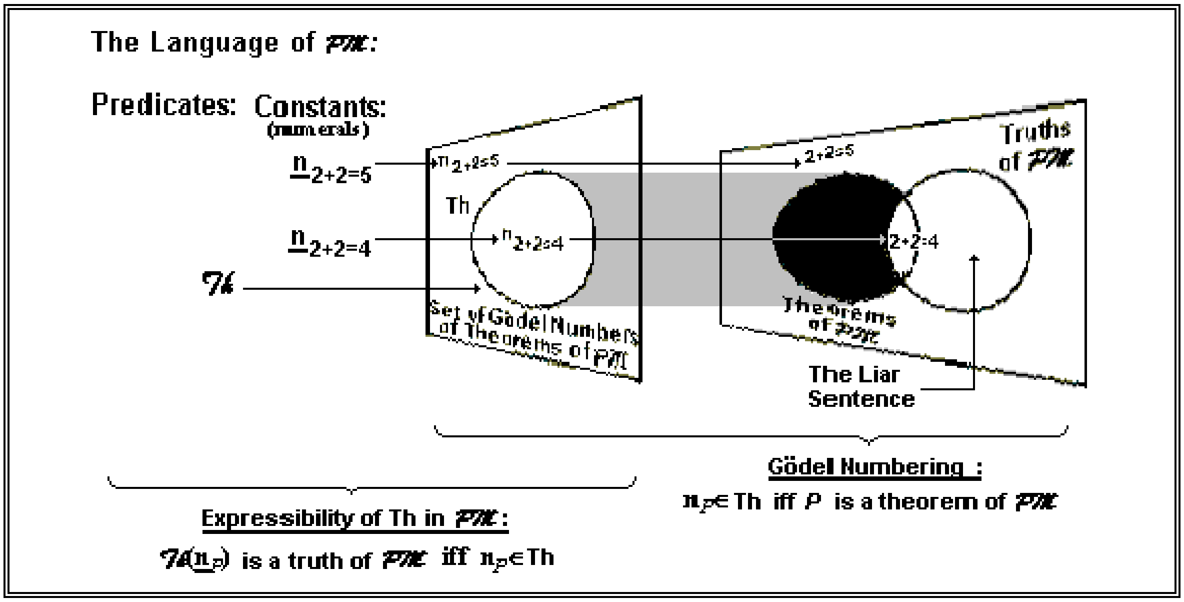
(T) For any P, $\mathfrak{S} \models \mathscr{Th}(\underline{\mathbf{n}}_P) \leftrightarrow P$

Proof of T:

$\mathfrak{S} \models \mathscr{Th}(\underline{\mathbf{n}}_P)$	iff	$\mathbf{n}_P \in \mathbf{Th}$	By the Expressibility of **Th**
	iff	$\vdash_{PM} P$	By the definition of **Th**
	iff	$\mathfrak{S} \models P$	By Completeness

Hence $\mathfrak{S} \models \mathscr{Th}(\underline{\mathbf{n}}_P) \leftrightarrow P$. Hence T is true. But by Tarski's Theorem, T cannot be true. Hence the original assumption of completeness must be false. **QED**.

The true situation drawn upon by the proof is diagrammed below. The picture represents the expressibility of **Th**, the arithmetization of syntax (Gödel numbering), and Tarski's theorem. It also illustrates the non-provability of the Liar Sentence, which is proven in the supplementary section below.



G. Supplementary Material: Gödel Original Proof Strategy

In his original work Gödel offers a somewhat more complex proof and establishes a strong version of the incompleteness theorem. The version presented above makes use of Tarski's Theorem. It is the simplest and clearest way to prove the theorem. In proving Tarski's Theorem, however, we make the implicit assumption that arithmetic is itself true.[29] Gödel's original version makes a weaker assumption that ***PM*** is in a sense consistent. In this section we shall review the highlights of Gödel's original proof strategy.

Definition

If **R** is an k-place relation on the natural numbers, then **R** is said to be ***provable*** in ***PM*** iff there is some sentence $P(x_1,...,x_k)$ of $\mathbf{L}_{\boldsymbol{PM}}$ (containing only $x_1,...,x_k$ as terms) such that:

$$\vdash_{PM} P(\underline{\mathbf{n}}_1,...,\underline{\mathbf{n}}_k) \text{ iff } \mathbf{n}_1,...,\mathbf{n}_k \text{ stand in relation } \mathbf{R} \text{ in arithmetic.}$$

Let **C** be a subset of the natural numbers. **C** is said to be ***provable*** in ***PM*** if there is some sentence $P(x)$ of $\mathbf{L}_{\boldsymbol{PM}}$ (containing only the term x) such that, for any **n**:

$$\vdash_{PM} P(\underline{\mathbf{n}}) \text{ iff } \mathbf{n}\in\mathbf{C}$$

[29] To see how Tarski's Theorem assumes the truth of arithmetic, let us rephrase the theorem. It says: for any predicate $T(v)$, there is some P such that $\models_{\mathfrak{S}} T(\underline{n}_P)\leftrightarrow P$ is false. By "is false" here we mean that the sentence is false in our world in which the axioms of $\mathcal{PM}$ (arithmetic) are true. Hence the theorem assumes the static perspective of the actual world as the determiner of judgments of truth or falsity of sentences in $\mathbf{L}_{\mathcal{PM}}$.

Theorem. **Provability.** Every recursive function and decidable set is provable.

If **R** is a k-place recursive function, we can find an open sentence $P(x_1,...,x_{1+k})$ of $\mathbf{L}_{\boldsymbol{PM}}$ that defines a primitive recursive function and

define a predicate $\boldsymbol{Prd}(x_1,...,x_{1+k})$ to mean $P(x_1,...,x_{1+k})$ such that for any $\mathbf{n}_1,...,\mathbf{n}_k,\mathbf{m}$,

$$\text{if } \mathbf{R}(\mathbf{n}_1,...,\mathbf{n}_k)=\mathbf{m} \text{ in arithmetic, then } \vdash_{\boldsymbol{PM}} \boldsymbol{Prd}(\underline{\mathbf{n}}_1,...,\underline{\mathbf{n}}_k,\underline{\mathbf{m}})$$

Proof Strategy. The proof proceeds in two stages. First Gödel shows that basic recursive facts are provable. That is, he shows we can prove any relational sentence in which the relational predicate is one that expresses a basic recursive function and in which the subject terms are all numerals for specific numbers. The second step consists of showing a complex conditional: if it is possible to prove such facts about the recursive functions used to define a more complex recursive function, then it is possible to prove such facts for the complex recursive function so defined. The proof itself, which we shall not attempt here, is long and complex, embodying much of the detail and journeyman's logic of Gödel's original publication.

It is important for our over-view that we understand what the theorem says. It assures us that so long as we are talking just about specific numbers and asserting facts about whether they obey recursive functions or fall in decidable sets, there is some way to express that proposition and prove it in ***PM***. When combined with the soundness theorem, the theorem amounts to a sort of partial completeness. Recall that an axiom system is ***sound*** if its axioms are true and its rules are valid (preserve truth). First we state the Soundness Theorem and then combine it with the previous result to state partial completeness.

Theorem. Soundness of *PM*. For any P in $\mathbf{L}_{\boldsymbol{PM}}$,

$$\text{If } \vdash_{\boldsymbol{PM}} P, \text{ then } \mathfrak{S} \models P$$

Proof Sketch. The proof strategy is straightforward. Gödel assumes that all the axioms of ***PM*** are true. It is easy to show (by truth-tables) that *modus ponens* is valid, i.e. that if premises of *modus ponens* are true, then the conclusion is also true. Since all the theorems are proven from the axioms (which are assumed to be true) by *modus ponens* (which preserves truth), it follows that all the theorems are true.

We now combine the two previous theorems.

Corollary. Partial Completeness of *PM*. If **R** is a recursive function, we can find an open sentence $P(x_1,...,x_{1+k})$ of $\mathbf{L}_{\boldsymbol{PM}}$ and define a predicate $\boldsymbol{Prd}(x_1,...,x_{1+k})$ to mean $P(x_1,...,x_{1+k})$ such that, for any $\mathbf{n}_1,...,\mathbf{n}_k,\mathbf{m}$,

$$\mathbf{R}\ (\mathbf{n}_1,...,\mathbf{n}_k)=\mathrm{m} \text{ in arithmetic iff } \vdash_{\boldsymbol{PM}} \boldsymbol{Prd}(\underline{\mathbf{n}}_1,...,\underline{\mathbf{n}}_k,\underline{\mathbf{m}})$$

The part of ***PM*** that is complete consists of those theorems that use numerals and predicates that express recursive functions.

Let us now apply these results to the case of the relation **Ded** which is decidable and to the set **Th** which is not.

Corollary. Provability of *Deducibility in* Theorem.
The relation **Ded** is provable:
there is some sentence $\mathscr{Ded}(x,y)$ of $\mathbf{L}_{\boldsymbol{PM}}$ such that, for any $\mathbf{n}_\mathrm{D},\mathbf{n}_P$:

$$\mathbf{Ded}(\mathbf{n}_\mathrm{D},\mathbf{n}_P) \text{ iff } \vdash_{\boldsymbol{PM}} \mathscr{Ded}(\underline{\mathbf{n}}_\mathrm{D}, \underline{\mathbf{n}}_P)$$

Proof. The provability of **Ded** follows from the earlier theorems that establish that **Ded** is decidable and that every decidable set is provable. **QED**

Lemma. Provability is provable (P is provable iff it is provable that it is provable). For the predicate $\mathscr{Th}$ that expresses the set $\mathbf{Th} = \{\mathbf{n}_P| \vdash_{\boldsymbol{PM}} P\}$:

$$\vdash_{\boldsymbol{PM}} \mathscr{Th}(\underline{\mathbf{n}}_P\ ) \text{ iff } \vdash_{\boldsymbol{PM}} P$$

and

$$\vdash_{\boldsymbol{PM}} {\sim}\mathscr{Th}(\underline{\mathbf{n}}_P\ ) \text{ iff } \vdash_{\boldsymbol{PM}} {\sim}P$$

Proof Sketch.

Assume first that $\vdash_{\boldsymbol{PM}} \mathscr{Th}(\underline{\mathbf{n}}_P\ )$. Then by soundness $\mathscr{Th}(\underline{\mathbf{n}}_P)$ is a truth of arithmetic. Since $\mathscr{Th}(\underline{\mathbf{n}}_P\ )$ expresses membership in **Th**, $\mathbf{n}_P\in\mathbf{Th}$. Hence by the definition of **Th**, it follows that $\vdash_{\boldsymbol{PM}} P$.

Assume next that $\vdash_{\boldsymbol{PM}} P$. Then there is some deduction D of P such that $\mathbf{Ded}(\mathbf{n}_\mathrm{D},\mathbf{n}_P)$. By the provability of **Ded** for the predicate $\mathscr{Ded}$ which expresses it, $\vdash_{\boldsymbol{PM}} \mathscr{Ded}(\underline{\mathbf{n}}_\mathrm{D},\underline{\mathbf{n}}_P)$. Then by logical inference in ***PM*** (namely an existential generalization from $\mathscr{Ded}(\underline{\mathbf{n}}_\mathrm{D},\underline{\mathbf{n}}_P)$)we obtain $\vdash_{\boldsymbol{PM}}(\exists x)\mathscr{Ded}(x, \underline{\mathbf{n}}_P)$. But $(\exists x)\mathscr{Ded}(x, \underline{\mathbf{n}}_P)$ is by definition the same as $\mathscr{Th}(\underline{\mathbf{n}}_P)$. Hence $\vdash_{\boldsymbol{PM}} \mathscr{Th}(\underline{\mathbf{n}}_P)$.

The negated version of the theorem follows by the logic of negation. **QED.**

Corollary. Provability of *Theoremhood* in *PM*.

$\vdash_{PM} \mathscr{Th}(\underline{\mathbf{n}}_P)$ iff $\vdash_{PM} P$ iff $\mathbf{n}_P \in \mathbf{Th}$ iff $\mathfrak{S} \models \mathscr{Th}(\underline{\mathbf{n}}_P)$

Proof

$\vdash_{PM} \mathscr{Th}(\underline{\mathbf{n}}_P)$ iff $\vdash_{PM} P$ from a previous theorem

iff $\mathbf{n}_P \in \mathbf{Th}$ by the def of **Th**

iff $\mathfrak{S} \models \mathscr{Th}(\underline{\mathbf{n}}_P)$ by the Expressibility of **Th**

QED

Corollary on Deducibility. Not $\vdash_{PM} P$ iff, for all $\mathbf{n}_D$ in **Nn**, $\vdash_{PM} \sim \mathscr{Ded}(\underline{\mathbf{n}}_D,\underline{\mathbf{n}}_P)$

Equivalently, $\vdash_{PM} P$ iff, for some $\mathbf{n}_D$ in **Nn**, not $\vdash_{PM} \sim \mathscr{Ded}(\underline{\mathbf{n}}_D,\underline{\mathbf{n}}_P)$

Proof

(1) $\vdash_{PM} P$ iff $\vdash_{PM} (\exists x)\mathscr{Ded}(x, \underline{\mathbf{n}}_P)$ previous proof

(2) iff for some $\underline{\mathbf{n}}_D$ in **Nn**, **Ded**($\mathbf{n}_D$,$\mathbf{n}_P$) Soundness

(3) not $\vdash_{PM} P$ iff for all $\underline{\mathbf{n}}_D$ in **Nn**, not **Ded**($\mathbf{n}_D$,$\mathbf{n}_P$) 2 and negation

(4) iff for all $\underline{\mathbf{n}}_D$ in **Nn**, $\mathfrak{S} \models \sim\mathscr{Ded}(\underline{\mathbf{n}}_D,\underline{\mathbf{n}}_P)$ meaning of ~

QED

Review of the Liar Sentence J. Consider the open sentence $\sim\mathscr{Th}(x)$. We know by the self-predication theorem, that there is some sentence J, namely,

(J) $\sim\mathscr{Th}(\underline{\mathbf{n}}_J)$

such that there is a truth of arithmetic:

(I) $\mathfrak{S} \models \underline{\mathbf{n}}_J = \underline{\mathbf{n}}_{\sim\mathscr{Th}(\underline{\mathbf{n}}_J)}$

Moreover, since identity is a decidable relation when asserted between numerals, I insures:

(PI) $\vdash_{PM} \underline{\mathbf{n}}_J = \underline{\mathbf{n}}_{\sim\mathscr{Th}(\underline{\mathbf{n}}_J)}$

It is necessary now to appeal to various concepts of consistency. The core idea of syntactic consistency was defined in the previous lecture, and we restate the definition below. Let ***S*** stand for an arbitrary axiom system, and let *P* stand for a sentence in the syntax of ***S***.

Definitions

S is (***syntactically***) ***consistent*** iff for some *P*, not $\vdash_{\boldsymbol{S}} P$

(Equivalently, ***S*** is consistent iff not for some *P*, $\vdash_{\boldsymbol{S}} P{\wedge}{\sim}P$)

S is (***syntactically***) ***inconsistent*** iff for all *P*, $\vdash_{\boldsymbol{S}} P$

(Equivalently, ***S*** is consistent iff for all *P*, not $\vdash_{\boldsymbol{S}} P{\wedge}{\sim}P$)

Theorem. If $\vdash_{\boldsymbol{PM}}$ is consistent, then not $\vdash_{\boldsymbol{PM}} {\sim}\mathscr{Th}(\underline{\mathbf{n}}_J)$

Proof

(1) ***PM*** is consistent — hypothesis of the theorem

(2) $\vdash_{\boldsymbol{PM}} {\sim}\mathscr{Th}(\underline{\mathbf{n}}_J)$ — hypothesis for a *reductio*

(3) $\vdash_{\boldsymbol{PM}} {\sim}\mathscr{Th}(\underline{\mathbf{n}}_{\sim\mathscr{Th}(\underline{\mathbf{n}}_J)})$ — PI and the substitutivity of =

(4) $\vdash_{\boldsymbol{PM}} {\sim}{\sim}\mathscr{Th}(\underline{\mathbf{n}}_J)$ — by the provability of **Th** earlier

(5) $\vdash_{\boldsymbol{PM}} \mathscr{Th}(\underline{\mathbf{n}}_J)$ — by double negation

(6) not $\vdash_{\boldsymbol{PM}} {\sim}\mathscr{Th}(\underline{\mathbf{n}}_J)$ — since (2)-(5) are contradictory

QED

A second and weaker notion of consistency requires that within the axiom system ***S*** the theorems related to the existential quantifier are consistent, though there may be an inconsistency elsewhere: no sentence beginning with ∃ (*for some*) is contradicted by theorems that assert specific instances. That is, if it is provable that a property holds of some number, then it should not be provable of every number individually that it does not have that property. This notion too is syntactic since it talks about what is provable and proof is a matter of syntax. It is customary to call this idea ω-***consistency***, because ω (the Greek letter *omega*) is another name in mathematics for the set of natural numbers **Nn**.

S is ω-***consistent*** iff for any open sentence *P*(*x*),

if $\vdash_{\boldsymbol{S}} \exists xP(x)$, then for some n in **Nn**, not $\vdash_{\boldsymbol{S}} {\sim}P(\underline{\mathbf{n}})$

S is ω-***inconsistent*** iff there is some open sentence *P*(*x*),

for all n in **Nn**, $\vdash_{\boldsymbol{S}} {\sim}P(\underline{\mathbf{n}})$, but $\vdash_{\boldsymbol{S}} \exists xP(x)$.

The weaker notion of syntactic consistency entails the stronger.

Theorem. If ***S*** is inconsistent, then ***S*** is ω-inconsistent.

Proof. If ***S*** is not consistent then every sentence is provable, all the sentences required for ω-inconsistency are provable. **QED**

Corollary. If ***S*** is ω-consistent, then ***S*** is consistent.

Theorem. If $\vdash_{\boldsymbol{PM}}$ is ω-consistent, then not $\vdash_{\boldsymbol{PM}} \mathscr{Th}(\underline{\mathbf{n}}_J)$

Proof

(1) ***PM*** is ω-consistent — hypothesis of the theorem

(2) $\vdash_{\boldsymbol{PM}} \mathscr{Th}(\underline{\mathbf{n}}_J)$ — hypothesis for a *reductio*

(3) $\vdash_{\boldsymbol{PM}} \exists x\mathscr{Ded}(x, \underline{\mathbf{n}}_J)$ — by the definition of $\mathscr{Th}$

(4) $\mathfrak{I} \models \exists x\mathscr{Ded}(x, \underline{\mathbf{n}}_J)$ — by soundness

(5) for some $\mathbf{n}_D$ in **Nn**, $\mathbf{Ded}(\mathbf{n}_D,\mathbf{n}_P)$ — meaning of $\exists$ (def of truth)

(6) $\mathbf{Ded}(\mathbf{n}_D,\mathbf{n}_P)$ — existential instantiation

(7) $\vdash_{\boldsymbol{PM}} \mathscr{Ded}(\underline{\mathbf{n}}_D,\underline{\mathbf{n}}_J)$ — partial completeness

(8) $\vdash_{\boldsymbol{PM}} \exists x\mathscr{Ded}(\underline{\mathbf{n}}_D,\underline{\mathbf{n}}_J)$ — existential generalization in ***PM***

(9) ***PM*** is consistent — 1 and a previous theorem

(10) not $\vdash_{\boldsymbol{PM}} \sim\mathscr{Th}(\underline{\mathbf{n}}_{\sim\mathscr{Th}(\underline{\mathbf{n}}_J)})$ — the previous theorem

(11) for all $\mathbf{n}_D$ in **Nn**, $\vdash_{\boldsymbol{PM}} \sim \mathscr{Ded}(\underline{\mathbf{n}}_D, \underline{\mathbf{n}}_{\sim\mathscr{Th}(\underline{\mathbf{n}}_J)})$ — corollary on Deducibility

(12) for all $\mathbf{n}_D$ in **Nn**, $\vdash_{\boldsymbol{PM}} \sim \mathscr{Ded}(\underline{\mathbf{n}}_D,\underline{\mathbf{n}}_J)$ — ID and substitutability of =

QED

Theorem. $\mathfrak{I} \models \sim\mathscr{Th}(\underline{\mathbf{n}}_J)$

Proof

(1) not $\vdash_{\boldsymbol{PM}} \sim\mathscr{Th}(\underline{\mathbf{n}}_J)$ — hypothesis for *reductio*

(2) for all $\mathbf{n}_D$ in **Nn**, $\vdash_{\boldsymbol{PM}} \sim\mathscr{Ded}(\underline{\mathbf{n}}_D, \underline{\mathbf{n}}_{\sim\mathscr{Th}(\underline{\mathbf{n}}_J)})$ — lemma on Deductibility

(3) for all $\mathbf{n}_D$ in **Nn**, $\vdash_{\boldsymbol{PM}} \sim\mathscr{Ded}(\underline{\mathbf{n}}_D,\underline{\mathbf{n}}_J)$ — ID and substitutivity of =

(4) for all $\mathbf{n}_D$ in **Nn**, $\mathfrak{I} \models \sim\mathscr{Ded}(\underline{\mathbf{n}}_D,\underline{\mathbf{n}}_J)$ — soundness

(5) $\mathfrak{I} \models \forall x\sim\mathscr{Ded}(x,\underline{\mathbf{n}}_J)$ — meaning of $\forall$, def of truth

(6) $\mathfrak{I} \models \sim\exists x\mathscr{Ded}(x,\underline{\mathbf{n}}_J)$ — quantifier negation

(7) $\mathfrak{I} \models \sim\mathscr{Th}(\underline{\mathbf{n}}_J)$ — definition of $\mathscr{Th}$

QED

Corollary. Incompleteness (Gödel's Original Version).
If ***PM*** is consistent, then ***PM*** is incomplete.

Proof. Since ***PM*** is consistent, we know from a previous theorem that not $\vdash_{\boldsymbol{PM}}\sim\mathscr{Th}(\underline{\mathbf{n}}_J)$. On the other hand, we have just proved that the sentence $\sim\mathscr{Th}(\underline{\mathbf{n}}_J)$ of ***PM*** is true. Therefore, there is some sentence of ***PM*** that is true but not provable in ***PM***. The system is therefore incomplete. **QED**

III. Exercises

A. Skills: Formal Proof in the System F.

Here is an example of a proof in **F.** It established that $\vdash_F A\rightarrow A$.

1. $\vdash_F((A\rightarrow((A\rightarrow A)\rightarrow A))\rightarrow((A\rightarrow(A\rightarrow A))\rightarrow(A\rightarrow A)))$	axiom schema 2
2. $\vdash_F(A\rightarrow((A\rightarrow A)\rightarrow A))$	axiom schema 1
3. $\vdash_F((A\rightarrow(A\rightarrow A))\rightarrow(A\rightarrow A))$	*modus ponens* on 1 and 2
4. $\vdash_F(A\rightarrow(A\rightarrow A))$	axiom schema 1
5. $\vdash_F(A\rightarrow A)$	*modus ponens* on 3 and 4

Both Copi's system of logic for sentential logic and Gentzen's natural deduction system are much easier to use than this axiomatic system. But logicians still understand axiom systems to be the most exact way to state a mathematical or logical theory.

Exercises

1. Give an *informal* proof in Gentzen's system, in a proof that has no assumptions, that $\sim A\rightarrow(A\rightarrow B)$.
2. Give a *formal* proof in the system F that $\vdash_F \sim A\rightarrow(A\rightarrow B)$. Each line should either be an instance of an axiom schema or follow from previous lines by *modus ponens.* (*Hint*. In applying axiom schemata, different letters may be replaced by the same sentence, and these sentences may themselves be complex. It helps to work both forward and backward, to and from sentences that fit the form of *modus ponens.* The proof is not long but it is not obvious.)

B. Skills: Informal Proofs in Naïve Set Theory

i. Naïve Set Theory: The Notion of Implicit Definition.

The motivation for axiomatic set theory is to give the concept of set a kind of "analysis" or "explanation." In the history of philosophy the notion of *property* or *quality* had always been an obscure term. Functionally there had always been a close correlation between property-talk and set- or collection-talk. For example, in Aristotle's metaphysics *Socrates is rational* is necessarily equivalent to *Socrates is a member of the species human being*. Philosophically *property* was never given a plausible explicit definition, nor its conditions for identity spelled out. By using axiomatic methods, however, Cantor and subsequent researches in axiomatic set theory have been very successful at explaining the notion of set, the functional equivalent of property.

In axiom system ideas are "explained" in two ways. The first is in explicit definitions. These consist of a series of abbreviations that supplement the axioms and inference rules of the system and allow new terms to abbreviate longer expressions in the system. Often these abbreviations take the form of biconditionals and serve the function of a traditional definition by necessary and sufficient conditions. For example, in the axiomatized set theory below the notion of union is introduced by definition: $A\cup B$ is an abbreviation for $\{x|x\in A\wedge x\in B\}$. The latter captures the "meaning" of the former.

But not all terms in a system can abbreviate others because the system has to start with some basic vocabulary. This consists of the so-called "primitive terms," the expressions used in writing the axioms themselves. But the axioms in a sense explain these terms. Together with the system's inference the axioms determine all the theorems deducible using primitive terms. If the system is sound and complete, it fully circumscribes the use of the sentences to assert what is true. In this way their "meaning" is explained. "Axiomatizations" of concepts in this way is one of the major methodological differences between logic and mathematics, on the one hand, and traditional philosophy on the other. Though there have been sporadic attempts by philosophers to axiomatize their ideas (*e.g.* Proclus and Spinoza) their attempts were technical failures. Advances in mathematics over the past two hundred years have opened this method to philosophical work. The analysis of *set* as the functional equivalent of *property* is an example of its success.

ii. The Axioms and Definitions of Naive Set Theory

Axioms:

Abstraction: $\forall y(y\in\{x|P[x]\}\leftrightarrow P[y])$ (in its practical form)

Extensionality: $A=B \leftrightarrow \forall x(x\in A \leftrightarrow x\in B)$

or equivalently, $\{x|P[x]\}=\{x|Q[x]\} \leftrightarrow \forall y(y\in\{x|P[x]\}\leftrightarrow y\in\{x|Q[x]\})$

Definitions :			**Technical Name:**
x≠y	*x* is not identical to *y*	$\sim(x=y)$	***non-identity*** or ***inequality***
x∉A	*x* is not an element of set A	$\sim(x\in A)$	***non-membership***
A⊆B	*Everything in* A *is in* B	$\forall x(x\in A\rightarrow x\in B)$	A is a ***subset*** of B
A⊂B	A⊆B *& some* B *is not in* A	$A\subseteq B\wedge\sim A=B$	A is a ***proper subset*** of B
∅ or Λ	*set containing nothing*	$\{x\|\ x\neq x\}$	the ***empty set***
V	*set containing everything*	$\{x\|\ x=x\}$	the ***universal set***
A∩B	*set of things in both* A *and* B	$\{x\|\ x\in A\wedge x\in B\}$	the ***intersection*** of A and B
A∪B	*set of things in either* A *or* B	$\{x\|\ x\in A\vee x\in B\}$	the ***union*** of A and B
A–B	*set of things in A but not in B*	$\{x\|\ x\in A\wedge x\notin B\}$	the ***relative complement*** of B in A
˜ –A or **Error! Bookmark not defined.**Ā	*set of things not in A*	$\{x\|\ x\notin A\}$	the ***complement*** of A
P(A)	*the set of subsets of* A	$\{B\|\ B\subseteq A\}$	the ***power set*** of A

Much of applied metatheory consists of "showing" in the metalanguage that entities are in sets, and students of logic need to be able to use naïve set theory to do this. Proofs in metatheory are informal. There are roughly three sorts of assumptions you can assume in metatheory:

- the informal appeal to the truths and inference rules of system of first-order logic (*e.g.* the rules of Copi or Gentzen summarized earlier),
- the "axioms" and definitions of naïve set theory, and
- obvious facts of arithmetic and syntax.

The problems below consist of proving facts about sets. Your proofs should be informal, using Copi's or Gentzen's rules cited earlier. You must also use the two axioms of naïve set theory and the set definitions above. In applying the axioms and variables be sure to change variables as necessary so you do not state an instance of an axiom or rule that uses the same variable for two or more variables that are kept distinct in the original.

Take special note of the Principle of Abstraction. As the Principle says, an element y is shown to be in the set $\{x|P[x]\}$ by showing it possesses the set's defining property $P[y]$. In practice, however, applying the axiom can be confusing because the defining property is often rather complex. A one-placed (monadic) "property" is expressed by an open sentence $P[y]$ that contains only one free-variable y. This variable may occur in more than one place. But as long as $P[y]$ only contains the one free-variable, it can be a complex sentence, containing various clauses joined by connectives, numerous predicates, constants, quantifiers and other variables bound by the quantifiers that occur someplace within $P[y]$.[30]

Example. Theorem. $A\cap B\subseteq B$.

Proof. Consider an arbitrary x and assume for conditional proof that $x\in A\cap B$. Hence by the definition of intersection, $x\in\{y|\ y\in A\wedge y\in B\}$. [Notice the need here to change variables to avoid confusion.] Hence by the Principle of Abstraction, $x\in A\wedge x\in B$. Hence by simplification ($\wedge$ elimination), $x\in B$. Thus by conditional proof ($\rightarrow$ introduction), it has be proven that $x\in A\cap B\rightarrow x\in B$. Since we have been general in x (*i.e.* since x has be "arbitrary"), we may universally generalize ($\forall$ introduction), getting $\forall x(x\in A\cap B\rightarrow x\in B)$. Thus, by the definition of subset, $A\cap B\subseteq B$. QED.

Exercises. Give informal proofs of the following in first-order logic using the two axioms and the definitions. Once a fact is proven, you may use it as an assumption in later proofs.

3. $A\subseteq A$
4. $A\cap B\subseteq B\cup A$
5. $-A\subseteq-(A\cup B)$
6. $\varnothing\subseteq A$
7. $A\subseteq V$

[30] In Chapter 2 we shall meet 2-place relations, set of "ordered-pairs," defined by formulas with two free-variables, and more generally n-place relations consisting of sets of ordered n-tuples defined by open sentences with n free-variables.

8. $A \subseteq A$
9. $(A \subseteq B \wedge B \subseteq A) \leftrightarrow A=B$
10. $-(A \cup B)=-A \cap -B$
11. $-A=V-A$
12. $A \subseteq \mathbf{P}(A)$

iii. Relations

<u>*Axioms and Definitions.*</u> A generalized and more complete set of axioms and definitions is give below that extend set theory to relations.

Definitions:

$<x,y> = \{x, \{x,y\}\}$

$<x_1,\ldots,x_n,y>= <<x_1,\ldots,x_n,>y>$

Theorems: $<x,y>=<x,y>$ iff $(x=y \;\&\; y=x)$

$<x_1,\ldots,x_n>=<y_1,\ldots,y_n>=$ iff $(x_1= y_1 \;\&\; \ldots \;\&\; x_n= y_n)$

Axioms :

Abstraction:[31] $\forall y_1,\ldots,y_n(<y_1,\ldots,y_n>\in\{<x_1,\ldots,x_n>|P[x_1,\ldots,x_n]\}\leftrightarrow P[y])$ (in its practical form)

Extensionality: $R=R' \leftrightarrow \forall x_1\ldots x_n(<x_1,\ldots,x_n>\in R \leftrightarrow <x_1,\ldots,x_n>\in R')$

An Equivalent Version $\{<x_1,\ldots,x_n>|P[x_1,\ldots,x_n]\}=\{<x_1,\ldots,x_n>|Q[x_1,\ldots,x_n]\} \leftrightarrow \forall x_1\ldots\forall x_n(P[x_1,\ldots,x_n] \leftrightarrow Q[x_1,\ldots,x_n])$

Definitions :

AxB	Cartesian product of A and B	$\{<x,y>\mid x\in A\wedge y\in B\}$
A^2	Cartesian product of A and A	AxA
A^n	Cartesian produce of $A_1,\ldots,A_n$	$A_1x\ldots xA_n$
V^2	The universal (binary) relation	VxV
$\mathbf{P}(V^2)$	the set of 2-place relations	
$\mathbf{P}(V^n)$	the set of *n*-place relations	

R is a ***binary relation***		$R\subseteq V^2$
R is a **n-place relation**		$R\subseteq V^n$
f is a ***1-place function***	*f* is a binary relation and $\forall x\forall y\forall z((<x,y>\in f \wedge <x,z>\in f)\rightarrow y=z)$	
f is a ***n+1-place function***	*f* is an *n*-place relation and $\forall x_1\ldots x_n\forall y\forall z((<x_1,\ldots,x_n,y>\in f \wedge <x_1,\ldots,x_n,z>\in f)\rightarrow y=z)$	

If *f* is a 1-place function, $f(x)=y$ *means* $<x,y>\in f$

If $f(x)=y$, then *x* is an argument of *f* and *y* is a value.

Domain(f) = $\{x\mid \exists y f(x)=y\}$

Range(f) = $\{y\mid \exists y f(x)=y\}$

$f^{-1} = \{<y,x,>\mid f(x)=y)\}$ f^{-1} is called the ***inverse*** of *f*

If *f* is a *n*-place function, $f(x_1,\ldots,x_n)=y\}$ means $<x_1,\ldots,x_n,y>$

If $f(x_1,\ldots,x_n)=y$, then $<x_1,\ldots,x_n>$ is an argument of *f* and *y* is a value.

[31] Technically in set theory this "axiom" follows as a theorem from the Principle of Abstraction as previously given for sets. The proof requires the introduction of order *n*-tuple in terms of sets: $<x,y> = \{x, \{x,y\}\}$ and $<x_1,\ldots,x_n,y>= <<x_1,\ldots,x_n,>y>$.

	Domain(f) = $\{<x_1,\ldots,x_n>\mid \exists y f(x_1,\ldots,x_n)=y\}$ Range(f) = $\{y\mid \exists y f(x_1,\ldots,x_n)=y\}$
$f(A\xrightarrow{\text{into}}B)$	f is a 1-place function, Domain(f), and Range(f)$\subseteq$B
$f(A\xrightarrow{\text{onto}}B)$	f is a 1-place function, Domain(f), and Range(f)=B
$f(A\xrightarrow{\text{1-1 onto}}B)$	f is a 1-place function, Domain(f), Range(f)=B, and f^{-1} is a 1-place function
f is a partial function on A	$\exists C\exists B(C\subset A\ \&\ f(C\xrightarrow{\text{into}}B))$

Exercises. Give informal proofs of the following:

13. $\forall x(x \subseteq V^2$ iff $x\in$ **P**$(V^2))$
14. $A^2 \subseteq V^2$ (Here show: $\forall x\forall y(<x,y>\in A^2$ iff $<x,y>\in V^2$, and appeal to extentionality)
15. $\{<x,y>\mid (x \times x) = y\}$ is a function.
16. $\{<x,y>\mid (x \times x) = y\}^{-1}$ is not a function.

C. Logistic Systems

17. Consistency. In elementary logic you met the distinction between syntactic consistency (leading by a derivation to a contradiction) and satisfiability (true in some structure or model). In which sense of consistency was Reinmannian Geometry shown consistent and Frege's *Grudgesetze* inconsistent? Why?
18. Reduction. In the *Grudgesetze* and again in *Principia Mathematica* the attempt was to "reduce" one theory to another, *i.e.* arithmetic to logic and set theory. In your own words formulate what would be the necessary conditions for such a reduction to be successful. Notice what is meant by "theory" here. (These ideas still grip discussions of reduction in the philosophy of science.)

D. Gödel's Proof: Technical Details

In each of these computations show each step of your work

19. What is the Gödel number of the (ungrammatical) expression: $\sim\in\in$?
20. Express 420 as the product of primes listing the primes in order from least to greatest.
21. What expression is such that 192 is its Gödel number?
    a. Explain the difference between the number two, the numeral 2, and the Gödel number 2,
    b. Explain the difference between the set of proofs in ***PM***, **Ded**, and $\mathscr{Ded}$,
    c. Explain the difference between the set of theorems of ***PM***, **Th**, $\mathscr{Th}$, and the truths of arithmetic.
    d. Gödel makes an assumption that arithmetic has a at least one standard model. Try to characterize how that assumption is made in the proof. Is the assumption plausible?

E. Gödel’s Proof: Theoretical Implications

22. In a short essay summarize for yourself (in terms that you will be able to understand five years from now):
    a. the proposition that Gödel proved,
    b. why it is interesting,
    c. the general strategy of his proof including any features that are clever or interesting.

Chapter 2

First-Order Logic Soundness and Completeness

I. Standard Logical Theory

In Chapter 1 logic was defined as falling into two parts, syntax and semantics. Over the last sixty-five years a standard version of formal logic has evolved. Its more elementary version that explores the logical properties of the sentential connectives is known as ***propositional*** or ***sentential logic***. The fuller account extends to the logic of subject-predicate sentences and quantifiers and is known as ***quantificational***, ***predicate***, and most commonly ***first-order logic***. What is important from our perspective is the fact that the logical properties of arguments written in its syntax have been studied in great detail. Indeed, though less than a century old, first-order logic is often called ***classical logic***. Its axioms systems and deduction rules have become the standard against which all innovations in logic are measured. Most of what makes up the body of “discoveries” that constitutes the science of logic consists of details about the syntax, semantics, and proof theory of first-order logic.

Accordingly, if we state something in first-order logic, its syntax, semantics, and proofs are clear, much clearer than they would be if written in natural languages like English, which are still poorly understood. But what makes first-order logic important is not just its clarity, but its expressive power. Using its restricted resources, it is possible to state most mathematical and scientific theories. When science is written in first-order logic, it is eminently clear, and this clarity has important consequences for the scientific method. If a deduction is offered, or a mathematical computation that plays the role of a traditional “deduction” (as is often the case in computer science), we can tell instantly whether is its formally valid. We thus know how to verify a large number of sentences. Up to certain limits we can even calculate the consequences of hypotheses, and hence what we should look for by way of indirect confirmation when the hypotheses themselves cannot be directly confirmed. We also know what it is for a theory to be inconsistent with itself or with other propositions.

It is for such reasons that first-order logic has become a kind of *lingua franca* used by practicing researches in mathematical fields like logic, mathematics, computer science, and the branches of the empirical sciences that lend themselves to mathematical methods. After this introduction, you will be able to spot scientists who are using logic without telling you. Indeed, the rather barbarous mathematical English in which virtually all math and many science books are written is really a compromised version of first-order syntax. Without explaining what they mean, authors sprinkle their writing with references to variables, constants, equations, proofs, theorems, and use turns of phrase like *for all x*, and *for some y*. This talk is intended to be understandable to people who only know English, but it masks the fact that the researcher is really thinking in first-order logic and conveying his thoughts in a precise way to readers who know it also. When we are finished, you should have some understanding of this esoteric language and an appreciation for why it is used when possible.

A. Grammar

i. Parts of Speech

A Simple Grammar: Sentential Logic. The first step in defining any grammar consists of laying down initial building blocks. These are called the ***atomic*** expressions. Next a finite set of grammar rules is laid down for the construction of longer ***molecular*** expressions. The full set of ***well-formed*** expression is then defined as consisting of all expressions that can be built up from the atoms by the rules. Let us begin by considering the grammar of sentential logic. Let atomic formulas consist of all indexed sentential letters P_i, for any positive integer *i.* There is one grammatical rule for each of the sentential connectives: $\sim,\wedge,\vee,\rightarrow$, and $\leftrightarrow$.[32]

Syntax of Sentential Logic

A ***syntax*** **SL** ***for sentential logic*** consists of the sets $\mathbf{AF_{SL}}$, $\mathbf{R_{SL}}$ and $\mathbf{F_{SL}}$ meeting the following conditions:

$\mathbf{AF_{SL}}$, called the set of ***atomic formulas*** (or ***sentences***) of **SL**, is some subset of the sentence letters: $P_1,\ldots, P_n,\ldots.$

$\mathbf{R_{SL}}$, called the set of ***grammatical rules*** of **SL**, consists of five rules, one for each connective:

$\mathrm{R}_{\sim}$ constructs $\sim x$ from any string x; *i.e.* $\mathrm{R}_{\sim}(x)={\sim}x$

$\mathrm{R}_{\wedge}$ constructs $(x\wedge y)$ from strings x and y; *i.e.* $\mathrm{R}_{\wedge}(x,y)=(x\wedge y)$

$\mathrm{R}_{\vee}$ constructs $(x\vee y)$ from strings x and y; *i.e.* $\mathrm{R}_{\vee}(x,y)=(x\vee y)$

$\mathrm{R}_{\rightarrow}$ constructs $(x\rightarrow y)$ from strings x and y; *i.e.* $\mathrm{R}_{\rightarrow}(x,y)=(x\rightarrow y)$

$\mathrm{R}_{\leftrightarrow}$ constructs $(x\leftrightarrow y)$ from strings x and y; *i.e.* $\mathrm{R}_{\leftrightarrow}(x,y)=(x\leftrightarrow y)$

$\mathbf{F_{SL}}$, the set of (***well-formed***) ***formulas*** (or ***sentences***) of **SL** is defined inductively as follows:

1. **Basis Clause.** All formulas in $\mathbf{AF_{SL}}$ are in $\mathbf{F_{SL}}$.
2. **Inductive Clause.** If P and Q are in $\mathbf{F_{SL}}$, then the results of applying the rules $\mathrm{R}_{\sim},\mathrm{R}_{\wedge},\mathrm{R}_{\vee},\mathrm{R}_{\rightarrow}$, and $\mathrm{R}_{\leftrightarrow}$ to them, namely ${\sim}P$, $(P\wedge Q)$, $(P\vee Q)$, $(P\rightarrow Q)$, $(P\leftrightarrow Q)$, are all in $\mathbf{F_{SL}}$;
3. Nothing is in $\mathbf{F_{SL}}$ except by clauses 1 and 2.

A More Complex Grammar: First-Order Logic. The syntax for first-order logic is a more complex. Though sentential logic has a number of grammar rules, it has only one "part of speech:" all expressions are sentences, either simple or complex. First-order logic goes beyond sentential logic is "parsing" the structure of simple sentences into more detailed grammatical components. The building blocks of simple sentences fall into three broad categories: singular terms, functors, and predicates.

A ***singular term***, broadly speaking, is a word or phrase that stands for a single entity in the universe. But what counts as an "entity in the universe" is a question for philosophers. Indeed, it is a central question in that branch of

[32] Strictly speaking, not all these connectives are needed, because some may be introduced from others by definition. For example, if the grammar has just two rules, one for $\sim$ and one for $\wedge$, then the other connectives may be defined: $P\vee Q$ as ${\sim}({\sim}P\wedge{\sim}Q)$, $P\rightarrow Q$ as ${\sim}P\vee Q$, and $P\leftrightarrow Q$ as $(P\rightarrow Q)\wedge(Q\rightarrow P)$.

metaphysics called ontology. In logic “an entity” is any existing thing that the speaker intends to talk about. In a somewhat circular manner, we tell that the speaker intends that a particular thing exists by the fact that he uses a singular term to stand for it. In this way language is supposed to mirror the world. Corresponding to the entities existing in a world are the singular terms of the language that refer to them. In logic, moreover, “the universe” includes not only people, places and things, but also somewhat controversial mathematical entities like numbers. Whether mathematicians are justified in referring to such “things” is disputed by philosopher of mathematics. Physical theories written in a first-order language likewise usually presuppose non-common sense entities. The laws of physics, for example, talk about forces, times, places, and other exotic particles and entities. First-order logic names and “quantifies over” whatever sort of entities the theory in question presumes to talk about and name. It is these that constitute its “universe” or “domain.”

As in natural language, singular terms in logic fall into three types: constants, variables, and the mathematically important class of terms made up of functor or operation symbols. These correspond roughly to proper names, pronouns, and expressions that make up noun phrases that begin with *the*.

Playing the role of proper names are ***constants***. Syntactically these are lower case letters from the beginning of the alphabet: *a, b, c, d,....* Just as a proper name in English, like *Socrates*, stands for an individual thing, so too *a, b*, and *c* are supposed to stand for things. Some first-order languages, in particular arithmetic, has among its constants the numerals: **1**, **2**, **3**, ... It is assumed that these are names for numbers -- **1** naming the number one, **2** the number two, etc. (The Romans used a different set of names, **I** for one, **II** for two, etc.) These expression are called *constants* because, unlike pronouns, they have a reference that is fixed no matter what expressions they appear within.

The role of pronouns is taken by expressions called ***variables***. These are lowercase letters from the end of the alphabet, *x, y, z,* .... Like pronouns they function as what linguists call ***anaphora***, expression whose referent varies from sentence to sentence depending on some previous term in the speech context. The symbol that plays the role in a sentence of fixing the referent of a variable is called a ***quantifier***. For example, if a variable *x* follows the quantifier *for all*, which called the ***universal quantifier*** and represented symbolically by $\forall$, then *x* stands for every individual entity in the universe. If it follows the expression *for some*, which is called the ***existential quantifier*** and represented by $\exists$, *x* stands for at least one existing thing. How the referent of *x* can vary in this way, sometimes standing for everything and sometimes for at least one thing, is what the semantics of quantifiers explains.

The third type of singular term in logic is a complex phrase roughly analogous to natural language’s noun phrases that begin with *the*. These are composed of two sorts of parts. One is a singular term or series of terms. The other is a new type of expression called a ***functor***. The purpose of the functor is to produce a complex singular term when attached to shorter singular terms. In English, for example, the matrix *the wife of ....* approximates a functor. When it is combined with the singular term *Socrates*, it yields the singular noun phrase *the wife of Socrates*, which stands for an entity, the same one that serves as the referent of the proper name

Xanthippe. Likewise, *the state between* ---- *and* ~~~ combines with a series of two singular terms. When joined with the names *Nevada* and *Colorado*, it produce a long noun phrase *the state between Nevada and Colorado* that is co-referential with the name *Utah*.

Semantically we say that functors stand for ***functions*** or ***operations***. What are functions and operations? Philosophers wonder. They are clearly not normal physical objects. Mathematicians often speak of them as “rules.” These rules take inputs (called ***arguments***) and yield outputs (called ***values***). The rule picked out by *the wife of* takes a man as an argument and yields his spouse as its value. In truth, functors and functions have been inspired by the needs of mathematics. Familiar expressions like **+** and **x** are functors that stand for operations on numbers -- they stand for the functions or “rules” addition and multiplication. Syntactically we combine the functor **+** with the numerals **2** and **5**. to make up a complex singular term **2+5**. Semantically, **+** stands for the addition operation on numbers, **2** for the number two, and **5** for the number five. The complex singular term **2+5** stands for the result of applying the addition rule to the numbers two and five. We know that this is the number seven. We also know that **7** is a numeral that stands for seven. Hence the two singular terms **7** and **2+5**, the former simple and the latter complex, stand for the same entity in the world, the number seven.

In addition to singular terms and functors, there is a third variety of new atomic expression in first-order logic, ***predicates***. These are roughly what logicians since Aristotle have called predicates in subject-predicate propositions. In natural languages like English, predicates take many forms. An important variety consists of the so-called ***one-place*** or ***monadic*** predicates. These combine with a subject term to yield a complete sentence. They include intransitive verbs like ....*runs*, common nouns and adjectives combined with forms of the verb *to be*, as in *....is a human* and ....*is rational*. You may recall from the history of philosophy that the semantics of such predicates generated centuries of debate in the Middle Ages. If predicates of this sort stand for something, that sort of entity is rather odd, and was traditionally called (in the terminology of Abelard) a ***universal***. In the syntactic theories of natural languages as developed by modern linguists, sentences are usually treated as being formed from a noun phrase and a verb phrase. It is the verb phrase that assumes the role of the traditional predicate.

Also included in the set of predicates are so-called ***two*** and ***higher place*** predicates. These are terms that require two or more singular terms to make up a simple sentence. Examples from English include transitive verb forms like ....*loves*---- , verbs-preposition combinations like ....*is hiding under*---, and combinations of comparative adjectives with the verb *to be*, as in ....*is taller than*------. Semantically such a “predicate” is traditionally said to stand for a ***relation***, a category of entity even more mysterious than ordinary universals. (In modern linguistics relational predicates of this sort are treated by verb phrases that incorporate names for the direct object, indirect object, or other names mentioned in the verb phrase.)

In logic predicates are represented by uppercase letters *F,G,H*. etc. Defining their semantic role, however, is more difficult. The universals and relations of traditional philosophy hardly seem clear or precisely enough to be used in serious science. A more adequate substitute is found in set theory. Modern logicians generally say that predicates stand for sets. Sets are held to be much clearer than universals because we have a detailed theory that explains how they work. The two laws of naive set theory (the axiom system **F** of Chapter 1 that captures Russell's rendering Frege's ideas on sets) are repeated below with their English translations.

The Axioms of Naïve Set Theory

The Principle of Abstraction. Any set that can be defined exists, containing all and only the objects meeting the definition:

$$\exists A\forall x(x\in A \leftrightarrow P[x])$$

The Axiom of Extensionality. Two sets are identical if, and only if, they have exactly the same members:

$$\forall x\forall y[x=y\leftrightarrow\forall z(z\in x\leftrightarrow z\in y)]$$

The Principle of Abstraction provides an existence criterion for sets, and explains their "composition." The Axiom of Extensionality provides a criterion for their identity. From the two an entire theory may be deduced filling out the properties of sets. This theory is then a kind of "science" that "explains" sets. Sets are, in this sense, quite well explained.[33] It is for this reason that philosophers like Quine thinks that it is a philosophical advancement to employ talk of sets in place of what philosophers have called properties, which obey no law like the Principle of Abstraction explaining when they apply nor any like the Principle of Extensionality setting out their "identity conditions".[34]

Moreover, both relations and functions can be explained as special sorts of sets. Recall that in high school algebra functions are associated with their solution sets. In set theory both relations and functions are identified with sets. A two-place relation is treaded as a set of order-pairs. For example, the relation *x loves y* is represented by the set of all pairs <*x,y*> that make the sentence "*x loves y*" is true. This set is represented symbolically as {<*x,y*>| *x loves y*}, read "the set of all ordered-pairs <*x,y*> such that *x loves y.*" Thus, <Romeo,Julliet>, <Julliet,Romeo>, <Helen, Paris>, <Paris,Helen>, <Menelaus,Helen>} are all in this set, but <Helen,Menelaus> is not – hence the Trojan War. The three place relation *x is between y and z* is similarly treated as a set of ordered triples. Thus, <Cincinnati, Lexington, Dayton> and <Cincinnati,Dayton,Lexington> are in this set but <Dayton,Lexington,Cincinnati> is not. An *n*-place relation is in general defined as a set of ordered *n*-tuples <$a_1,...,a_n$> that stand to one another in the manner dictated by the relation in question. An *n*-place function is defined as a set of ordered *n*+1-place "tuples" <$a_1,...,a_n$,b> such that the functional "rule" assigns to the "input values" $a_1,...,a_n$ (called an ***argument*** of the function), the output value b (called ***the value of the function*** for arguments $a_1,...,a_n$).

[33] This claim is greatly overstated and must be qualified because of the important fact that the axiom system of naive theory is formulated above it entails a contradiction. Rather sophisticated reformulations of the system are necessary to avoid contradictions, and it is these more complex systens that properly "explain" what sets are. This branch of logic is called ***axiomatic set theory***.

[34] See W.V.O. Quine, *Set Theory and Its Logic.*

We are ready now to see the statement of the full grammar for FOL.

The Syntax of First-Order Logic

A ***syntax for first-order language*** **FOL** consists of a series of sets ("parts of speech") $\mathbf{Vbls_{FOL}}$, $\mathbf{Cons_{FOL}}$, $\mathbf{Funcs_{FOL}}$, $\mathbf{Preds_{FOL}}$, $\mathbf{Trms_{FOL}}$, $\mathbf{AF_{FOL}}$ and $\mathbf{F_{FOL}}$, and the rule set $\mathbf{R_{FOL}}$ meeting the following conditions:

We stipulate that the following sets exist, called sets of ***atomic*** (or ***basic***) ***expressions***:

the (infinite) set of ***variables***: $\mathbf{Vbls_{FOL}}=\{v_1,...,v_n,...\}$;

some set $\mathbf{Cons_{FOL}}$ of ***constants*** (***proper names***) drawn from (*i.e.* subset of): $\{c_1,...,c_n,...\}$;

some set $\mathbf{Preds_{FOL}}$ of ***predicates*** drawn from: $\{P^0{}_1,...,P^0{}_n,...;P^1{}_1,...,P^1{}_n,...;\ldots;P^m{}_1,...,P^m{}_n,...;\ldots\}$; it is stipulated that $P^0{}_1$ is $\perp$, called the ***contradiction symbol*** (it's meaning is explained below);

some set $\mathbf{Funcs_{FOL}}$ of ***functors*** drawn from: $\{f^1{}_1,...,f^1{}_n,...;\ldots;f^m{}_1,...,f^m{}_n,...;\ldots\}$.

The set $\mathbf{R_{FOL}}$ of ***grammatical rules*** includes $R_{\sim}$,$R_{\wedge}$,$R_{\vee}$,$R_{\rightarrow}$, and $R_{\leftrightarrow}$ from sentential syntax and three new rules:

R_{AF} takes a symbol x and a string of n items $y_1...y_n$ and makes up the string $xy_1...y_n$:

i.e. $R_{AF}(x,y_{1,},...,y_n)= xy_1...y_n$;

R_{Func} takes a symbol x and a string of n items $y_1...y_n$ and makes up the string $x(y_1...y_n)$;

i.e. $R_{Func}(x,y_{1,}...,y_n)= x(y_1...y_n)$;

$R_{\forall}$ takes the sign x and string y and makes up the string $\forall xy$; *i.e.* $R_{\forall}(x,y)=\forall xy$.

The set $\mathbf{Trms_{FOL}}$ of ***terms*** is defined inductively:

1. **Basis Clause.** All constants and variables are terms (i.e. $\mathbf{Vbls_{FOL}}$ and $\mathbf{Cons_{FOL}}$ are subsets $\mathbf{Trms_{FOL}}$).
2. **Inductive Clause.** If $t_1,...,t_n$ are terms and f^n is a functor, then the result $f^n(t_1...t_n)$ made up by apply to them the rule R_{Func} is a term.
3. Nothing else is a term.

The set $\mathbf{AF_{FOL}}$ of ***atomic formulas*** of **FOL** generated by $\mathbf{Trms_{FOL}}$, $\mathbf{Preds_{FoL}}$, and $\mathbf{Funcs_{FOL}}$ is the set of all $P^n{}_mt_1...t_n$ made up by applying the rule R_{AF} to a predicate $P^n{}_m$ and the string of terms $t_1,...,t_n$.

The set $\mathbf{F_{FOL}}$ of ***formulas*** (also called ***the language***) generated by $\mathbf{Trms_{FOL}}$, $\mathbf{Preds_{FoL}}$ and $\mathbf{Funcs_{FOL}}$ is inductively defined:

1. **Basis Clause.** If P is in $\mathbf{AF_{FOL}}$ then P is in $\mathbf{F_{FOL}}$ (i.e. $\mathbf{AF_{FOL}}$ is a subset of $\mathbf{F_{FOL}}$).
2. **Inductive Clause.** If P and Q are in $\mathbf{F_{FOL}}$, and v is a variable, then the strings $\sim P$, $(P\wedge Q)$, $(P\vee Q)$, $(P\rightarrow Q)$, $(P\leftrightarrow Q)$, and $\forall vP$ that result from applying to them the rules $R_{\sim}$,$R_{\wedge}$,$R_{\vee}$,$R_{\rightarrow}$, $R_{\leftrightarrow}$, and $R_{\forall}$ are all in $\mathbf{F_{FOL}}$.
3. Nothing else is in $\mathbf{F_{FOL}}$.

ii. Syntactic Conventions and Abbreviations

We shall follow the customary conventions of using special typefaces to indicate parts of speech.

Typographic Conventions for Part or Speech in First-Order Syntax

The following expressions, with and without subscripts, ranges over the part of speech indicated:

Typeface	**Part of Speech**
a,b,c	constants
v,w,x, and *y*	variables
P^n	*n*-place predicates
f^n	*n*-place functors
P, Q, and *R*	formulas
X,Y and *Z*	sets of formulas

Since we shall have occasion to refer in some detail to expressions written in first-order notation, let us pause here to define some terms.

Basic Expressions. Linguistics believe that in natural languages the set of basic expressions is finite. In logic, however, we abstract away from natural language – it is always better in mathematics to be as abstract as the subject matter allows – and define the basic sets of constants, predicates, and functors in a way that permits them to be countably infinite. The set of variables is even required to be infinite. Though it is infinite, we construct the set from a finite number of more basic signs. We do this by understanding a subscript or superscript to be, strictly speaking, a series of *n* vertical strokes. Thus constant c_5 is really *c*|||||, v_9 is *v*|||||||||, P^2_4 is (*P*||)|||| and f^3_7 is (*f*|||)|||||||.

Example. The infinite set C={c_1,...,c_n,...}, which contains the expressions from which the constants of a syntax are taken, may be constructed (*i.e.* defined inductively) by concatenation from just the two symbols *c* and | as follows:

1. *c* is in C;
2. if *x* is in C, then *x*| (*i.e.* the result of concatenating *x* to the left of |) is also in C;
3. nothing else is in C.

In a similar way the basic sets of variables, predicates, and functors can be constructed.

Predicates. We read the super- and sub-scripts of P^n_m as indicating that it is the m-th predicate of degree *n*. Let us first explain what ***degrees*** are.

Predicates of degree 1, namely P^1_1,...,P^1_m,...(called ***one-place predicates***), are intended to function like the common nouns, adjectives and intransitive verbs of natural language. They stand for sets.

Predicates of degree 2 (also called ***two-place predicates***), namely P^2_1,...,P^2_m,..., function like transitive verbs or comparative adjectives. They stand for binary relations, which in set theory are sets of ordered-pairs.

More generally, $P^n{}_1,...,P^n{}_m,...$ are *n*-place predicates standing for *n*-place relations, which in set theory are sets of ordered n-tuples <d$_1$,...,d$_n$> of elements of the "domain."

There is also the special "degenerate" case predicates of "degree 0:" $P^0{}_1,...,P^0{}_m,...$ These are followed by 0 terms. They are intended to look and function semantically just like the atomic sentences letters of sentential logic. Their inclusion provides a simple way to replicate within the syntax and semantics of first-order logic the standard syntax and semantics of sentential logic. They insure that in fact the set $\mathbf{F}_{\mathbf{SL}}$ of formulas in sentence logic is a subset of the set $\mathbf{F}_{\mathbf{FOL}}$ of formulas in first-order logic. In this way sentential logic proves to be a special case of first-order logic in a very literally sense.

Of these predicate letters of 0-degree, a.k.a. atomic sentences, the very first one $P^0{}_1$ is singled out for special attention. We stipulated that it is $\perp$ and call it the ***contradiction symbol***. Later in the semantics we will require that it is always false. The symbol is useful in stating logical rules like reduction to the absurd.

Functors. We use the notation $f^n{}_m$ for the *m*-th *n*-place functor. This is the *m*-th functor that stands for an *n*-place function. Such symbols mainly occur when the language is being used to express mathematics.

Connectives. A formula is called a ***truth-function*** if it does not contain the quantifier ∀ or ∃, and is made up from atomic formulas by the connectives ~,∧,∨,→, and ↔. (A truth-function may contain variables.) An atomic sentence or its negation is called a ***literal***.

Quantifiers. In the primitive syntax as defined above there is only one quantifier, the universal quantifier ∀ (read *for all*). The existential quantifier ∃ (*for some*) is then introduced by definition.

Definition of the Existential Quantifier

$\exists vP$ is an abbreviation for $\sim\forall v\sim P$

iii. Substitution

Substitution depends on several preliminary ideas.

Preliminary Definitions

- An occurrence of a term t in P said to be ***free*** in P if
  If t is a variable v then its occurrence is not part of some formula $\forall vQ$ or $\exists vQ$ in P, or
  If t contains an occurrence of a variable v, then that occurrence of v is not part of some formula $\forall vQ$ or $\exists vQ$ in P;
  otherwise the occurrence of t is said to be ***bound***.
- A term or literal that does not contain a variable is said to be ***grounded***.
- An occurrence of a term t' is ***free for (replacement by) a term*** t (which may be complex and contain variables) in P iff, for any variable v that is in t, the occurrence of t' in P is not part of some formula $\forall vQ$ or $\exists vQ$ in P.
- A formula without free variables is customarily called a ***sentence***.
- A formula P is called ***general*** if all its quantifiers occur on the outside (leftmost side) of P in the sense that P is some $\mathrm{E}_1v_1...\mathrm{E}_1v_nQ$ such that $\{\mathrm{E}_1,...,\mathrm{E}_n\}\subseteq\{\forall,\exists\}$ and Q is some truth-functional formula in which neither $\forall$ nor $\exists$ occur. Such a general formula $\mathrm{E}_1v_1...\mathrm{E}_nv_nQ$ is said to be ***universal*** if all E_i are $\forall$.

The Substitution Function

A ***substitution function for terms*** in $\mathbf{F}_{\mathbf{FOL}}$ is a partial function σ from $\mathbf{Trms}_{\mathbf{FOL}}$ (*i.e.* σ̣ need not be defined for all arguments in its domain $\mathbf{Trms}_{\mathbf{FOL}}$) into some subset of $\mathbf{Trms}_{\mathbf{FOL}}$ and is defined recursively, or as we shall usually say, inductively as follows:

Atomic Case. If t is some variable v or constant c, then $\sigma(t)$, if defined, is in $\mathbf{Trms}_{\mathbf{FOL}}$,
Molecular Case. If t is a complex term $f^n(t_1...t_n)$ and $\sigma\ (t_1)...\ \sigma(t_n)$ are all defined, then $\underset{.}{\sigma}(f^n(t_1...t_n\ ))=f^n(\sigma(t_1)...\ \sigma(t_n))$. If any of $\sigma(t_1)...\ \sigma(t_n)$ are undefined, so is $\underset{.}{\sigma}(f^n(t_1...t_n\ ))$ (hence making σ a partial function).

Examples

Intuitively, t' is free for t in P means it is OK to replace t' by t because doing so will not change the over all pattern of bound and free variables in a way that would make the new variable bound where the one it replaces was free.

- The occurrence $f(x,y)$ is free for $g(x,z)$ in $Ff(x,y)$ because the occurrence of $f(x,y)$ in $Ff(x,y)$ is not part of any $\forall xQ$ or $\exists xQ$, nor part of any $\forall zQ$ or $\exists zQ$ in $Ff(x,y)$, where x and z are the variables in $g(x,z)$. Here the variables of $g(x,z)$ remain free in the new formula.
- The occurrence $f(x,y)$ is not free for $g(x,z)$ in $\forall \mathrm{z}(\mathrm{F}f(x,y)\rightarrow Gz)$ because the occurrence of $f(x,y)$ in $\forall \mathrm{z}(\mathrm{F}f(x,y)\rightarrow Gz)$ is part of some $\forall zQ$ or $\exists zQ$ in $\forall \mathrm{z}(\mathrm{Ff}(x,y)\rightarrow Gz)$. If $g(x,z)$ where to replace $f(x,y)$ in $\forall \mathrm{z}(\mathrm{F}f(x,y)\rightarrow Gz)$ its variable z would be bound, the result being $\forall \mathrm{z}(\mathrm{F}g(x,z)\rightarrow Gz)$, whereas all the variables of the term being replaced, namely $f(x,y)$, are free in $\forall \mathrm{z}(\mathrm{F}f(x,y)\rightarrow Gz$

The substitution function is extended to formulas in three ways.

Substitution for *All Free* Terms

The extension of a substitution function σ to sentences, for all free terms, is defined inductively as follows:

Atomic Case. $(P^nt_1...t_n)_\sigma = P^n\sigma(t_1)..\sigma(t_n)$

Molecular Case. $(\sim P)_\sigma=\sim(P_\sigma)$

$(P\wedge Q)_\sigma= (P_\sigma.\wedge(Q_\sigma)$

$(P\vee Q)_\sigma= (P_\sigma \vee(Q_\sigma)$

$(P\rightarrow Q)_\sigma= (P_\sigma.\rightarrow(Q_\sigma)$

$(\forall vP)_\sigma=\forall v(P_\sigma)$ if both the terms replacing are free in the result and the terms being replaced are free in the original,

where these conditions are defined as follows: if σ(*t*) is defined, then

the replaced term t is free in ($\forall vP$), i.e.:

for any *v′* in *t*, *v′* is free in ($\forall vP$), and

the replaced terms t are free for the replacing term σ(*t*), i.e.:

no variable in a replacing term becomes bound:

for any variable *v* that is in *t*, the occurrence of σ(*t*) in *P* is not part of some formula $\forall vQ$ in *P*.

Otherwise $(\forall vP)_\sigma$ is undefined.

We introduce now some simpler notation for substitution.

Simplified Substitutional Notation

We introduce the notation ***P*[*t*₁...*tₙ*]** to indicate a formula *P* that may contain the terms $t_1...t_n$.

We then use the simplified substitution notation $\boldsymbol{P[t'_1,...,t'_n/t_1...t_n]}$ to indicate the result of substituting in *P* for all free occurrences of $t_1...t_n$ respectively by $t'_1,...,t'_n$. Formally, $\boldsymbol{P[t'_1,...,t'_n/t_1...t_n]}$ is defined as P_σ where σ is substitution function for $\mathbf{F_{FOL}}$ and $\sigma(t'_i)=t_i$.

Sometimes rather than list all terms being substituted, it is more convenient to refer first to a sentence ***P*** using the notation $\boldsymbol{P[t_1...t_n]}$ and then later to $\boldsymbol{P[t'_1,...,t'_n/t_1...t_n]}$ by the even more abbreviated notation $\boldsymbol{P[t'_1,...,t'_n]}$. Thus, the earlier mention of $P[t_1...t_n]$ simply means that we are dealing with the formula *P* and that we are selecting for special attention the terms $t_1...t_n$, which may or may not be in *P*. In the same context if we then later refer to $P[t'_1,...,t'_n]$, we are then referring to the result of substituting $t'_1,...,t'_n$ for $t_1...t_n$ in *P*, *i.e.* to $P[t'_1,...,t'_n/t_1...t_n]$.

Examples. If a t_i does not occur in $P^n{}_mt_1...t_n$, then its replacement by t'_i makes no change in $P^n{}_mt_1...t_n$. In the extreme case in which there are no occurrences of any $t_1...t_n$ in *P*, then $P^n{}_m[t'_1,...,t'_n/t_1...t_n]=P^n{}_mt_1...t_n$, Moreover, substitution is only a partial operation on terms and formulas. If there is even one occurrence of t_i that is not free in *P* or is not free for $\sigma(t_i)$, then $P^n{}_m[t'_1,...,t'_n/t_1...t_n]$ is undefined.

$Fx[y/x]$	=	Fy	
$Fz[y/x]$	=	Fz	*x* does not occur in *Fz.*
$Fx\wedge Gy[y/x]$	=	$Fy\wedge Gy$	
$(\forall xFx)[y/x]$		undefined	The occurrence of *x* in $\forall xFx$ is not free.
$(\forall y(Fy\wedge Gx))[y/x]$		undefined	The occurrence of *x* in $\forall y(Fy\wedge Gx)$ is not free for *y*.
$(Fx\wedge(\forall y(Fy\wedge Gx))[y/x]$		undefined	One occurrence of *x* in $Fx\wedge\forall y(Fy\wedge Gx)$ is not free for *y*.

Substitution for *Some Free* Terms

An extension of a substitution function σ ***to sentences, for some free terms***, is a partial function from $\mathbf{F_{FOL}}$ to $\mathbf{F_{FOL}}$ defined inductively as follows:

Atomic Case. $(P^n t_1 ... t_n)_\sigma = P^n t'_1 ... t'_n$, such that for any $i \leq n$, either $t'_i = t_i$ or $t'_i = \sigma(t_i)$

Molecular Case. $(\sim P)_\sigma = \sim(P_\sigma)$

$(P \wedge Q)_\sigma = R \wedge S$ such that $R = P_\sigma$ or $S = Q_\sigma$

$(P \vee Q)_\sigma = R \vee S$ such that $R = P_\sigma$ or $S = Q_\sigma$

$(P \rightarrow Q)_\sigma = R \rightarrow S$ such that $R = P_\sigma$ or $S = Q_\sigma$

$(\forall vP)_\sigma = \forall v(P_\sigma)$ if for any t such that $\sigma(t)$ is defined, then

1. for any v' in t, v' is free in $(\forall vP)$, and
2. for all t in $(\forall vP)$, t is free for $\sigma(t)$ in $(\forall vP)$.

Otherwise $(\forall vP)_\sigma$ is undefined.

***P*[*t*′₁,...,*t*′ₙ//*t*₁...*t*ₙ],** read "a result of substituting in P for some free occurrences of $t_1 ... t_n$ by of respectively some of $t'_1,...,t'_n$" is defined as P_σ where σ is substitution function for $\mathbf{F_{FOL}}$ for some free terms and $\sigma(t'_i) = t_i$.

Substitution for *All Terms: Alphabetic Variance*

The extension of a substitution function σ ***to full sentences, for all terms***, is a partial function from $\mathbf{F_{FOL}}$ to $\mathbf{F_{FOL}}$ defined inductively as follows:

Atomic Case. $(P^n t_1 ... t_n)_\sigma = P^n \sigma(t_1) ... \sigma(t_n)$

Molecular Case. $(\sim P)_\sigma = \sim(P_\sigma)$

$(P \wedge Q)_\sigma = (P_\sigma . \wedge (Q_\sigma)$

$(P \vee Q)_\sigma = (P_\sigma . \vee (Q_\sigma)$

$(P \rightarrow Q)_\sigma = (P_\sigma \rightarrow (Q_\sigma)$

$(\forall vP)_\sigma = \forall \sigma(v)(P_\sigma)$ if the replacing terms are free in $\forall vP$ (define as before).

Otherwise $(\forall vP)_\sigma$ is undefined.

P is an ***alphabetic variant*** of Q, briefly $P \equiv Q$, iff there is some 1-1 substitution function σ for $\mathbf{F_{FOL}}$ for all terms such that for all constants c, $\sigma(c) = c$, and $Q = P_\sigma$.

Metatheorem 1-1. If P_σ is well defined, P_σ is an alphabetic variant of *P*, i.e. $P \equiv P_\sigma$.

Examples of Alphabetic Variants. Let σ be a full extension substitution function for all terms.

$Fx[y/x] = Fy$

$(\forall x(Fx \wedge Gx))[y/x] = \forall y(Fx \wedge Gx)[y/x] = \forall y(Fx[y/x] \wedge Gx)[y/x]) = \forall y(Fy \wedge Gy)$

$(\forall xFx \wedge Gx)[y/x] = (\forall xFx)[y/x] \wedge Fx[y/x] = \forall xFx \wedge Fy$

$(\forall xFx \wedge Gx)[x/x] = (\forall xFx)[x/x] \wedge Fx[x/x] = \forall xFx \wedge Fx$

$(\forall y(\forall xFx \wedge Gy))[z/x] = \forall y((\forall xFx \wedge Gy)[z/x]) = \forall y((\forall xFx)[z/x] \wedge Gy[z/x]) = \forall y(\forall zFz \wedge Gy)$

$(\forall y(\forall xFx \wedge Gy))[x/x] = \forall y(\forall xFx \wedge Gy[x/x]) = \forall y((\forall xFx)[x/x] \wedge Gy[x/x])) = \forall y(\forall xFx \wedge Gy)$

$(\forall y(Hy \wedge \forall x(Fx \wedge Gy)))[y/x] =$ undefined, y is not free for x in $\forall y(Hy \wedge \forall x(Fx \wedge Gy))$

$(\forall z(Hz \wedge \forall x(Fx \wedge Gy)))[y/x] = \forall z((Hz[y/x] \wedge \forall x(Fx \wedge Gy)[y/x])) = \forall z(Hz \wedge \forall x(Fx \wedge Gy)[y/x]) =$ undefined, because y is not free for x in $\forall x(Fx \wedge Gy)$

One important special variety of first-order languages are those that have the power to express what logicians call ***numerical identity***. (The term was coined by

Aristotle.) In this sense the only thing identical to an object is itself. If you count the number of things identical to *x* there is only one, *x* itself.) To do so in a first-order language we single out the first two-place predicate $P^2{}_1$, and stipulate that it be the traditional identity predicate =. (Also instead of writing =*xy* as rule R_{AF} officially requires, we usually use the more natural order typical of European languages *x*=*y*, and its negation ~*x*=*y* as *x*≠*y*.) We call this syntax ***first-order logic with identity***. Later in its semantics we stipulate that = stands for the (numerical) identity relation.

The Syntax of First-Order Logic with Identity

A syntax for ***first-order logic with identity*** is any first-order syntax in which $P^2{}_1$ in $\mathbf{Preds_{FOL}}$ is =.

Another important type of first-order language is set theory. Most of modern math and science can in fact be written in this notation. Its minimal syntax contains no constants and only three predicate: the contradiction sign, the predicate = for identity and the predicate ∈ for set membership.

The type theoretic syntax of *Principia Mathematica* is a special case. It is what is called a many-sorted first-order language with a special variety of variables assigned to each sort. In $\mathbf{L_{PM}}$ the type $^{\tau}$ is assigned a variables to form v^{τ}. Such variables are introduced into a first-order syntax by means of sortal predicates. The syntax specifies that for each type, which is understood to be a set, there is a special one-predicate, called its ***type predicate,*** that is set aside to stand for (take as its extension) that type in the standard interpretation of the syntax. Since $\mathbf{L_{PM}}$ needs no other one-place predicates, we may simply specify that there is some $P^1{}_i$ for each positive integer i, and introduce type-indexed variables as quantifications relative to this predicate. That is, $Q[v^{\tau}]$ will mean *for all v of type i,* $Q[v]$.

Example.[35] A ***(minimal) set theoretic syntax*** is any first-order logic with identity such that the set of constants is empty and there are only three predicates, as follows:

$P^0{}_1$ is ⊥

$P^2{}_1$ is =

$P^2{}_2$ is ∈

The language $\mathbf{L_{PM}}$ of ***type theory*** is a set theoretic syntax that has the further specification that there is a predicate $P^1{}_{\tau}$, for each i≥0 and that

$$Q[v^{\tau}{}_i] \qquad \text{means} \qquad \forall v(P^1{}_{\tau}v \rightarrow Q[v])$$

[35] The special set theoretic notation {x|*P*[*x*]} (read *the set of all x such that P*[*x*]) is not part of the basic (primative) vocabulary of firs-order set theory because it is introduced by definition as follows. Let us adopt the convention that by *Q*[{x|*P*[*x*]}] we mean the syntactic string of symbols *Q* that contains within it as a part the expression {*x*|*P*[*x*]}. When used in logic (*e.g.* in axiomatic set theory) this string is understood as an abbreviation for a longer formula:

$$Q[\{x|P[x]\}] \qquad \text{means } \exists y\{\forall x(x{\in}y{\leftrightarrow}P[x]) \wedge (\forall z(\forall x(x{\in}z{\leftrightarrow}P[x]){\rightarrow}z{=}y)\}$$

It follows that, being an abbreviation, *Q*[{x|*P*[*x*]}] is a well-formed formula of the syntax iff $\exists y\{\forall x(x{\in}y{\leftrightarrow}P[x]) \wedge (\forall z(\forall x(x{\in}z{\leftrightarrow}P[x]){\rightarrow}z{=}y)\}$ is.

B. Semantics

Semantics is that branch of logic in which the meaning of expressions is explained. It is important for logic because it is in semantics that good and bad arguments are distinguished. The good ones, called *valid*, transmit truth from the premises to the conclusion. To explain validity, semantics must accordingly explain truth, the more basic notion. Truth in turn is explained in terms of the correspondence of expressions with the world. A simple sentence is true if in its world its terms pick out entities that stand in the relation picked out by the predicate. The truth-value of atomic formulas is thereby assigned. The truth-values of complex formulas made up by connectives are calculated by truth-functions, one for each connective, from the truth-values of its parts. Once truth is defined, then the key logical ideas can also be defined. A sentence is a ***logical truth*** (***tautology*** in sentence logic) if it is always true, and an argument is ***valid*** if whenever its premises are true, so is its conclusion. This is the general picture that must be applied to the details of sentential and first-order syntax.

i. Sentential Semantics

Its application to sentential logic is straight forward. Atomic sentences are simply assigned a truth-value T or F indicating their descriptive success in a given world. There are as many "worlds" (or more accurately, world types) as there are different assignments of T and F to these atoms. The truth-values of molecular sentences are then calculated by truth-tables. It is customary to call an assignment of truth-values to sentences an ***interpretation*** or ***valuation****.* If the syntax contains sentences from a language that describes the things that exist in our world, then one of these interpretations will record the truth-values the sentences have in the world we actually inhabit. This assignment in a sense recreates the actual world, and for some purposes may serve as a kind of crude proxy for the "actual world" itself, much as a novel, which may be viewed as a set of sentences, represents a possible world. The formal definition of the semantics runs as follows.

The Semantics for Sentential Logic

An ***interpretation*** (or ***valuation***) for the set $\mathbf{F_{SL}}$ of ***formulas*** of an **SL language** generated by $\mathbf{AF_{SL}}$ is any assignment $\Im$ of a truth-values T or F to the formulas in $\mathbf{F_{SL}}$ that meets the following conditions:

- $\Im$ assigns to every atomic sentence in $\mathbf{AF_{SL}}$ either T or F;
- $\Im$ assigns to negations, conjunctions, disjunctions, conditionals and biconditionals the truth-value calculated by the truth-tables from the truth-values that $\Im$ assigns to its parts.

The formula P is a ***tautology*** (abbreviated $\models_{\mathbf{SL}}P$) iff for all $\Im$, $\Im$ assigns T to P.

The argument from $P_1,...P_n,...$ to Q is ***valid*** (abbreviated, $P_1,...P_n,... \models_{\mathbf{SL}}Q$) iff for any $\Im$, if $\Im$ assigns T to all of $P_1,...P_n,...$, then $\Im$ assigns T to Q*.*

ii. First-Order Semantics

First-order semantics is more complex. Though not as complicated as natural languages like English, a first-order syntax nevertheless has several different parts of speech, sentence types and varieties of atomic expression. The meanings in a world for the group of expressions as a whole is defined "recursively." First the referents in the world for the atomic expressions – constants, constants, variables, predicates, and functors – are assigned referents. Then, rules are defined for each complex expression type that fixes its referent or truth-value. There is a rule that specifies the referent of a complex term as a whole given the referent of its functor and those of its argument-terms already determined. There is a rule that specifies the truth-value of a subject-predicate sentence given the referents of its singular terms and its predicate. There are rules that calculate the truth-values of sentences made up of the sentential connectives given the truth-values of their parts. Lastly, there is a rule that determines the truth-value of a universally quantified sentence given that of its open sentence. All these specifications add up to a definition of truth for sentences.

The Domain and the Interpretations of the Atomic Descriptive Expressions. Before the referents of the descriptive terms can be fixed it must be determined what entities make up the "world". It is from within this broad set that the terms take their meaning. These are the entities that exist in that world, and it is these that the variables and quantifiers are said to "range over." Thus the first step is to fix this set, which is called the ***domain of discourse.*** In standard first-order logic this is always required to be some non-empty set D. It is therefore an assumption of the semantics of first-order logic that in every world there is at least one existing entity. There are as many possible domains, however, as there are non-empty sets, and each set may be viewed as consisting of the entities that exist in a possible world – *possibilia* they are called in mediaeval logic. Atomic descriptive expressions – constants, functors, and predicates – then are assigned referents relative to a given domain, and from these the truth-values for sentences are calculated. The descriptive expressions are those that have a fixed referent no matter where they occur. (Variables stand for entities but their referent varies relative to the position the occupy in a formula.) An assignment of meaning to these fixed expressions is called an ***interpretation relative to*** D and is represented in this book by the letter $\mathfrak{I}$. To interpret expressions, therefore, we need to specify both a domain D and the interpretation function $\mathfrak{I}$ that maps basic descriptive expressions to their referents relative to D. Using the conventions of algebra for representing an "abstract structure", we combine D and $\mathfrak{I}$ into an ordered pair <D,$\mathfrak{I}$>, which is called a ***model*** or ***structure*** and given a single letter name, for example $\mathfrak{A}$.

A model is the "mathematical" construct corresponding to our intuitive notion of a possible world. As we shall see, given the syntax and a model, we will be able to assign referents to all the referring expressions of the language, and fix a truth-value – the truth-value in that model – for every sentence.[36]

[36] More accurately, a model represent a *world type* or *set of worlds.* It is perfectly possible for there to be two worlds that are the same as far as the objects and sets named by the language go but differ from one another because there are some entities that are unnamed and that differ in the two worlds

It is traditional to name models with German Goth (*fraktur*) letters: $\mathfrak{A}$,$\mathfrak{B}$,$\mathfrak{C}$,$\mathfrak{D}$,…. With this background, we are now ready to go into a little more detail about the referents of the various atomic descriptive expressions.

Constants. Constants stand for "things" in the domain. Relative to a model their referents at every occurrence in any expression of the language is fixed. Thus relative to a domain D, we specify that a constant *c* stands for a fixed single entity $\mathfrak{I}(c)$ in D.

Predicates. Intuitively, predicates stand for sets of entities in the domain or relations that hold among these entities. Relations too, strictly speaking are sets because in set theory an *n*-place relation is a sets of *n*-tuples. Thus, if P^n is an *n*-place predicate, its interpretation $\mathfrak{I}(P^n)$ is an n-place relation on D. The set of all ordered pairs from D is conventionally named D^2, the set of all ordered triples by D^3, and the set of all ordered n-tuples by D^n. Since P^2 is a two-place predicate, $\mathfrak{I}(P^2)$ is some set of order-pairs. This fact may be written $\mathfrak{I}(P^2)\subseteq D^2$. In general $\mathfrak{I}(P^n)\subseteq D^n$.

We are almost ready to explain how the truth-values of sentences and the referents of complex singular terms are to be calculated given the domain D and the interpretation $\mathfrak{I}$ of its constants, predicates, and functors. First, however, we need some more information. We need to know for each free variable what entity in the domain it stands for.

Variables and Variable Assignments. Like constants, variables stand for "things" in the domain, but like pronouns and unlike constants their referents vary from occurrence to occurrence, from sentence to sentence. The interpretation $\mathfrak{I}$ does not fix the meaning of free or bound variables. Rather $\mathfrak{I}$ only assigns referents to those expressions that have fixed meanings in a model no matter what sentence they are in. An expression of this type, which is called a ***descriptive term*** in modern logic and ***categorematic term*** in mediaeval logic, is such that if one of its occurrences is assigned a meaning in the model, then every other occurrence, no matter what larger expression it is contained within, continues to have that meaning in that model. Relative to a given model, its occurrences all have a single meaning invariantly across all syntactic contexts. In first-order logic expressions of this type consist of just the constants, predicate letters, and functors. Unlike these terms, variables do not have a fixed meaning, but vary in their referent while the meanings of descriptive terms remain the same. While a constant will stand for the same individual, and a predicate the same set or relation, in every occurrence, a variable is said to "range over" groups of referents. In *for all x, John loves x,* for example, *John* and *loves* have fixed meaning, but the second occurrence of *x* refers back to the universal quantifier *for all x* (in linguistic terminology it functions as an "anaphora") and ranges over ranges over everything that exists. In the sentence *for some x, x loves Mary,* the second occurrence of *x* refers back to the existential quantifier *for some x* and ranges over some things in the domain. In the sentence *for all x, John loves x and for some x, x loves Mary* the second and fourth occurrences of *x* refer back to different quantifiers and range differently over the

because they exemplify properties or relations that have no predicates in the language. These worlds will then be different, but they would determine the same model.

domain. The occurrences of the variable have different meanings, *i.e.* range over referents in systematically different ways, according to their grammatical context.

Let us address specifically the semantics of free variables. Free variables function somewhat like pronouns and proper names. They stand for things. In formal logic we idealize natural language. Just as we do not allow constants to be ambiguous, once interpreted we do not permit a free variable to name more than one thing, although we do allow that once interpreted two different free variables may name the same thing. Once its meaning is fixed, it has only one referent. Viewed mathematically, a fixing of the variables' meanings is a many-one mapping, a function from the set of variables into the domain of existing entities. In logic it is called a ***variable assignment*** relative to the model. Relative to a domain D and interpretation $\mathfrak{I}$, a variable assignment is a function that pairs each variable with a unique entity in D. We shall use lower case letters ***s***, with and without prime markers and subscripts, to stand for variable assignments. Recall that for a single D and $\mathfrak{I}$, there will be as many such ***s*** as there are ways to map the set of variables into D.[37]

Consider the formula *John loves x.* The referents of *John* and *loves* are fixed relative to D by the function $\mathfrak{I}$ that specifies the meaning of all descriptive terms. But to interpret the *x* we need in addition a variable assignment ***s.*** Hence to determine the truth-value of the sentence *John loves x* three "parameters" need to be fixed, D, $\mathfrak{I}$, and ***s***. Thus, *John loves x* is true relative to <D,$\mathfrak{I}$> and ***s*** if, and only if, the individual $\mathfrak{I}$(*John*) bears the relation $\mathfrak{I}$(*loves*) to the individual ***s***(*x*).

iii. Inductive Definition of the Interpretation of Terms.

First order-logic contains not only constants, which are the logical equivalents of proper names, and free variables, the logical equivalents of pronouns, but also complex singular terms made up out of these by functors. Functors stand for functions that take arguments from the domain and map them to values also in the domain. They are regarded as descriptive terms. A functor has a fixed meaning in a model <D,$\mathfrak{I}$>, and it is fixed by the interpretation function $\mathfrak{I}$. If f^n is an *n*-place functor, then its referent $\mathfrak{I}(f^n)$ is a function (also called an operation) that maps *n* entities from D, taken in the specified order, to some entity d_{n+1} in D.

Several notational conventions have evolved to talk about this fact. First, a sequence of *n* entities from D taken in a specific order is represented as an *n*-tuple $<d_1,...,d_n>$. Thus, the fact that $\mathfrak{I}(f^n)$ pairs any $<d_1,...,d_n>$ with a unique d_{n+1} is written symbolically as $\mathfrak{I}(f^n)(d_1,...,d_n)=d_{n+1}$. Likewise, there is a convention that the set of all functions from a set B into a set A is symbolized as A^B, and that the set of

[37] The notation ***s*** for variable assignments, as well as the idea itself, is due to Tarski. To specify a variable assignment in a domain D, he used the device of an infinite sequence of entities $d_1,...,d_n$ of elements from, D and named this sequence ***s***. His convention was to "read off" the referents of variables from the sequence, which strictly speaking does not mention the variables themselves. His rule was that the first variable v_1 stands in ***s*** for the first entity in ***s***, namely for d_1, and in general the *n*-th vaiable in the syntax stands for the *n*-th entity in ***s***, namely for d_n . It is now more common to view an assignment of objects to variable as we have defined it here, *i.e.* as a mapping (function) from the set of variables into D.

functions from the set B^2 (the set of all the ordered pairs from B, which is also called B×B) into A is $A^{(B^2)}$. Thus the fact that $\Im(f^1)$ is a function from D into D may be expressed $\Im(f^1)\in D^D$, as that $\Im(f^2)$ is a function from D^2 into D as $\Im(f^1)\in D^{(D2)}$ In general it is required that $\Im(f^n)\in D^{(D^n)}$.[38]

In set theory, the family (set) of all subsets of a set A is called ***the power set*** of A and is symbolized P(A). Now, since a relation is some set of *n*-tuples, it is a subset of the set of all n-tuples. Hence instead of writing $\Im(P^2)\subseteq D^2$, we could state the same fact by saying it is a subset of the set of all ordered pairs made up of elements from D, *i.e.* as $\Im(P^2)\in$ P(D^2). In general, for any *n*-place predicate P^n, $\Im(P^n)\in$ P(D^n).

This wealth of notation allows us to talk about the "categories" that correspond to "parts of speech". We are treating parts of speech here as sets of descriptive expressions – terms, predicates, functors, and formulas. Since Aristotle, the set of entities corresponding to a part of speech has been called a ***category***. According to standard first-order semantics, then, the category of terms is D, that of *n*-place predicates is P(D^n), that of *n*-place functors is $D^{(Dn)}$, and that of formulas is {F,T} (this set is also known as 2={0,1}). These are the modern descendants of Aristotle's famous list of categories: substance, quality, quantity, time, place, etc. The descriptive terms of modern logic are, thus, what were called ***categorematic*** terms in mediaeval logic, *i.e.* the terms that stand for entities in the world, the terms that correspond to a "category" of real existents. The so-called "logical expressions", *e.g.* ~,∧,∨,→, ↔, ∀, (, and), on the other hand, do not stand for entities, they are the ***syncategorematic*** expressions of modern logic, the ones that "go along with" (*syn*=*with* in Greek) the genuine categorematic ones.

What do syncategoremata do? We can say that they provide a syntactic mark indicating that a grammatical rule has been applied because though they do not themselves have referents or truth-values, they are added to shorter referring expressions in the syntactic construction of a longer referring expression. Thus the formation rule in syntax for negation is marked by the negation sign: ~ is added to *P* to make up ~*P.* The rule for making up a conjunction $P\wedge Q$ from *P* and *Q* is marked by the introduction of the logical sign ∧.

If this convention were to be followed systematically, there should also be a two additional logical signs, one for functor combinations and one for atomic sentences, *i.e.* one to signal that the terms $t_1...t_n$ and the functor f^n are put together in the formation rule for the complex term $f^n(t_1...t_n)$, and another to signal that $t_1...t_n$ and P^n are put together in the atomic formula formation rule to construct the formula $P^nt_1...t_n$. In first order syntax, however, these rules are marked not by a special logical sign but simply by the word order (concatenation) of the component expressions. A functor combination is indicated by the concatenation in order of the

[38] The origins of this notation and that for relations is a bit baroque. For example, the notation $\Im(P^2)\subseteq D^2$ derives from the following customs. The set of all pairs of elements from D, traditionally called DxD, which is called the Cartesian product of D with D (because it is a generalization of Descartes' original investigations of the real number plane called $\mathbb{R}$x$\mathbb{R}$, or $\mathbb{R}^{2.}$ The exponential notation is used because if there are *n* objects in $\mathbb{R}$, then there are n^2 objects in $\mathbb{R}$x$\mathbb{R}$.

functor, an open parenthesis, the terms, and a closed parenthesis, and the formation of an atomic formula by the concatenation of the predicate with its terms. (In mediaeval logic, Latin was the language of logic, and it does contain what was recognized as a syncategorematic term marking the joining of a subject and a predicate to form a simple proposition, namely the verb *to be,* which was called the ***copula***.) Though logical signs lack referents, there is still a sense in which they have "meaning", because corresponding to each formation rule, marked by its characteristic logical sign, there is a "semantic rule". This is a rule in the semantic theory that states how the referents of the immediate parts of an expression determine that of the whole. Each formation rule, moreover has its own such semantic rule. Corresponding to the negation formation rule, and its marker ~, it the negation truth-function, and corresponding to the rule for conjunction and the marker $\wedge$ is the conjunction truth-function. Likewise each of the other connectives has its truth-function. As we shall see shortly, there are also distinctive rules for the rule constructing complex terms, for the rule constructing atomic formulas, and for the rule constructing universally quantified formulas and its marker $\forall$. The meaning of logical signs, in other words, may be said to be "explained" by their corresponding semantic rules.

Let us turn then to the task of defining, in general terms relative to a model <D,$\Im$> and variable assignment ***s,*** the referent of a complex singular term $f^n(t_1...t_n)$ made of an *n*-place functor f^n and the singular terms $t_1...t_n$. Since in the syntax the set of terms (simple and complex) is defined by induction, so is the notion of the referent of a term. The inductive definition has a basis clause and an inductive clause. In the basis clause the referent of the basic terms (constants and free variables) is defined. Note that these have already be fixed by $\Im$ for constants and by ***s*** for variables. In the inductive step the referent of complex term $f^n(t_1...t_n)$ relative to <D,$\Im$> and ***s*** is defined on the assumption that the referents of each of its immediate parts $t_1...t_n$ have been defined relative to any model and variable assignment for that model.

The only domains, interpretations and variable assignments relevant to determining the referent of $f^n(t_1...t_n)$ in <D,$\Im$> and ***s*** are <D,$\Im$> and ***s*** themselves. The specific condition is simply that the object paired with $f^n(t_1...t_n)$ is that determined by applying the function $\Im(f^n)$ to the *n*-tuple of objects in the domain named by its immediate parts taken in order, *i.e.* by applying the function $\Im(f^n)$ to < $\Im^{\mathfrak{A}}_{s}$ (t_1),..., $\Im^{\mathfrak{A}}_{s}$ (t_n))>. The entire definition may be written very simply:

Basis Clause. $\Im^{\mathfrak{A}}_{s}(t)=\Im(t)$ if *t* is a constant, and $\Im^{\mathfrak{A}}_{s}(t)=$***s*** (t) if *t* is a variable
Inductive Clause. $\Im^{\mathfrak{A}}_{s}(f^n(t_1...t_n))= \Im(f^n)(\Im^{\mathfrak{A}}_{s}(t_1),..., \Im^{\mathfrak{A}}_{s}(t_n))$.

iv. Inductive Definition of the Interpretation of Formulas.

Definition of Truth or Satisfaction. The central part of the semantic theory is the actual definition of *truth* in a model. More precisely, truth-in-a-model must be relativized to yet a further parameter. It only makes sense if a variable assignment is also fixed. Thus, what is defined is "truth-in-a-model relative to a variable assignment". For this relativized notion of truth Tarski coined the technical name *satisfaction.* Thus the first step is to define for an arbitrary formula *P* what it is for *P* to be satisfied relative to a model <D,$\Im$> and variable assignment ***s.*** (We shall find later that for some formulas, namely sentences, the extra parameter carries no special information because a sentences does not differ in truth-value across variable assignments. At that point, we will have motivation to define a non-relativized notion of truth as "satisfaction over all variable assignments." It is important to see, however, that this later non-relativized notion of truth depends on the prior definition of satisfaction because satisfaction is used to define non-relativized truth.)

Since the set of formulas is defined by induction, so is the notion of satisfaction. Since the basic elements in the construction of formulas are atomic formulas, the basis step in the definition of satisfaction consists in stating "satisfaction conditions" for atomic formulas. In the inductive step of the definition, however, there are multiple clauses, one for each grammar rule used in the inductive definition of formula. There is one for each connective and one for the universal quantifier.

Basis Clause: Subject-Predicate Sentences. It is at this point that we define for an atomic formula $P^nt_1...t_n$ the statement:

$P^nt_1...t_n$ is satisfied relative to model <D,$\Im$> and variable assignment ***s***.

This statement is customarily abbreviated by two equivalent notations

$\mathfrak{A}_{\boldsymbol{s}}\models P^nt_1...t_n$, or

$\Im^{\mathfrak{A}}_{s}(P^nt_1...t_n)$=T

where $\mathfrak{A}$=<D,$\Im$>.

To review, let us list what determines the truth-value of $P^nt_1...t_n$ in <D,$\Im$> and ***s***. There are three things:

(1) the entities $\Im^{\mathfrak{A}}_{s}(t_1)$,..., $\Im^{\mathfrak{A}}_{s}(t_n)$ named by the singular terms $t_1...t_n$,

(2) the set or relation picked out by the predicate $\Im^{\mathfrak{A}}_{s}(P^n)$, and

(3) whether the entities $\Im^{\mathfrak{A}}_{s}(t_1)$,..., $\Im^{\mathfrak{A}}_{s}(t_n)$ stand in the relation (or are in the set) $\Im(P^n)$. If they do, the sentence is satisfied; otherwise it is not.

In brief, the atomic formula is true relative to these parameters if the terms name entities that in order fall in the relation named by the predicate. The rule below states this idea precisely. It does so in two equivalent forms. The second uses set theoretical notation and makes use of the fact that $\Im(P^n)$ is understood as a set of *n*-tuples.

$\mathfrak{A}_{\boldsymbol{s}}\models P^nt_1...t_n$ (equivalently $\Im^{\mathfrak{A}}_{s}(P^nt_1...t_n)$=T) iff $\Im^{\mathfrak{A}}_{s}(t_1)$,..., $\Im^{\mathfrak{A}}_{s}(t_n)$ stand in relation $\Im^{\mathfrak{A}}_{s}(P^n)$.

$\mathfrak{A}_{\boldsymbol{s}}\models P^nt_1...t_n$ (equivalently $\Im^{\mathfrak{A}}_{s}(P^nt_1...t_n)$=T) iff $<\Im^{\mathfrak{A}}_{s}(t_1),...,\Im^{\mathfrak{A}}_{s}(t_n)>\in\Im(P^n)$.

Inductive Clauses: Formulas Made Up from ~, ∧, ∨, →, and ↔. The next task is to define satisfaction for of a complex formula made by sentential connective relative to <D,$\Im$> and ***s*** in terms of the satisfaction of its immediate parts. For the connectives the only domains, interpretations and variable assignments relevant to determining the satisfaction of a term in <D,$\Im$> and ***s*** are <D,$\Im$> and ***s*** themselves. Each connective has its own inductive clause which is essentially an appeal to its traditional truth-table:

$\mathfrak{A}_s \models \sim P$ iff not $\mathfrak{A}_s \models P$; or $\mathfrak{I}^{\mathfrak{A}}_s(P)$=T iff $\mathfrak{I}^{\mathfrak{A}}_s(P)\neq$T
$\mathfrak{A}_s \models P\wedge Q$ iff ($\mathfrak{A}_s \models P$ and $\mathfrak{A}_s \models Q$); or $\mathfrak{I}^{\mathfrak{A}}_s(P\wedge Q)$=T iff, $\mathfrak{I}^{\mathfrak{A}}_s(P)$=T and $\mathfrak{I}^{\mathfrak{A}}_s(Q)$=T
$\mathfrak{A}_s \models P\vee Q$ iff ($\mathfrak{A}_s \models P$ or $\mathfrak{A}_s \models Q$); or $\mathfrak{I}^{\mathfrak{A}}_s(P\vee Q)$=T iff, $\mathfrak{I}^{\mathfrak{A}}_s(P)$=T iff $\mathfrak{I}^{\mathfrak{A}}_s(Q)$=T
$\mathfrak{A}_s \models P\rightarrow Q$ iff (not $\mathfrak{A}_s \models P$ or $\mathfrak{A}_s \models Q$); or $\mathfrak{I}^{\mathfrak{A}}_s(P\rightarrow Q)$=T iff, $\mathfrak{I}^{\mathfrak{A}}_s(P)\neq$T or $\mathfrak{I}^{\mathfrak{A}}_s(Q)$=T
$\mathfrak{A}_s \models P\leftrightarrow Q$ iff ($\mathfrak{A}_s \models P$ iff $\mathfrak{A}_s \models Q$); or $\mathfrak{I}^{\mathfrak{A}}_s(P\leftrightarrow Q)$=T iff, $\mathfrak{I}^{\mathfrak{A}}_s(P)$=T iff $\mathfrak{I}^{\mathfrak{A}}_s(Q)$=T

Inductive Clause: Formulas Made up from the Universal Quantifier. In the syntax of first-order logic, the formula $\forall xPx$ has as its immediate parts P and $x.$ Intuitively, $\forall xP$ is satisfied if Px is satisfied when its free variable x ranges over the entire domain. But what does it mean for a variable x to "range over" entities in the domain? The idea is explained by means of variable assignments. We may use they to that the universal formula $\forall xPx$ is satisfied if Px would be satisfied no matter what x stands for. In other words, $\forall xPx$ is satisfied exactly when Px is satisfied under every possible assignment of a referent to $x.$

Let us be more precise. The general framework dictates that we are trying to come up with a inductive definition of truth or, as it is called in this context, satisfaction. [39] The definition begins with the basis step (which we have already stated) that determines for each atomic formula, each model, and each variable assignment for the model, whether or not the formula is satisfied. Next comes the inductive step. For each grammar rule, it specifies the conditions under which the formula produced by the rule is satisfied relative to a model and variable assignment. Here it is assumed, and it is indeed necessary for the definition to be well-defined, that the conditions have be previously defined that specify when the formula's immediate parts are satisfied relative to a model and variable assignment. The inductive clauses for the sentential connectives, which were just stated above, fit this form. In the case of the universal quantifier the whole formula is $\forall xPx$ and its immediate part is $Px.$ The task then is to explain when $\forall xPx$ is satisfied relative to a model <D,$\mathfrak{I}$> and variable assignment $\boldsymbol{s}$. It is assumed that the conditions are already defined that specify when Px is satisfied relative to a model and variable assignment. It is important to note that this prior specification extends beyond the single model <D,$\mathfrak{I}$> and assignment $\boldsymbol{s}$. We assume we know what it is for Px to be satisfied relative to any model and variable assignment. It would not be sufficient to restrict attention to just D, $\mathfrak{I}$, and $\boldsymbol{s}$ because knowing whether Px is satisfied relative to them alone would tell us only whether the formula Px is satisfied by a single entity in the domain, namely $\boldsymbol{s}(x)$. But what we need to know is whether everything in D satisfies Px. Let us then draw on the fact that we have previously defined not just

[39] It is also called a recursive definition, but the terminology needs some explanation. Any definition that defines a relation or function by first definiing it for some basic elements, and then defining its application to an output on the assumption that it is defined for its input is called a ***recursive definition*** on analogy with the process of definition of functions "by recursion" in Gödel's theory of primative recursive function where functions are defined this way by the technique called "recursion". If a relation is understood as a set of *n*-tuples, a recursive definition in this broader sense is simply another name for an inductive definition of a set of *n*-tuples. This use of "recursive" is however rather broader than Gödel's because it does not make the requirement imposed in Gödel's special theory that the inputs or the outputs of the function being defined "by recursion" are "calculable functions".

what it is for Px to be satisfied relative to a <D,$\mathfrak{I}$> and $\boldsymbol{s}$, but relative to <D,$\mathfrak{I}$> and any variable assignment whatever for <D,$\mathfrak{I}$>.

To see how it is legitimate to appeal to this broad set of variable assignments, let us look a little deeper into the process of defining satisfaction by induction. Let us refer to satisfaction by the symbol $\models$.[40] Strictly speaking, $\models$ is a 5-place relation that holds relates a formula P and the three "parameters" necessary for determining satisfaction to a truth-value V. The three parameters that need to be specified in addition to P itself are a domain D, an interpretation $\mathfrak{I}$ over D, and a variable assignment $\boldsymbol{s}$ relative to D and $\mathfrak{I}$ Accordingly, we speak of P as *true* or *satisfied* relative to a model <D,$\mathfrak{I}$> and variable assignment $\boldsymbol{s}$ for <D,$\mathfrak{I}$>. Since any 5-place relation in set theory is really a set of 5-tuples, $\models$ is really a set of 5-tuples <P,D,$\mathfrak{I}$,$\boldsymbol{s}$,V>. Thus, strictly speaking, the set theoretic notation for the fact that P is satisfied in <D,$\mathfrak{I}$> relative to the $\boldsymbol{s}$ is <P,D,$\mathfrak{I}$,$\boldsymbol{s}$,V> is a member of a relation (in this case a set of 5-tuples) incated by the symbol $\models$. Because the symbol $\models$ has other uses – for example it is used to say the argment from P to Q is valid in the notation $p\models Q$, or to say that P is a logical truth in the notation $\models P$ – let us use another symbol for the 5-place relation. Let us call it **Sat** and call it the *satisfaction* relation. Thus, we will write

$<P,D,\mathfrak{I},\boldsymbol{s},V>\in\mathbf{Sat}$

which means

> In the structure <D,$\mathfrak{I}$> relative to variable assignment $\boldsymbol{s}$ the formula P has the truth-value V.

As we shall see the later, there is a briefer and more common notation for this fact that is easier to read::

$\mathfrak{A}_{\boldsymbol{s}}\models P$

or

$\mathfrak{I}^{\mathfrak{A}}_{s}(P)$=T.

Our goal at this point is to explain how the 5-place relation **Ref** is defined using an inductive definition. For that purpose let us use **Ref** as the name for that set of 5-

[40] The notation needs some explanation. Historially the single turn-style $\vdash$ was introduced by Frege in his *Begriffscrift*. The horizonal part – is translated "it is true that", and ther vertical part as "it is a theorem that". Hence $\vdash$ meant "it is a theorem that it is true that…" This composite sense was simplified in later logic so that $\vdash$ meant just "it is a theorem that …" $\vdash P$ has aways retained its syntactic sense to mean relative to some axiom system "P is provable by a formal deduction from the axioms by the rules of proof ". It is now also extended to arguments but still in a syntaxtic sense. Thus $X\vdash P$ means relative to an axiom system "P is provable from X and the axioms using the rules of proof." The double turnsyle $\models$ (some books use $\Vdash$) was introduced to indicate a semantic relation. It may be limited to a single formula as in $\models P$ in which case it means "P is a logical truth, *i.e.* true in all interpretations" and it is also extended to arguments to indicate a valid argument, also a semantic notion. Thus $X\models P$ means "the argument from X to P is valid". Now, given this history, the notation $\mathfrak{A}_{\boldsymbol{s}}\models P$ seems anomalous because $\mathfrak{A}_{\boldsymbol{s}}$ is a "world", not a set of formulas. However, we might think of a "world" as a story, as for example the set of sentences contained between the covers of a novel. This is exactly the way a "model" was conceived in the 1940's before Tarski invented the current notion. Such a "world-as-set-of-formulas" was called a *state description* (by Carnap, *e.g* in *Meaning and Necessity*) and a *model set* (by Hinttikka, *e.g.* in *Knowledge and Belief*). The notation dates from this period. Thus $\mathfrak{A}_{\boldsymbol{s}}\models P$ may be read as "from the true formulas of the world $\mathfrak{A}_{\boldsymbol{s}}$ the formula P follows in a valid argument". This is a long-winded substitute for saying "P is true in the world $\mathfrak{A}_{\boldsymbol{s}}$."

tuples and set out how it constructed in the inductive manner. Like every inductive definition the definition of **Ref** will has basis and inductive steps.

The basis step defines the set of basic elements of **Ref** from which all others are constructed. This "starter set" is the set of all atomic formulas satisfied in some model under some variable assignment. We have earlier explained how this set is defined in terms of $<D,\Im>$ and $\boldsymbol{s}$. The basic set is:

$\{<P^n t_1,...,t_n,D,\Im,\boldsymbol{s},V>|$ $<D,\Im>$ is a model, $\boldsymbol{s}$ is a variable assignment for $<D,\Im>$ & either $<\Im^{\mathfrak{A}}_{s}(t_1),...,\Im^{\mathfrak{A}}_{s}(t_n)>\in\Im(P^n)$ and V= T, or $<\Im^{\mathfrak{A}}_{s}(t_1),...,\Im^{\mathfrak{A}}_{s}(t_n)>\notin\Im(P^n)$ and V=F$\}$

Thus for any atomic formula $P^n t_1,...,t_n$, any D, any $\Im$, and any $\boldsymbol{s}$, either $<P^n t_1,...,t_n,D,\Im,\boldsymbol{s},T>$ or $<P^n t_1,...,t_n,D,\Im,\boldsymbol{s},F>$ will be in **Ref** but not both. The clause above makes the determination (defines) in a single stroke for *all* atomic formulas, *all* models and *all* variable assignments.

Next comes the definition's inductive step. For each grammar rule R_i of the syntax a method is described for constructing new 5-tuples in **Ref** from old. It is assumed that it has already been determined for each of a formula's immediate parts, each model and every variable assignment for that model whether the part is satisfied or not. The rule takes this information and formulates a satisfaction condition for the whole appropriate to its grammar and meaning.

Let us be a bit more precise. Let the whole formula be P constructed from the immediate parts $Q_1,\ldots,Q_n$ by grammar rule R_i. Let V rage over truth-values (these will usually be T and F). Then, in general, there is some condition C_i such that:

$<P,D,\Im,\boldsymbol{s},V>\in\models$ iff condition C_i is met by some elements of the (previously defined) set $\{<Q_i,D',\Im',\boldsymbol{s}',V'>|$ Q_i is an immediate part of P, $<D',\Im'>$ is a model, $\boldsymbol{s}'$is a variable assignment for $<D',\Im'>$, and $Q_i,D',\Im',\boldsymbol{s}',V'>\in$ **Ref** $\}$

Note that the set $\{<Q_i,D',\Im',\boldsymbol{s}',V'>|$ Q_i is an immediate part of P, $<D',\Im'>$ is a model, and $\boldsymbol{s}'$is a variable assignment for it$\}$ is a large. In particular, in the definition above in the various $<Q_i,D',\Im',\boldsymbol{s}',V'>$ in the *definiens* (right side) , D′ may be different form the D in the *definiendum* (left side), $\Im'$ may differ from the $\Im$ in the *definiendum*, and $\boldsymbol{s}'$ from the $\boldsymbol{s}$ in the *definiendum*. Thus, the truth of P relative to one world and interpretation (namely $<D,\Im>$ and $\boldsymbol{s}$) is in general explained in terms of the truth of immediate part Q_i relative to other worlds $<D',\Im'>$ and variable assignments $\boldsymbol{s}'$.

Here are some examples. Sentence logic is simplest. The definition of satisfaction of $\sim P$ in a world and variable interpretation stated earlier may now be reformulated to fit this form:

$<\sim P\wedge Q,D,\Im,\boldsymbol{s},T>\in$**Ref** iff, $\{<P,D,\Im,\boldsymbol{s},F>\}\subseteq$ **Ref**

$<\sim P\wedge Q,D,\Im,\boldsymbol{s},F>\in$**Ref** iff, not $\{<P,D,\Im,\boldsymbol{s},F>\}\subseteq$ **Ref**

The definition of satisfaction of $P\wedge Q$ in a world and variable interpretation stated earlier may now be reformulated to fit this form:

$<P\wedge Q,D,\Im,\boldsymbol{s},T>\in$ **Ref** iff, $\{<P,D,\Im,\boldsymbol{s},T>, <Q,D,\Im,\boldsymbol{s},T>\}\subseteq$ **Ref**

<*P*∧*Q*,D,ℑ,***s***,F>∈ **Ref** iff, not {<*P*,D,ℑ,***s***,T>, <*Q*,D,ℑ,***s***,T>} ⊆ **Ref**

First-order logic contains quantified formulas that have truth-conditions that depend on (are defined in terms of) multiple variable interpretations but relative to the same model:

<∀*xP*,D,ℑ,***s***,T>∈ **Ref** iff {<*P*,D,ℑ,***s′***,T>| for some ***s′***, ***s′*** is an *x*-variant of ***s***} ⊆ **Ref**

<∀*xP*,D,ℑ,***s***,F>∈ **Ref** iff not {<*P*,D,ℑ,***s′***,T>| for all ***s′***, ***s′*** is an *x*-variant of ***s***} ⊆ **Ref**

Modal logic contains formulas in a model that depend on the truth of its parts in wide ranges of other models. For example, □*P* (read "Necessarily *P*") is true in one model <D,ℑ> and variable assignment ***s*** iff its part *P* is true in every model <D′,ℑ′> and every variable assignment ***s′*** for <D′,ℑ′> every world and assignment whatever (in the so-called S5 interpretation):

<□*P*,D,ℑ,***s***,T>∈ **Ref** iff, {<*P*,D′,ℑ′,***s′***,T>| for some D′, some ℑ′, some ***s′***, <D′,ℑ′>is a model and ***s′***is a variable assignment for <D′,ℑ′>} ⊆ **Ref**

<□*P*,D,ℑ,***s***,F>∈ **Ref** iff, not {<*P*,D′,ℑ′,***s′***,T>| for some D′, some ℑ′, some ***s′***, <D′,ℑ′>is a model and ***s′***is a variable assignment for <D′,ℑ′>} ⊆ **Ref**

The important point is that, in the inductive step, when we state conditions defining whether <*P*,D,ℑ,***s***,V> is in **Ref**, we may draw on the fact that for each immediate part Q_i of *P*, we have already defined when <Q_i,D′,ℑ′,***s′***,V′> is in **Ref**, for every non-empty set D′, for every interpretation ℑ′ for D′, for every assignment ***s′*** for any <D′,ℑ′>, and for every truth-value V′.[41]

[41] **Note on the Inductive Definition of Terms.** Though the extra detail was not necessary earlier in stating the inductive definition of the interpretation of terms, strictly speaking that definition is likewise a induction over a four-place relation. Let us revisit that definition briefly using the vocabulary just introduced. Strictly speaking what is being defined by induction for terms is a set, which we may call for the moment **Ref**, that is a set of 5-tuples <*t*,D,ℑ,***s***,*d*> that joins a term *t,* domain D, interpretation ℑ, and variable assigment ***s***, to a referent *d* of *t* in D. The basis step defines the set of of 5-tuples in which *t* is a constant or variable. It is the set in which the term is mapped onto what it is assigned by ℑ if it is a constant or by ***s*** if it is a variable. The "starter set" for term reference then is:

{<*t*,D,ℑ,***s***,*d*>| <D,ℑ> is a model, ***s*** is a variable assignment for <D,ℑ> and *d*=ℑ(*t*) if *t* is a constant and *d*=***s***(*t*) if *t* is a variable}.

Or more briefly, for any constant or variable *t,*

<*t*,D,ℑ,***s***,*d*>∈**Ref** iff, *d*=ℑ(*t*) if *t* is a constant, and *d*=***s*** (*t*) if *t* is a variable

The inductive step states membership conditions for <$f^n(t_1...t_n)$,D,ℑ,***s***,*d*> in **Ref** in terms some condition *C* that must hold on the elements of **Ref** for the immediate parts $t_1...t_n$ of $f^n(t_1...t_n)$:

<$f^n(t_1...t_n)$,D,ℑ,***s***,*d*>∈**Ref** iff the condition *C* holds for specified elements of {<t_1,D′,ℑ′,***s′***,*d′*>|t_i is an immediate part of $f^n(t_1...t_n)$, <D′,ℑ′> is a model, ***s′***is a variable assignment for <D′,ℑ′> and, for some $d_1...d_n$ in D′, <t_i,D′,ℑ′,***s′***,d_1>∈**Ref**,…,<t_n,D′,ℑ′,***s′***,d_n>∈**Ref**, and *d′*=ℑ′(f^n)($d_1...d_n$)}

The only domains, interpretations and variable assignments relevant to determing the referent of a term in <D,ℑ> and ***s*** are <D,ℑ> and ***s*** themselves. The specific condition *C* is simply that the object paired with the complex term be that determined by applying the function named by the functor to the *n*-tuple of objects in the domain named by its immediate parts taken in order:

<$f^n(t_1...t_n)$,D,ℑ,***s***,*d*>∈**Ref** iff <t_1,D,ℑ,***s***,d_1>∈**Ref**,…,<t_n,D,ℑ,***s***,d_n>∈**Ref** and *d*=ℑ(f^n)($d_1...d_n$)

Variants of a Variable Assignment. Intuitively, to know whether $\forall xPx$ is satisfied in a model <D,$\Im$> relative to ***s***, we need to know that everything in D satisfies its immediate part *Px*. How can we formulate that condition? As is now clear, we can assume that we know what it means for *Px* to be satisfied not only in <D,$\Im$> and ***s***, but in any model and variable assignment for that model. In particular we can assume that we know what it means for *Px* to be satisfied not just relative to ***s*** but also in any variable assignment for the model <D,$\Im$>. That is, we have already defined what it means to say *Px* is satisfied in <D,$\Im$> for any interpretation of *x.* We can use that fact to formulate what it would be for *Px* to be true for every entity in D. Let us consider some examples of elements of the domain. Let us assume that d_1, d_2, d_3 are in D. Now let us vary the assignment that ***s*** makes to *x* so that *x* changes its referent to stand for these three things, but at the same time we want to require that ***s*** not change any of the assignments it makes to the other free variables in the in syntax. We define three new assignments, one for each d_i. Each is called an ***x*-variant** of ***s***. Let us call these new assignments $\boldsymbol{s}_1$, $\boldsymbol{s}_2$ and $\boldsymbol{s}_3$. They are to be like ***s*** except that $\boldsymbol{s}_1(x)=d_1$, $\boldsymbol{s}_2(x)=d_2$, and $\boldsymbol{s}_3(x)=d_3$. Since in set theory each variable assignment is really a set of pairs <*v,d*> such that *v* is a variable and *d* is the object assigned to *v*, to make the new assignments, we simply take the pair <*x*,***s***(*x*)> out of ***s*** and replace it with a new pair giving *x* its new assignment. We take a subset out with the complementation operator − and put one in with the union operator $\cup$:

$\boldsymbol{s}_1= \boldsymbol{s} - \{<x,\boldsymbol{s}(x)>\}\cup\{<x,d_1>\}$
$\boldsymbol{s}_2= \boldsymbol{s} - \{<x,\boldsymbol{s}(x)>\}\cup\{<x,d_2>\}$
$\boldsymbol{s}_3= \boldsymbol{s} - \{<x,\boldsymbol{s}(x)>\}\cup\{<x,d_3>\}$

It is by varying the referent of *x* in ***s*** through all its *x*-variants that *x* is made to ***range over*** the entire set D. (Note that below to insure that there is an *x*-variant for every object in the domain, we allow ***s*** to be an *x*-variant of itself.)

Sentences Made Up by the Universal Quantifier $\forall$. In the context of the inductive clause, we know by hypothesis whether *Px* is satisfied in $\boldsymbol{s}_1$, $\boldsymbol{s}_2$ and $\boldsymbol{s}_3$. In fact, by hypothesis it has been previously defined for any *x*-variant ***s***′ of ***s*** whether *Px* is satisfied relative to ***s′*** in <D,$\Im$>. We therefore have a way to say "*Px* is satisfied by every element *d* of the model <D,$\Im$>" and can do so in vocabulary that by hypothesis is already well defined:

for any *x*-variant ***s′*** of ***s***, *Px* is satisfied in ***s′***.

It is this clause, which is formulated in terms of variable assignments, that we use to state the conditions under which the universally quantified expression $\forall xPx$ is true in <D,$\Im$> relative to ***s***:

If we adopt the convention of assuming that the domain D of $\Im$ is clear from the context, we can adopt the simplifying notation:

$\Im_{\mathbf{s}}(t)=d$ means $<t,\mathrm{D},\Im,\boldsymbol{s},d>\in$**Ref**

and the entire definition may be written very simply:

Basis Clause. $\Im_{\mathbf{s}}(t)=\Im(t)$ if *t* is a constant and $\Im_{\mathbf{s}}(t)=\boldsymbol{s}\ (t)$ if *t* is a variable
Inductive Clause. $\Im_{\mathbf{s}}(f^n(t_1...t_n))= \Im(f^n)(\Im_{\mathbf{s}}(t_1),...,\Im_{\mathbf{s}}(t_n))$.

$\mathfrak{A}_{\boldsymbol{s}}\models\forall xPx$ (equivalently, $\mathfrak{I}^{\mathfrak{A}}_{s}(\forall xPx)$=T) iff for any *x*-variant ***s′*** of ***s***, $\mathfrak{A}_{\boldsymbol{s}'}\models Px$ (equivalently, $\mathfrak{I}^{\mathfrak{A}}_{s'}(Px)$=T)

Intuitively, if in every *x*-variant of ***s*** ′ of ***s***, ***s′***(*x*) is something to which ℑ(*John*) bears the relation ℑ(*loves*), then ℑ(*John*) bears the ℑ(*loves*) relation to everything. In other words, in that case the sentence *for all x, John loves x* is true in D relative to ℑ and ***s***. Likewise, *for some x, John loves x* is true in D relative to ℑ and ***s*** some *x*-variant of ***s*** ′of ***s***, ℑ(*John*) bears the relation ℑ(*loves*) to the individual ***s′***(*x*). Accordingly, though the existential quantifier is not included in the primitive syntax, it is introduced by definition:

$\exists xP$ is an abbreviation of $\sim\forall x\sim P$

This definition insures:

$\mathfrak{A}_{\boldsymbol{s}}\models\exists xPx$ (equivalently, $\mathfrak{I}^{\mathfrak{A}}_{s}(\exists xPx)$=T) iff for some *x*-variant ***s′*** of ***s***, $\mathfrak{A}_{\boldsymbol{s}'}\models Px$ (equivalently, $\mathfrak{I}^{\mathfrak{A}}_{s'}(Px)$=T)

When a sentence that contain both free and bound variables, like *for all x, y loves x,* is evaluated relative to D, an interpretation ℑ, and a variable assignment ***s***, it is ℑ that fixes the referent of the predicate *loves* because the role of ℑ is to interpret the descriptive terms. But it is ***s*** that fixes the referent of the free variables, in this case *y*. It is also ***s*** that determines the relevant variants for evaluating quantifiers. In this case there is a universal quantifier *for all x* and hence the relevant variants will be *x*-variant, and they will be variants of *x* relative to ***s***. Note that if ***s′*** is a *x*-variant of ***s***, then ***s′*** retains the same values as ***s*** for variables other than *x,* and hence ***s′***(*y*)=***s***(*y*). Thus, *for all x, y loves x* is true in D relative to ℑ and ***s***, iff, in every *x*-variant of ***s′*** of ***s***, ***s′***(*y*) (which is the same as ***s***(*y*)) bears the relation ℑ(*loves*) to the individual ***s′***(*x*). In this case, ***s′***(*y*) (which is the same as ***s***(*y*)) is an individual, ℑ(*loves*) is a relation in D, and there are as many different *x*-variants ***s′*** of ***s*** and as many individuals ***s′***(*x*) as there are entities in the domain.

Truth *Simpliciter*. The function $\mathfrak{I}^{\mathfrak{A}}_{s}$ assigns a referent to an expression relative to three things: (1) the domain D of 𝔄, (2) the assignment ℑ of referents to atomic expressions in D, and (3) the variable assignment ***s*** that determines the referent of each variable in D. All three "parameters" are needed to interpret formulas.

There are some formulas however for which specifying a variable assignment is irrelevant. We have already met the case of atomic formulas in which there are no variables. All we need to know for the truth-value of $P^nc_1...c_n$ when $c_1...c_n$ are constants is $\mathfrak{I}(c_1),...,\mathfrak{I}(c_n)$, and $\mathfrak{I}(P^n)$. Another case consists of sentences without free-variables. A closed formula, which called a ***sentence*** in first-order terminology, is true either in all variable assignments or in none. That is, it follows from the definition of $\mathfrak{I}^{\mathfrak{A}}_{s}$ that: for any closed formula *P,*

for some ***s***, $\mathfrak{A}_{\boldsymbol{s}}\models P$ is equivalent to for all ***s***, $\mathfrak{A}_{\boldsymbol{s}}\models P$.
for some ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(\forall xPx)$=T is equivalent to for all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(\forall xPx)$=T

While we do need the notion of variable assignment to figure out whether a universally quantified formula is true, once we know it is true, relativizing truth to a particular variable assignment is irrelevant because it is true in all of them. Accordingly, a closed formula is said to be ***true* simpliciter** (*i.e.* true regardless of variable assignment) iff it is true in any or all variable assignments.

Normally the case in which a variable assignment is absolutely relevant the truth is one in which the formula contains a free-variable. The free-variable functions much like a pronoun, and a variable assignment ***s*** functions as its "context of use" that fixes its referent within the wider world, which is represented in the formalism by the parameters D and $\mathfrak{I}$. There is, however, a second reading of free-variables that is found among mathematicians and logicians. It is not a usage common among non-specialists because ordinary people do not write formulas with free-variables. But it is common to see in mathematics books or on the blackboard in a math class a formula with free-variables like:

The formula for the line *l* is $y = ax + b$

Here the intention is that the variable really is not free. There are really universal quantifiers "understood" for be binding the variables. That is, what is intended may be expressed more fully by:

For the line *l* , for any *x* and for any *y,* $<x,y>\in l$ iff $y = ax + b$

That is, in technical work formulas with free-variables are often used as if their variables were all bound by universal quantifiers.

This usage is captured in first-order semantics by extending by the idea of truth *simpliciter* to include open formulas. That is, we adopt a general definition for any formula *P* whether it contains free variables or not:

P is **(*simply*) *true*** relative to D and $\mathfrak{I}$ (abbreviated as $\mathfrak{A}\vDash P$ or $\mathfrak{I}^{\mathfrak{A}}_{s}(P)$=T) is defined to mean:

for all ***s***, $\mathfrak{A}_{\boldsymbol{s}}\vDash P$

or equivalently,

for all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(P)$=T

By this definition, if *P* is an open formula, it is simply true when it is true in all variable assignments. When the open formula *P* is simple true in this sense, it follows automatically that its universal closure $\forall v_1,\ldots,v_nP$ is also simply true. Conversely if the universal closure $\forall v_1,\ldots,v_nP$ is true in any variable assignment, it is true in all, and hence the open formula *P* is true *simpliciter*. That is, given the definitions of *simple truth*, the following equivalents hold for any formula *P* (open or closed):

P is (simply) true	iff	for all ***s***, $\mathfrak{A}_{\boldsymbol{s}}\vDash P$	(equivalently, $\mathfrak{I}^{\mathfrak{A}}_{s}(P)$=T)
	iff	for all ***s***, $\mathfrak{A}_{\boldsymbol{s}}\vDash\forall v_1,\ldots,v_nP$	(equivalently, $\mathfrak{I}^{\mathfrak{A}}_{s}(\forall v_1,\ldots,v_nP)$=T)
	iff	for some ***s***, $\mathfrak{A}_{\boldsymbol{s}}\vDash\forall v_1,\ldots,v_nP$	(equivalently, $\mathfrak{I}^{\mathfrak{A}}_{s}(\forall v_1,\ldots,v_nP)$=T)

(We shall prove these facts more formally later in the chapter.)

It is traditional to extend the turnstile notation to simple truth. The definition is: a formula *P is* ***true (*****simpliciter*****)*** relative to $\mathfrak{A}$=<D,$\mathfrak{I}$>, (abbreviated $\mathfrak{A}\models P$) iff, for all ***s***, $\mathfrak{A}_{\boldsymbol{s}}\models P$[42].

Let us summarize the facts so as to contrast head-on the two notions, (1) satisfaction relative to a variable assignment and (2) truth simpliciter. First, let us consider satisfaction. All formulas, both with and without variables, and both open or closed, must first have the notion of satisfaction relative to a variable assignment defined for them, for it is this notion that is the very basic notion of truth. This is the general concept that is defined inductively for all formulas. Satisfaction relative to a variable assignment is also more basic than truth-simpliciter because the former concept is actually used in the definition of the latter. But even though satisfaction must be defined for all formulas and is the more basic notion of truth, it carries a parameter that fails to mark a semantic difference for a important class of formulas, namely sentences. Formulas with no variables and those without free variables (sentences) are either satisfied in every variable assignment (relative to a model) or in none. For *sentences* there is no information content in the specification of any particular variable assignment. For these the generalization to truth-simpliciter is no more than the dropping of an uninformative parameter. If a sentence is ever satisfied by any variable assignment it is automatically true simpliciter. But for open formula there is a huge distinction between satisfaction relative to a variable assignment and truth simpliciter, *i.e.* satisfaction in relative to *all* variable assignments. Being satisfied in *some* variable assignments is no guarantee that a formula is satisfied in *all*. In fact there is a general rule that an open formula is satisfied in all variable assignments if and only if its universal closure is satisfied in all. That is, the only open formulas that are true simpliciter are those that in that world say the same thing as the universally quantified version of the formula. The custom has therefore evolved among mathematicians and logicians to speak of open formulas as simply true as an alternative way of saying its universal closure is simply true, and the usage in terms of open formulas allows one to avoid writing down a visually distracting and unnecessary list of initial universal quantifiers.

Logical Truth and Validity. Recall that each model $\mathfrak{A}$=<D,$\mathfrak{I}$> with the assignment $\mathfrak{I}$ relative to domain D is a kind of "possible world:" D specifies what exists in that world and $\mathfrak{I}$ what descriptive terms stand for. As we have said, one of these interpretations will be a close approximation to the actual world if its domain is made up of the things that actually exist and the assignment pairs familiar words to their actual referents. Likewise the set of all interpretations is an approximation to the set of all possible worlds. They are accordingly suited for use in the formal definitions of the logical idea traditionally defined in terms of possible worlds. Logical truth, for example, is traditionally distinguished by the fact that its truth is invariant across worlds: it is truth "in all possible worlds." A valid argument is likewise defined as one such that in any possible world w, if the premises are true

[42] This is another sense of the turnstile derived from the early practice of understanding a model $\mathfrak{A}$ as state description.

in w, then the conclusion is also true in w. In the theory possible worlds are represented by models. Hence, P is defined to be ***logical truth*** (which we abbreviate as $\models P$) iff, for all models $\mathfrak{A}$=<D,$\mathfrak{I}$>, $\mathfrak{A}\models P$. Likewise by definition the argument from X to P is said to be ***valid***, and that X ***logically*** or ***semantically entails*** P (abbreviated as $X\models P$) iff, for all models =<D,$\mathfrak{I}$>, if (for all Q in X, $\mathfrak{A}\models Q$) then $\mathfrak{A}\models P$.

In many logic books the notion of *validity* is extended from arguments to embrace formulas as well. When applied to formulas *validity* is just a synonym for *logical truth*. This seemingly equivocal notation is justified later by proving a metatheorem that the logical truth of a formula corresponds to the "degenerate case" in which it is a valid consequence of the empty set: $\models P$ (where $\models$ has the meaning of logical truth) holds if and only if $\varnothing\models P$ holds (where $\models$ has the meaning of valid argument). Note that that $\varnothing\models P$ is by definition a disguised universally quantified conditional, namely:

for any $\mathfrak{A}$, if (for any Q, if $Q\in\varnothing$, then $\mathfrak{A}\models Q$) then $\mathfrak{A}\models P$.

But the antecedent of this conditional, namely,

for any Q, if $Q\in\varnothing$, then $\mathfrak{A}\models Q$

is always true, because its antecedent (namely, $Q\in\varnothing$) is always false. But if the antecedent of a conditional is T, then the truth-value of the conditional is the same as that of its consequent. The consequent is this case is $\mathfrak{A}\models P$.

Hence this antecedent essentially "drops out" and the original conditional is equivalent to

for any $\mathfrak{A}$, $\mathfrak{A}\models P$.

Hence the following are all equivalent:

$\varnothing\models P$
for any $\mathfrak{A}$, if (for any Q, if $Q\in\varnothing$, then $\mathfrak{A}\models Q$) then $\mathfrak{A}\models P$
for any $\mathfrak{A}$, $\mathfrak{A}\models P$
$\models P$

With these remarks as introduction we can state the semantics of first-order logic quite succinctly.

v. Formal Statement of the Primary Semantic Definitions

The Semantics (Model Theory) for First-Order Logic. Let $\mathbf{F_{FOL}}$ be a first-order language.

A ***model*** or ***structure for basic expressions of*** $\mathbf{F_{FOL}}$ ***relative to a non-empty domain*** D and an interpretation operation $\mathfrak{I}$ is any $\mathfrak{A}$=<D,$\mathfrak{I}$>, that meets the following conditions:

- D≠∅.
- Every constant *c* is assigned by $\mathfrak{I}$ to an object in D. That is, $\mathfrak{I}(c)\in$D.
- Every *n*-place predicate is assigned by $\mathfrak{I}$ to an *n*-place relation on objects in D. There are three special cases:
  1. If *n*=1, then $\mathfrak{I}(P^1)\subseteq$D, *i.e.* a one place predicate P^1 stands for a subset of D.
  2. If *n*>2, then $\mathfrak{I}(P^n)\subseteq\mathrm{D}^n$, *i.e.* $\mathfrak{I}(P^n)$ is an *n*-place relation on elements of D, *i.e.* some set of *n*-tuples drawn from D^n. If the syntax specifies that first 2-place predicate ${P^2}_1$ is the identity predicate =, then it is required that $\mathfrak{I}({P^2}_1)$ be the identity relation on D.
  3. If *n*=0, then $\mathfrak{I}({P^0}_i)\in$\{T,F\}, *i.e.* 0-place predicates function semantically like sentence letters of sentential logic in that they do require any terms to their right and they stand for a truth-value. Furthermore, the syntax specifies that the first 0-place predicate ${P^0}_1$ is the contradiction sign ⊥. It is required that $\mathfrak{I}(\perp)$=F, *i.e.* ⊥ always takes the value F.
- Every *n*-place functor f^n is assigned by $\mathfrak{I}$ to an *n*-place function (also called an operation) on objects in D. That is, $\mathfrak{I}(f^n)\in\mathrm{D}^{\mathrm{D}^n}$.

Let $\mathfrak{A}$=<D,$\mathfrak{I}$> be a model.

A ***variable assignment*** over D for $\mathbf{F_{FOL}}$ is any function ***s*** of mapping the set of variables into D.

An ***interpretation*** $\mathfrak{I}$ ***relative to a model*** $\mathfrak{A}$=<D,$\mathfrak{I}$> ***and an assignment s of the variable over*** D for $\mathbf{F_{FOL}}$, briefly $\mathfrak{I}^{\mathfrak{A}}_s$, is defined inductively:

- **Basis Clause.** If *t* is a constant, then $\mathfrak{I}^{\mathfrak{A}}_s(t)=\mathfrak{I}(t)$.
  If *v* is a variable, then $\mathfrak{I}^{\mathfrak{A}}_s(v)=\boldsymbol{s}(v)$.
- **Inductive Clause.** If *t* is some complex term, then $f^n(t_1...t_n)$, $\mathfrak{I}^{\mathfrak{A}}_s(f^n(t_1...t_n))= \mathfrak{I}(f^n)(\mathfrak{I}^{\mathfrak{A}}_s(t_1),...,\mathfrak{I}^{\mathfrak{A}}_s(t_n))$.

P ***is satisfied in model*** $\mathfrak{A}$=<D,$\mathfrak{I}$> ***relative to an a variable assignment s over*** D for $\mathbf{F_{FOL}}$ (abbreviated equivalent as $\mathfrak{A}_{\boldsymbol{s}}\models P$ or $\mathfrak{I}^{\mathfrak{A}}_s(P)$=T) is defined inductively:

- **Basis Clause.** An atomic formula $P^nt_1,...,t_n$ is T in $\mathfrak{I}^{\mathfrak{A}}_s$, iff the objects picked out by its terms under $\mathfrak{I}^{\mathfrak{A}}_s$ (in order) stand in the relation picked out in $\mathfrak{I}$ by its predicate. In symbols

  $\mathfrak{A}_{\boldsymbol{s}}\models P^nt_1,...,t_n$ iff <$\mathfrak{I}^{\mathfrak{A}}_s t_1),..., \mathfrak{I}^{\mathfrak{A}}_s(t_n)$>$\in \mathfrak{I}(P^n)$

  or equivalently,

  $\mathfrak{I}^{\mathfrak{A}}_s(P^nt_1,...,t_n)$=T) iff < $\mathfrak{I}^{\mathfrak{A}}_s(t_1),..., \mathfrak{I}^{\mathfrak{A}}_s(t_n)$>$\in \mathfrak{I}(P^n)$
- **Inductive Clauses.** The satisfaction of a molecular formula relative to variable assignment is broken down into case one for each formation rule of the syntax:

$\mathfrak{A}_{\boldsymbol{s}}\models\sim P$ iff not $\mathfrak{A}_{\boldsymbol{s}}\models P$; or	$\mathfrak{I}^{\mathfrak{A}}_s(\sim P)$=T iff $\mathfrak{I}^{\mathfrak{A}}_s$ $P)\neq$T
$\mathfrak{A}_{\boldsymbol{s}}\models P\wedge Q$ iff ($\mathfrak{A}_{\boldsymbol{s}}\models P$ and $\mathfrak{A}_{\boldsymbol{s}}\models Q$); or	$\mathfrak{I}^{\mathfrak{A}}_s(P\wedge Q)$=T iff, $\mathfrak{I}^{\mathfrak{A}}_s(P)$=T and $\mathfrak{I}^{\mathfrak{A}}_s(Q)$=T
$\mathfrak{A}_{\boldsymbol{s}}\models P\vee Q$ iff ($\mathfrak{A}_{\boldsymbol{s}}\models P$ or $\mathfrak{A}_{\boldsymbol{s}}\models Q$);or	$\mathfrak{I}^{\mathfrak{A}}_s(P\vee Q)$=T iff, $\mathfrak{I}^{\mathfrak{A}}_s(P)$=T or $\mathfrak{I}^{\mathfrak{A}}_s(Q)$=T
$\mathfrak{A}_{\boldsymbol{s}}\models P\rightarrow Q$ iff (not $\mathfrak{A}_{\boldsymbol{s}}\models P$ or $\mathfrak{A}_{\boldsymbol{s}}\models Q$); or	$\mathfrak{I}^{\mathfrak{A}}_s(P\rightarrow Q)$=T iff, $\mathfrak{I}^{\mathfrak{A}}_s(P)\neq$T or $\mathfrak{I}^{\mathfrak{A}}_s(Q)$=T
$\mathfrak{A}_{\boldsymbol{s}}\models P\leftrightarrow Q$ iff ($\mathfrak{A}_{\boldsymbol{s}}\models P$ iff $\mathfrak{A}_{\boldsymbol{s}}\models Q$);or	$\mathfrak{I}^{\mathfrak{A}}_s(P\leftrightarrow Q)$=T iff, $\mathfrak{I}^{\mathfrak{A}}_s(P)$=T iff $\mathfrak{I}^{\mathfrak{A}}_s(Q)$=T
$\mathfrak{A}_{\boldsymbol{s}}\models\forall xPx$ iff for any *x*-variant ***s′*** of ***s***, $\mathfrak{A}_{\boldsymbol{s'}}\models Px$, or	$\mathfrak{I}^{\mathfrak{A}}_s(\forall xPx)$=T iff for any *x*-variant ***s′*** of ***s***, $\mathfrak{I}^{\mathfrak{A}}_s(Px)$=T

P ***is true (*simpliciter*)*** *in* $\mathfrak{A}$=<D,$\mathfrak{I}$> (abbreviated $\mathfrak{A}\models P$) iff, for all ***s*** over D, $\mathfrak{A}_{\boldsymbol{s}}\models P$.

The existential quantifier is introduced by definition, and its truth-conditions directlu follow from the above definitions:

The Existential Quantifier

$\exists vP$ abbreviates $\sim\forall \mathrm{v}\sim \mathrm{P}$

$\mathfrak{I}^{\mathfrak{A}}_{s}(\exists xPx)$=T iff for some x-variant $\boldsymbol{s'}$ of $\boldsymbol{s}$, $\mathfrak{I}^{\mathfrak{A}}_{s}(Px)$=T

Proof. The proof consists of applying the above definition, the clauses for the satisfaction-conditions of the universal quantifier and negation, and logical equivalents for quantifiers and negations in the metalanguage: $\mathfrak{I}^{\mathfrak{A}}_{s}(\exists xPx)$=T iff [by the abbreviation] $\mathfrak{I}^{\mathfrak{A}}_{s}(\sim\forall x\sim Px)$=T iff [by satifaction conditions for negation] not($\mathfrak{I}^{\mathfrak{A}}_{s}(\forall x\sim Px)$=T) iff [by the satisfaction conditions for the universal quantifier] not(for all x-variant $\boldsymbol{s'}$ of $\boldsymbol{s}$, $\mathfrak{I}^{\mathfrak{A}}_{s}(\sim Px)$=T) iff [by logical equivalence for quantifiers in the metalanguage] for some all x-variant $\boldsymbol{s'}$ of $\boldsymbol{s}$, not $\mathfrak{I}^{\mathfrak{A}}_{s}(\sim Px)$=T) iff [by satisfaction conditions for negation] for some all x-variant $\boldsymbol{s'}$ of $\boldsymbol{s}$, not not $\mathfrak{I}^{\mathfrak{A}}_{s}(Px)$=T) iff [by a truth-functional equivalence in the metalanguage] for some all x-variant $\boldsymbol{s'}$ of $\boldsymbol{s}$, $\mathfrak{I}^{\mathfrak{A}}_{s}(Px)$=T. **QED.**

These are the definitions which we shall use in the rest of the chapter because they are customary and likely to be found in other logic books. It is useful, however, to reformulate the definitions now that they are fresh in a way that exhibits more clearly their inductive structure. Let us pause to recast the notion of the interpretation of terms, which we shall call her Ref, and satisfaction, which we shall call here Sat, in a way that makes explicit that their definitions are inductive and that what is being defined in each case is really a 5-place relation. That is, let us make clear that what is being defined is a set of 5-tuples. Each 5-tuple begins with its "arguments". These consists of an expression plus its three "parameters" – a domain, an interpretation, and a variable assignment. The five tuple concludes with its "value", the referent or truth value that is paired with its argument by that relation. If $<t,\mathrm{D},\mathfrak{I},\boldsymbol{s},d>\in$Ref, then Ref assigns the argument $<t,\mathrm{D},\mathfrak{I},\boldsymbol{s}>$ the value d. If $<P,\mathrm{D},\mathfrak{I},\boldsymbol{s},\mathrm{T}>\in$Sat, then Sat assigns the argument $<P,\mathrm{D},\mathfrak{I},\boldsymbol{s}>$ the value T. Since the 5-place relation assigns a unique value to its 4-place argument, it is a 4-place function, and it is correct reformulate $<t,\mathrm{D},\mathfrak{I},\boldsymbol{s},d>\in$Ref as Ref($t$,D,$\mathfrak{I}$,$\boldsymbol{s}$)=$d$, and $<P,\mathrm{D},\mathfrak{I},\boldsymbol{s},\mathrm{T}>\in$Sat as Sat($P$,D,$\mathfrak{I}$,$\boldsymbol{s}$)=T.

Ref is defined inductively as follows:
1. Basis Clause. $\{<t,\mathrm{D},\mathfrak{I},\boldsymbol{s},d>|$ $<\mathrm{D},\mathfrak{I}>$ is a model, $\boldsymbol{s}$ is a variable assignment for $<\mathrm{D},\mathfrak{I}>$ and $d=\mathfrak{I}(t)$ if t is a constant and $d=\boldsymbol{s}\,(t)$ if t is a variable$\}\subseteq$ Ref.
2. Inductive Step.
$\{<f^n(t_1...t_n)),\mathrm{D},\mathfrak{I},\boldsymbol{s},d>$ | $<t_1,\mathrm{D},\mathfrak{I},\boldsymbol{s},d_1>\in$ Ref, …, $<t_n,\mathrm{D},\mathfrak{I},\boldsymbol{s},d_n>\in$ Ref, and $\mathfrak{I}(f^n)(d_1,...,d_n)=d\ \}\subseteq$ Ref**.**
3. Nothing else is in Sat.
$\mathfrak{I}^{\mathfrak{A}}_{s}(t)=d$ relative to a model $<\mathrm{D},\mathfrak{I}>$ is then defined as $<t,\mathrm{D},\mathfrak{I},\boldsymbol{s},d>\in$ Ref.

Sat is the defined inductively as follows:
1. Basis Clause. $\{<P^nt_1,...,t_n,\mathrm{D},\mathfrak{I},\boldsymbol{s},\mathrm{V}>|$ $<\mathrm{D},\mathfrak{I}>$ is a model, $\boldsymbol{s}$ is a variable assignment for $<\mathrm{D},\mathfrak{I}>$ & either $<\mathfrak{I}^{\mathfrak{A}}_{s}(t_1),...,\mathfrak{I}^{\mathfrak{A}}_{s}(t_n)>\in\mathfrak{I}(P^n)$ and V=T, or $<\mathfrak{I}^{\mathfrak{A}}_{s}(t_1),...,\mathfrak{I}^{\mathfrak{A}}_{s}(t_n)>\notin\mathfrak{I}(P^n)$ and V=F$\}\subseteq$Sat.
2. Inductive Clause. There are

- If $<P,\mathrm{D},\mathfrak{I},\boldsymbol{s},\mathrm{F}>\in$Sat, then $<\sim P,\mathrm{D},\mathfrak{I},\boldsymbol{s},\mathrm{T}>\in$ Sat.

- If <P,D,ℑ,s,T>∈Sat, then <∼P,D,ℑ,s,F>∈ Sat.
- If <P,D,ℑ,s,T>∈ Sat and <P,D,ℑ,s,T>∈ Sat, then <P∧Q,D,ℑ,s,T>∈ Sat.
- If <P,D,ℑ,s,F>∈ Sat or <P,D,ℑ,s,F>∈ Sat, then <P∧Q,D,ℑ,s,F>∈ Sat.
- If <P,D,ℑ,s,T>∈ Sat or <P,D,ℑ,s,T>∈ Sat, then <P∨Q,D,ℑ,s,T>∈ Sat.
- If <P,D,ℑ,s,F>∈ Sat and <P,D,ℑ,s,F>∈ Sat, then <P∨Q,D,ℑ,s,F>∈ Sat.
- If <P,D,ℑ,s,F>∈ Sat or <P,D,ℑ,s,T>∈ Sat, then <P→Q,D,ℑ,s,T>∈ Sat.
- If <P,D,ℑ,s,T>∈ Sat and <P,D,ℑ,s,F>∈ Sat, then <P→Q,D,ℑ,s,F>∈ Sat.
- If <P,D,ℑ,s,T>∈Sat and <P,D,ℑ,s,T>∈Sat, or <P,D,ℑ,s,F>∈Sat and <Q,D,ℑ,s,F>∈Sat, then <P↔Q,D,ℑ,s,T>∈ Sat.
- If <P,D,ℑ,s,T>∈Sat and <P,D,ℑ,s,F>∈ ⊨, or <P,D,ℑ,s,F>∈Sat, and <Q,D,ℑ,s,T>∈Sat, then <P↔Q,D,ℑ,s,F>∈ Sat.
- If {<P,D,ℑ,s′,T>| s′is a *x*-variant of s}⊆Sat, then <∀xP,D,ℑ,s,T>∈ Sat.
- If {<P,D,ℑ,s′,F>| s′is a *x*-variant of s}is not empty and is a subset of Sat, then <∀xP,D,ℑ,s,F>∈ Sat.

3. Nothing else is in Sat.

$\mathfrak{A}_{s}\models P$ or, equivalently $\mathfrak{I}^{\mathfrak{A}}_{s}(P)$=T*,* relative to a model 𝔄=<D,ℑ>, is then defined as <P,D,ℑ,s,*d*>∈Sat.

The virtue of this formulation is that it exhibits how an inductive condition can in principle be formulated so that it draws on facts about the 5-tuples associated of component expressions that range beyond a single domain, interpretation or model. In particular, the clause for the universal quantifier declares that <∀xP,D,ℑ,s,T> is in Sat only if a range of 5-tuples <P,D,ℑ,s′,T> are in Sat, where the various s′ in question meet a certain condition (*viz.* that they are *x*-variants of s). In modal, tense, intensional, and indexical logics – all of which introduce new syntactic "wholes" and "parts" not found in the syntax of FOL, some of which we shall meet in Chapter 4 – have inductive clauses admitting a *n*-tuple for a "whole" that draw on facts about *n*-tuples for "parts" that not only a range over variable assignments but over models and other relevant indices.

Let us now introduce the rigorous definitions for the semantic versions of "logical" concepts – logical truth, validity, and satisfiability. These are defined in terms of models.

- *P* is an FOL ***logical truth*** (abbreviated $\models_{\mathbf{FOL}} P$) iff, for all models 𝔄, $\mathfrak{A}\models P$.
- An argument from the (possibly infinite) set X of premises to conclusion *P* is ***valid*** in FOL (abbreviated $X\models_{\mathbf{FOL}} P$) means for all models 𝔄, if (for any *Q,* if Q∈*X,* then $\mathfrak{A}\models Q$) then $\mathfrak{A}\models P$.
  If *X* is some finite set we usually drop the brackets and rewrite $\{P_1,...P_n,...\}\models_{\mathbf{FOL}} Q$ as $P_1,...P_n,...\models_{\mathbf{FOL}} Q$.
  It is common to abbreviate the fact that 𝔄 assigns T to all formulas in *X, i.e.* the fact that (for any *Q,* if Q∈*X,* then $\mathfrak{A}\models Q$), by the locution "𝔄 ***satisfies*** *X*". Using this terminology, $X\models_{\mathbf{FOL}} P$ iff, for all 𝔄, 𝔄 satisfies *X.*
- *P* is FOL ***satisfiable*** iff, for some 𝔄, $\mathfrak{A}\models P$.
- A set *X* of formulas is ***satisfiable*** iff, there is some 𝔄 such that for all *P* in *X*, $\mathfrak{A}\models P$.

Example. $\forall xFx\models Fc$

Proof. Let 𝔄=<D,ℑ> be arbitrary and assume for conditional proof that $\mathfrak{A}\models\forall xFx$. Then by definition, for any s, $\mathfrak{A}_{s}\models\forall xFx$ or in alternative notation that $\mathfrak{I}^{\mathfrak{A}}_{s}(\forall xFx)$=T. Let us instaniate for s. Hence,

$\Im^{\mathfrak{A}}_{s'}(\forall xFx)$=T. Hence by definition, for any *x*-variant ***s′′*** of ***s′***, $\Im^{\mathfrak{A}}_{s''}(Fx)$=T. Since this is universally true for all such *x*-variants, let us instantiate for that *x*-variant, call it ***s″***, such that ***s″***(*x*)=$\Im(c)$. Hence $\Im^{\mathfrak{A}}_{s''}(Fx)$=T. Then by definition $\Im^{\mathfrak{A}}_{s''}(x)\in\Im(F)$. Since by definition e have stipulated that ***s″***(*x*)= $\Im^{\mathfrak{A}}_{s}(c)$, by substitutivity of identity, $\Im(c)\in\Im(F)$. Let us consider an arbitray ***s‴***. Now, for any ***s*** it is true by definition that $\Im(c)$= $\Im^{\mathfrak{A}}_{s}(c)$. Let us instantiate this fact for ***s‴***: $\Im(c)$= $\Im^{\mathfrak{A}}_{s'''}$. Hence by substitutivity of identiy, $\Im^{\mathfrak{A}}_{s'''}(c)\in\Im(F)$. Hence by definition, $\Im^{\mathfrak{A}}_{s'''}(Fc)$=T. Since ***s‴*** is arbitray, we may universally generalize, for all ***s***, $\Im^{\mathfrak{A}}_{s}(Fc)$=T, or in alternative notation $\mathfrak{A}_{\boldsymbol{s}}\models Fc$. Hence, by definition, $\mathfrak{A}\models Fc$. Since $\mathfrak{A}$ is arbitrary, we may generalize: for all models $\mathfrak{A}$, if $\mathfrak{A}\models\forall xFx$ then $\mathfrak{A}\models Fc$. Hence by definition, $\forall xFx\models Fc$. **QED.**

Example. $\forall x(Gx\to Hx), \forall x(Fx\to Gx) \not\models \exists x(Fx\wedge Hx)$

Proof. Define $\mathfrak{A}$=<D,$\Im$> such that D={1,2}, and $\Im(F)=\varnothing$ and $\Im(G)=\varnothing$ and $\Im(H)$={1}. We show three propositions:

(1) $\mathfrak{A}\models\forall x(Fx\to Gx)$.
(2) $\mathfrak{A}\models\forall x(Gx\to Hx)$.
(3) $\mathfrak{A}\not\models\exists x(Fx\wedge Hx)$

(1) Let ***s*** be an arbitray variable assignment over D, and let ***s′*** be an arbitray *x*-variant of ***s***. Hence $\Im^{\mathfrak{A}}_{s'}$ ***s′***(*x*)$\notin\Im(F)$. Now by definition, ***s′***(*x*)= $\Im^{\mathfrak{A}}_{s'}(x)$. Hence by substitutivity of identity, $\Im^{\mathfrak{A}}_{s'}(x)\notin\Im(F)$. Hence by definition $\Im^{\mathfrak{A}}_{s'}(Fx)\neq$T. Hence by truth-functional login in the metalanguage, either $\Im^{\mathfrak{A}}_{s'}(Fx)\neq$T or $\Im^{\mathfrak{A}}_{s'}(Gx)$=T. Hence, by definition, $\Im^{\mathfrak{A}}_{s'}(Fx\to Gx)$=T. Since ***s′*** is an arbitray *x*-variant of ***s,*** we may generalize, for any *x*-variant ***s′*** of ***s***, $\Im^{\mathfrak{A}}_{s'}(Fx\to Gx)$=T. Hence by definition, $\Im^{\mathfrak{A}}_{s}\forall x\,(Fx\to Gx)$=T. Since ***s*** is arbitray, we may generalize, for any variable assignment ***s*** of $\mathfrak{A}$, $\Im^{\mathfrak{A}}_{s}\forall x\,(Fx\to Gx)$=T. Hence by definition, $\Im^{\mathfrak{A}}_{s}\forall x\,(Fx\to Gx)$=T, or in alternative notation $\mathfrak{A}\models\forall x(Fx\to Gx)$.

(2) is shown *mutatis mutandis*, by replacing *F* with *G,* and *G* with *H,* in (1).

(3) Let us define ***s*** to a the arbitray variable assignment over D, such that it assigns all variables to 2, *i.e.* ***s***(*x*)=2 (it doesn't matter how it is defined really). Now let ***s′*** be an arbitrary *x*-variant of ***s***: Clearly, since $\Im(F)=\varnothing$, ***s′***(*x*)$\notin\Im(F)$. Moreover, by definition, $\Im^{\mathfrak{A}}_{s'}(x)$=***s′***(*x*). Hence $\Im^{\mathfrak{A}}_{s'}(x)\notin\Im(F)$. Hence, by definition, $\Im^{\mathfrak{A}}_{s'}(Fx)\neq$T. not ($\Im^{\mathfrak{A}}_{s'}(Fx)$=T and $\Im^{\mathfrak{A}}_{s'}(Hx)$=T). Hence, not ($\Im^{\mathfrak{A}}_{s'}(Fx\wedge Hx)$=T). Now since ***s′*** is an arbitrary *x*-variant of ***s,*** we may generalize from the case of ***s′***: for any *x*-variant ***s′*** of ***s***, not($\Im^{\mathfrak{A}}_{s'}(Fx\wedge Hx)$=T). Further, we may existentially generalize over ***s*** to get: for some ***s***, for any *x*-variant ***s′*** of ***s***, not($\Im^{\mathfrak{A}}_{s'}(Fx\wedge Hx)$=T). But this is logically equivalent in the metalanguage to the fact that not (for all ***s***, for some *x*-variant ***s′*** of ***s***, $\Im^{\mathfrak{A}}_{s'}(Fx\wedge Hx)$=T). That is, by definition, not (for all ***s***, $\Im^{\mathfrak{A}}_{s'}\exists x(Fx\wedge Hx)$=T), which in turn by definition means not($\mathfrak{A}\models\exists x(Fx\wedge Hx)$).

Since (1)-(3) hold, we may existentially generalize from the case of $\mathfrak{A}$ and conclude that there is some model in which the premises of the argument are true but the conclusion false. Hence it is invalid. **QED.**

C. Proof Theory

i. Axiom Systems

Proof theory is that branch of syntax that studies how valid arguments may be described in entirely formal terms, *i.e.* in terms of the shapes and physical properties of signs. In the earliest attempts at proof theory it was the notion of logical truth that was captured, and the tool for doing so was an axiom system specified in a completely syntactic manner. Let us begin a classical example, the axiomatization of "tautology" in sentence logic.

Example. An Axiom System for the Tautologies of Sentence Logic.

Let P and Q be formulas of some sentential language **SL**.
The set $\vdash_{\mathbf{SL}}$ of **theorems *of* SL** is defined inductively. It is the smallest set containing the axioms and closed under a single "construction rule", which called an "inference rule" in an axiom system, *modus ponens.*

1. Basis Clause. The set of $\mathbf{Ax_{SL}}$ of axioms for **SL**, which is defined as the set of all instances of the axiom schemata 1-3, is a subset of $\vdash_{\mathbf{SL}}$.

1. $\vdash_{\mathbf{SL}} P\to(Q\to P)$
2. $\vdash_{\mathbf{SL}}(P\to(Q\to R))\to((P\to Q)\to(P\to R))$
3. $\vdash_{\mathbf{SL}}(\sim P\to\sim Q)\to(Q\to P)$ **Basis Clause.** The set $\mathbf{Ax_{SL}}$ is a subset of $\vdash_{\mathbf{SL}}$

2. Inductive Clause, *Modus Ponens:* If $P\in\vdash_{\mathbf{SL}}$ and $P\to Q\in\vdash_{\mathbf{SL}}$, then $Q\in\vdash_{\mathbf{SL}}$.
It is more conventional (but less transparent) to express the fact that $P\in\vdash_{\mathbf{SL}}$ by the notation $\vdash_{\mathbf{SL}}P,$ and then to state the rule using this notation:

If $\vdash_{\mathbf{SL}}P$ and $\vdash_{\mathbf{SL}}P\to\mathrm{Q}$, then $\vdash_{\mathbf{SL}}Q$

3. Nothing else is in $\vdash_{\mathbf{SL}}$

We say a sentence Q ***follows deductively from*** a finite set $\{P_1,...,P_n\}$ of formulas in $\vdash_{\mathbf{SL}}$ for **SL** (abbreviated $P_1,...,P_n\vdash_{\mathbf{SL}}Q$) iff $(P_1\to(P_2\to...,P_n))\to Q\in\vdash_{\mathbf{SL}}$.
We extend the notion of deduction to possibly infinite sets of premises X by saying $X\boldsymbol{\vdash}_{\mathbf{SL}}Q$ iff, there is some finite subset $\{P_1,...,P_n\}$ of X such that $P_1,...,P_n\vdash_{\mathbf{SL}}Q$.

Notice that the "single" turnstile $\vdash$ now has several meanings when used in connection with axiom systems. Its first meaning is to name the set of theorems $\vdash_{\mathbf{SL}}$. It is used to say that a sentence is in the set of theorems. Thus, $\vdash_{\mathbf{SL}}A\vee\sim A$ is another way of saying $A\vee\sim A\in\vdash_{\mathbf{SL}}$. The second usage of $\vdash$, however, is introduced above. In this usage $\{P_1,...,P_n\}\vdash_{\mathbf{SL}}Q$ the turnstile is flanked on the left by a set of sentences $\{P_1,...,P_n\}$ and on the right by a single sentence Q. Intuitively, it asserts that the relation of "syntactic deducibility" holds between the two as premises and conclusions. But really, in axiom systems this talk is strictly speaking somewhat misleading because an argument is "proven" indirectly by showing that its corresponding conditional is a theorem. That is, strictly speaking the fact expressed as $\{P_1,...,P_n\}\vdash_{\mathbf{SL}}Q$ really means $(P_1\to(P_2\to...,P_n))\to Q\in\vdash_{\mathbf{SL}}$. For example, $\{A,A\to B\}\vdash_{\mathbf{SL}}B$ says that the sentence $(A\wedge(A\to B))\to A$ is in $\vdash_{\mathbf{SL}}$.

Note that $\vdash$ is used only for finite premise sets. Strictly speaking, in the notation $\{P_1,...,P_n\}\vdash Q$, using the small (non-bold) turnstile $\vdash$, the notation is correct (well defined) only if the number of sentences to the left is finite, because the notation is short for $\vdash_{\mathbf{SL}}(P_1\rightarrow(P_2\rightarrow...,P_n))\rightarrow Q$ in which is the single sentence $(P_1\rightarrow(P_2\rightarrow...,P_n))\rightarrow Q$ is made up of only a finite number of signs. It is this limitation that motivates the introduction of the boldface turnstile $\boldsymbol{\vdash}$, which must be distinguished from is cousins. It is used when premises set might be infinite. Thus, $X\boldsymbol{\vdash}_{\mathbf{SL}}Q$ is defined in terms of $\vdash$, and says that there is some finite subset of $\{P_1,...,P_n\}$ of X , such that $\{P_1,...,P_n\}\vdash_{\mathbf{SL}}Q$.

We now introduce vocabulary for extending the set of theorems to include "non-logical axioms," as we do in science when we supplement the laws of math and logic with those of physics and chemistry. These latter cannot be proved from logic and math alone (otherwise the empirical sciences would merely be branches of mathematics), but must be added to the original axioms. Such an expanded set is called a "theory."

Th is a ***sentential theory*** in **L** with non-logical axioms **A** iff **Th** is the set of all $\boldsymbol{\vdash}_{\mathbf{SL}}$ consequences of **A**, *i.e.*

$$\mathbf{Th} = \{P \mid \mathbf{A}\boldsymbol{\vdash}_{\mathbf{SL}}P\}$$

Let us now switch languages and talk about the same syntactic ideas in the richer syntax of first-order logic. Its set of local axioms includes those of sentential logic but also two axioms to capture the logical truths that depend on the meaning of subject-predicate assertions and the quantifiers *all* and *some*.

Example. An Axiom System for the Logical Truths of First-Order Logic.

$\vdash_{\mathbf{FOL}}$ (the constructive) set of theorems *of* FOL is defined inductively as the smallest set containing the axioms of FOL and closed under *modus ponens:*

1. Basis Clause. The set $\mathbf{Ax}_{\mathbf{FOL}}$ of axioms for a first-order language, which is defined as the set of all instances of the axiom schemata 1-6, is a subset of $\vdash_{\mathbf{FOL}}$. Schemata 1-3 are the same as those above for $\mathbf{Ax}_{\mathbf{SL}}$..

4. $\vdash_{\mathbf{FOL}}\ \forall x(P\rightarrow Q)\rightarrow(\forall xP\rightarrow\forall xQ)$
5. $\vdash_{\mathbf{FOL}}\ P\rightarrow\forall xP$ where x is not free in P
6. $\vdash_{\mathbf{FOL}}\forall xP[x]\rightarrow P[y]$ where $P[y]$ is like $P[x]$ except for containing free occurrences of y where $P[x]$ contains free occurrences of x

2. Inductive Clause, *Modus Ponens.* If $\vdash_{\mathbf{FOL}}P$ and $\vdash_{\mathbf{FOL}}P\rightarrow$Q, then $\vdash_{\mathbf{FOL}}Q$

3. Nothing else is in $\vdash_{\mathbf{FOL}}$

We say a sentence Q ***follows deductively from*** a finite set $\{P_1,...,P_n\}$ relative to $\vdash_{\mathbf{FOL}}$ and **L** (abbreviated $P_1,...,P_n\vdash_{\mathbf{FOL}}Q$) iff $(P_1\rightarrow(P_2\rightarrow...,P_n))\rightarrow Q$ is in $\vdash_{\mathbf{FOL}}$.

We extend the notion to possibly infinite sets of premises by saying $X\boldsymbol{\vdash}_{\mathbf{FOL}}Q$ iff there is some finite subset $\{P_1,...,P_n\}$ of X such that $P_1,...,P_n\vdash_{\mathbf{FOL}}Q$.

We have defined $\vdash_{\mathbf{FOL}}$ so that if $X \vdash_{\mathbf{FOL}} Q$ holds only if there is a finite subset $\{P_1,...,P_n\}$ of X from which the conclusion Q can be deduced, *i.e.* $P_1,...,P_n \vdash_{\mathbf{FOL}} Q$. It follows by definition (i.e. trivially) that any deduction is provable only from a finite number of premises.

Metatheorem 1-2

(Finite Deductibility) X $\vdash_{\mathbf{FOL}}$Q iff there is some finite subset $\{P_1,...P_n\}$ of X such that $P_1,...P_n \vdash_{\mathbf{FOL}} Q$.

First-Order Theories. Historically, the great expressive power of first-order syntax combined with the possibility its axiomatization made a huge impression on scientists both in and out of mathematics in the early decades of this century. Any scientific theory, it was thought, could be put into first-order syntax and its "laws" added to its logical axioms to generate all the truths of the science as the deductive consequences of this enlarged set of "axioms." Recall that an extension of an axiom system beyond the axioms of logic is called a *theory*.

Definition

A set **Th** is a ***first-order theory*** with non-logical axioms **A** iff **Th** = $\{P \mid \mathbf{A} \vdash_{\mathbf{FOL}} P\}$.

Let us illustrate the notion of theory by two important examples. The first extends the axioms of logic to include those of identity. The second extends them further to include the laws ("axioms") of set theory in the manner of Russell.

For identity theory we employ a first-order syntax with the special two-place predicate =.

Example of a First-Order Theory: The Truths of First-Order Logic with Identity.

The set $\vdash_{\mathbf{FOL=}}$ is the set defined inductively as the smallest set containing the axioms below and closed under *modus ponens.*

1. Basis Clause. The set $\mathbf{Ax_{FOL=}}$ of axioms for ***first-order logic with identity*** consists of all instances of the axiom schemata 1-8. Axiom schemas 1-6 are given above.

7. $\vdash_{\mathbf{FOL=}} x=x$

8. $\vdash_{\mathbf{FOL=}} x=y \wedge P \rightarrow P[y//x]$

2. Inductive Clause, *Modus Ponens.* If $\vdash_{\mathbf{FOL=}} P$ and $\vdash_{\mathbf{FOL=}} P \rightarrow Q$, then $\vdash_{\mathbf{FOL=}} Q$

3. Notion else is in $\vdash_{\mathbf{FOL=}}$.

For set theory let us first specify the syntax $\mathrm{L}_{\mathbf{PM}}$. This is a first-order syntax with identity that has no constants or functors, and only the following predicates (in addition to =): the two-place predicate, and a countably infinite series of one-place predicates $P_1,\ldots,P_n,\ldots$ (called ***type predicates***). Let us the subscripts of these predicate be called ***type indices***, and we let $^{\tau}$ range over these indices. Let $Q[x]$ be any formula containing the variable x. We introduce by definition a special sort of variable for each type, Intuitive, P_{τ} is a predicate that is true of the entities in the type $^{\tau}$. We now define quantified expressions using type variables x^{τ} that "range over" entities in the type $^{\tau}$.

Definition of Type-Variables

$\forall x^{\tau}Q[x^{\tau}]$ means $\forall x(P_{\tau}x \rightarrow Q[x])$

$\exists x^{\tau}Q[x^{\tau}]$ means $\exists x(P_{\tau}x \wedge Q[x])$

In general, a first-order syntax with specially variables defined in this manner by restriction to specific predicates is called a ***many-sorted logic*** and the predicates are called ***sortal predicates***. Accordingly, the syntax $\mathrm{L}_{\mathbf{PM}}$ of the simple theory of types is an example of a theory written in a many-sorted syntax. Its axiom set consists of all the logical axioms of first-order logic with identity plus the two axioms that Russell uses to formulate Cantor's key intuitions about sets -- in a way that avoids the paradoxes.

Example of a First-Order Theory. The Axiom System PM, the Simple Theory of Types. (Modeled on Whitehead and Russell, 1910) [43]

$\vdash_{\mathbf{PM}}$ is the set inductively defined as the closure of its axioms given below under modus ponens:

1. Basis Clause. The set $\mathbf{Ax_{PM}}$ of axioms of $\vdash_{\mathbf{PM}}$, which is defined as the set of (1) all instances of the 1-8 (of $\mathbf{Ax_{FOL=}}$) altered so that x is everywhere replaced by x^{τ} and y by y^{τ} and (2) all instances of the schemata 9 and 10, is a subset of $\vdash_{\mathbf{PM}}$.

9. $\vdash_{\mathbf{PM}}\forall y^{\tau}\,(y^{\tau}\in\{x^{\tau}P[x^{\tau}]\}^{\tau+1}\leftrightarrow P[y^{\tau}])$, for any $P[x^{\tau}]$

10. $\vdash_{\mathbf{PM}}\,\forall x^{\tau+1}\forall y^{\tau+1}[x^{\tau+1}=y^{\tau+1}\leftrightarrow\forall z^{\tau}\,(z^{\tau}\in x^{\tau+1}\leftrightarrow z\in y^{\tau+1})]$

2. Inductive Clause, *Modus Ponens.* If $\vdash_{\mathbf{PM}} P$ and $\vdash_{\mathbf{PM}} P\rightarrow Q$, then $\vdash_{\mathbf{PM}} Q$.

3. Notion else is in **PM**.

ii. Natural Deduction

In the 1930's the German logician Gerhard Gentzen proposed a reorientation of proof theory. The proper study of logic on his view is not the truths of logic, but valid arguments. Logical truths, he observed, are really a special case of valid arguments in any case. *P* is a logical truth iff its opposite logically implies a contradiction. Indeed it is possible to prove several equivalent formulations.

[43] Strictly if the syntax of set theory is limited to the primitive set of descriptive predicates ∈ and =, then the notation for set abstract $\{x \mid P[x]\}$ must be introduced by defintion, as would be the type superscripts. Doing so is straight forward, and axioms 9 and 10 as written above then follow as theorems. We begin by partitioning the set $\mathbf{Vbls}_{\mathcal{FOL}}$ of variables into a denumerable series of denumerable subsets $\mathbf{Vbls}_{\mathcal{FOL}}(1)$,…, $\mathbf{Vbls}_{\mathcal{FOL}}(n)$, one for each type. We let τ rang over 1,…,*n*,…, and affix to each variable in $\mathbf{Vbls}_{\mathcal{FOL}}(\tau)$ a superscript τ. Intuitively $\mathbf{Vbls}_{\mathcal{FOL}}(\tau)$ is the set of variables of type τ. We then state a version of axiom 9 using only the primative notation:

9*. $\vdash_{\mathcal{PM}}\exists\, x^{\tau+1}\forall y^{\tau}\,(y^{\tau}\in x^{\tau+1}\leftrightarrow P[y^{\tau}])$, for any $P[x^{\tau}]$

We must also augment this axiom with two eliminative definitions. The first defines the notation $1x^{\tau}P[x^{\tau}]$, called a ***definite description***. $1x^{\tau}P[x^{\tau}]$ functions like a singular term and is read "the one and only x^{τ} such that $P[x^{\tau}]$". It is always used combined with predicates or open sentences as in $Q[1x^{\tau}P[x^{\tau}]]$ which is read "the one and only x^{τ} such that $P[x^{\tau}]$ is such that $Q[x^{\tau}]$". In the current syntax, it is used only in assertions of membership, to say that something is an element of $1x^{\tau}P[x^{\tau}]$ or that it is an element itself of a set . The two cases are defined as follows:

Definitions: $y^{\tau}\in 1x^{\tau+1}P[x^{\tau+1}]$ $=_{\text{def}}$ $\exists x^{\tau+1}\,(Px^{\tau+1}\wedge\forall z^{\tau+1}\,(Pz^{\tau+1}\rightarrow z^{\tau+1}=x^{\tau+1})\wedge y^{\tau}\in x^{\tau+1})$
$1x^{\tau}P[x^{\tau}]\in y^{\tau+1}$ $=_{\text{def}}$ $\exists x^{\tau}\,(Px^{\tau}\wedge\forall z^{\tau}\,(Pz^{\tau}\rightarrow z^{\tau}=x^{\tau})\wedge x^{\tau}\in y^{\tau+1})$

That is, $y^{\tau}\in 1x^{+1}P[x^{\tau+1}]]$ means there is an $x^{\tau+1}$ such that (i) it satisfies the open sentence $P[x^{\tau+1}]$, (ii) it is the only $x^{\tau+1}$ such that $P[x^{\tau+1}]$, and (iii) this $x^{\tau+1}$ is such that $y^{\tau}\in x^{\tau+1}$. Using this notation we are now able to introduce by eliminative definition the set abstract notation itself:

Definition: $\{x^{\tau}\,|P[x^{\tau}]\}^{\tau+1}$ $=_{\text{def}}$ $1y^{\tau+1}(\,x^{\tau}\in y^{\tau+1}\leftrightarrow P[x^{\tau}])$

From 9* and the definitions it is then possible to prove 9 as a theorem.

Metatheorem 1-3. The following are equivalent and interdeducible:

$\vDash P$	P is a *logical truth* (or P is *valid*)
$\sim P \vDash \bot$	$\sim P$ logically implies a contradiction
$\varnothing \vDash P$	P follows ("vacuously") from no premises.

Axiomatic theories, moreover, were shown by Gödel to be poor explications of mathematical method. Why should logic be lumbered with it? Rather the proper study of logic is *arguments*. Proof theory accordingly should direct its syntactic resources primarily towards explaining arguments rather than logical truths. If arguments are explained, then so will be the special case of logical truths. He proposed, therefore, to provide a purely syntactic constructive account of "good argument."

Accordingly, Gentzen proposed formal proof rules that do two things. First, they yield a purely syntactic constructive definition of the intuitively valid arguments. Second, they explain how to *use* connectives and quantifiers in proofs.

Let us review the construction. It is, first of all, a set of arguments. But what is an *argument*? We shall consider it to be an ordered pair consisting of a finite string of premises $P_1,...,P_n$ and a conclusion Q. That is, an argument or, as it is customary to call it in natural deduction theory, a ***deduction,*** is understood to be a pair of the form $<\{P_1,...,P_n\}, Q>$ such that $P_1,...,P_n$ are its premises and Q is its conclusion.

Now, some arguments are valid and others invalid. In sentential logic, $<\{P,P\rightarrow Q\},Q>$ is valid but $<\{Q,P\rightarrow Q\},P>$ is invalid. Let us use $\vDash$ as the name for the set of valid arguments as that idea is defined semantically, *i.e.* $<\{P_1,...,P_n\}, Q>\in\vDash$ iff for any $\mathfrak{A}$, if $\mathfrak{A}\vDash P_1$, …, $\mathfrak{A}\vDash P_n$ then $\mathfrak{A}\vDash Q$. Thus, $<\{P,P\rightarrow Q\},Q>\in\vDash$ but $<\{Q,P\rightarrow Q\},P>\notin\vDash$.

Now Gentzen's goal is to provide a purely syntactic definition of a set that captures all and only the arguments in $\vDash$. Let us call this set that is defined syntactically by the name $\vdash$. To say that $\vdash$ is defined syntactically means that its defining conditions will mention only the physical shape of formulas. The semantic definition of $\vDash$ by contrast reaches well beyond syntax because it is defined in terms of domains, interpretations, reference, satisfaction and true. Gentzen's goal then is to define $\vdash$ in pure syntactic terms but in such a way that it happens to exactly the same arguments that are in $\vDash$. That is, when all is said and done, it is his goal to be able to show that $\vdash$ and $\vDash$ are coextension. This coincidence will be proven in what is called the ***soundness and completeness*** theorem:

For any $<\{P_1,...,P_n\}, Q>$, $<\{P_1,...,P_n\}, Q>\in\vdash$ iff $<\{P_1,...,P_n\}, Q>\in\vDash$.

(We will actually prove something stronger.) Since the axiom of extensionality in set theory says that two sets are identical iff they have the same elements, Gentzen's goal (and the soundness and completeness theorem) may be reformulated. It is to show that $\vdash=\vDash$.

But first we must define $\vdash$ and do so in a purely syntactic way. How is this to be done? By syntactic construction: $\vdash$ is defined inductively in a way that the concepts used in its defining clauses only appeal to syntactic ideas. The construction is fairly straightforward.

First in its basis clause the set of basic elements (the "starter set") is specified. We require some initial arguments that we know independently are valid but that we can specify in purely syntactic terms. We use a particularly trivial and obvious set of arguments: all those that "beg the question." Let us pick out the special case of circular arguments that repeat as its conclusion one of its premises. Let us call any deduction $<\{P_1,...,P_n\}, Q>$ such that for some *i,* $Q=P_i$ a ***basic deduction***. Now, circular arguments have been reviled since Aristotle and labeled as a fallacy (*petitio principii*), but their "error" is that they are trivial or uninformative, not that they are invalid. On the contrary, there couldn't be a more obviously valid argument: $<\{P,Q,R\},Q>$ is obviously valid because in any world *w* if *P,Q,* and *R* are true in *w*, then clearly *Q* is true in *w*. Moreover, you can tell that a basic deduction is valid by a finite physical inspection of symbols on the page: move from left to right checking each premise to see whether it is the same shape as the conclusion. (Note that by definition a basic deduction has only a finite number of premises.) Thus the "starter set" for $\vdash$ is the set of all basic deductions.

Next construction rules are defined that put new elements (deductions) in to $\vdash$ given that other elements (deductions) are in $\vdash$. These rules must be formulated solely in syntactic terms. They have the form:

If $<\{P_{1,1},...,\mathrm{P}_{m,1}\}, Q_1>$, …, $<\{P_{1,k},...,\mathrm{P}_{n,k}\}, Q_k>$ are all in $\vdash$ and meet syntactic condition *C,* then $<\{P_1,...,P_o\}, Q>$ which meets syntactic condition *D* is in $\vdash$.

But we do not want to put just any arguments into $\vdash$. We want to put in just valid arguments. So, the rule must be designed with this ulterior motive in mind. It should be formulated so that it introduces just valid arguments. Now if all the "starter set" arguments are valid, and if the construction rules introduce just valid arguments, it will follow that all the arguments in $\vdash$ are valid. So, though the construction rules must be formulated solely in syntactic terms, they should be written so as to identify new valid arguments to add to $\vdash$ from other valid arguments already in $\vdash$. If the construction rules really do produce valid arguments from valid arguments, the system is called ***sound***. Moreover, we want enough rules to insure that we manage to get all the valid arguments into $\vdash$. If they succeed in capturing all the valid arguments, the system is called ***complete.*** Coming up with a set of rules, formulated only in syntactic terms, that manages to capture in $\vdash$ all and only the valid arguments, and then proving that you have done so, it no small accomplishment.

Indeed, providing a twofold characterization of the same set, once as $\vDash$ and again as $\vdash$, provides an extremely satisfying explanation of two divergent traditions in logic. One is that logical truth and validity is a matter of form. Aristotle, for example, showed that all the valid syllogistic moods could be reduced by syntactic appeal to a small group of syntactic rules to the perfect syllogisms Barbara and Celarent, and all logic students from the in the Middle Ages through the 19th century learned to do these syntactic reductions in elementary logic courses. Kant, to cite another example, says a proposition is logically true if a contradiction can be formally deduce from its negation. Moreover, because they are syntactic, proofs are easy to check. Hence whether a formal deduction concludes with a contradiction as its last line is a fact that is "epistemically" obvious in some basic sense. Logical truths in this sense are thus "evident" as a matter of form. Many philosophers, like Kant, have even said they are known *a priori*, *i.e.* independently of sense experience. It is this formal notion of logic that is captured by $\vdash$.

On the other hand, logic has always been claimed to be about what is necessary, and necessity is in turn explained in terms of possibilities. The necessity of logical truths, says Aristotle, makes them true at all possible times in world history. Leibniz says they are true in all possible worlds. This is the semantic sense of logic captured by $\models$. The identity of the two notions shows that both are right, that there are two conceptually independent routes to the same important idea.

The successful definition of $\vdash$ and $\models$, and the subsequent proof that the two are identical satisfies most logicians. The combination of (1) the set theoretic definition of model and its standard definition of $\models$, (2) the syntactic construction of $\vdash$ using traditional rules of logic, and (3) proofs that $\models$ and $\vdash$ are coextensive is called ***classical logic***. Gentzen's methods are important because provides a way for classical logicians to show the soundness and completeness of their systems.

But Gentzen had a further philosophical objective. One reaction to the paradoxes of set theory is to say that the original formulations of set theory advanced by Cantor and Frege were naïve. The paradoxes show, in fact, that the subject matter is more complex than they thought. One reaction is to improve their axiom systems in ways that avoid contradictions. Researchers who have followed this course have developed the branch of logic called ***axiomatic set theory***.

Quite a different philosophical reaction, however, is to say that the paradoxes show that the entire notion of set is confused. Such logicians claim that set theory is suspicious in a number of ways. The principle of abstraction is overly generous in the number of sets that it asserts exists. Not just any open formula determines a set. There must be some further restriction placed on the sets. One proposed condition is that the sets be "constructed." Another dubious feature of Cantor's theory is the use of reduction to the absurd and excluded middle in its proofs. (If you look back at the relevant proofs in Chapter 1, you will see these being used.) Thus there are schools called ***intuitionistic logic*** and ***constructivist mathematics*** that insist that sets should be introduced only by construction and that logic should eschew certain classical rules like *reductio* and excluded middle. Gentzen sympathized with the intuitionistic doubts.

But semantics as we have developed it thus far as made extensive use of sets. Domains are sets. The interpretation function and satisfaction relation are sets of *n*-types. Model theory is highly set theoretic. Formal semantics, then, is highly dubious according to intuitionistic logicians. But if we cannot make use of sets, how can we do semantics? How can we have a theory of meaning? What intellectual resources would be safe and legitimate if you were an intuitionist? Maybe these "safe concepts" – whatever they would be – could be used to formulate a theory of meaning.

One branch of logic that seems to be intutionistically safe is syntax. Its sets are either finite or countably infinite (at least in classical logic), and when infinite they are constructed by syntactic epistemically transparent methods. It is hard to see how a contradiction could sneak in. Perhaps there could be a purely syntactic account of meaning. If so, then this could be embraced by intuitionists and constructivists because sets in syntax are all constructive.

Now, one sense of *meaning* is *use*. If we knew how to *use* the connectives of sentential logic and the quantifiers of first-order logic, we could be said to understand their *meaning*. (As philosophy students know, Ludwig Wittgenstein devoted the latter part of his career to developing a very powerful philosophy of language that explicates the *meaning as use* thesis.) But how do logicians *use* the logical signs? In proofs. If we knew how we in fact use logic signs in proofs, we would know their meanings. But how do we use them? Gentzen's observation is that we *introduce* them into new lines of a proof, and we *eliminate* them from lines already there. That's all we do with them. If we could codify in purely syntactic terms how we do this, if we could codify the introduction and elimination rules for each logical sign, we would have a theory of meaning for logic in the meaning as use sense. Gentzen proposed such a rule set. He has one introduction and one elimination rule for each logical sign. (As we shall see, to get just one introduction and one elimination rule for each, it is necessary for him to adopt the fiction that the contradiction sign $\perp$ is a "zero-place connective.") Indeed, because the Gentzen' rules characterizing $\vdash$ are more like ones logicians actually use in practice, much more so that the convoluted and obscure proofs necessary in axiom system, he called his system "natural" deduction.

To summarize, then, Gentzen proposed a construction rule set for $\vdash$ which is such that (1) it is co-extensive with $\models$ as defined in standard model theoretic semantics and (2) provides an introduction and elimination syntactic "meaning as use" rule set.

Now it is a bit inconsistent to say, on the one hand, that sets make no sense and therefore we should jettison semantic model theory with its definition of $\models$ in terms of models, and, on the other hand, insist that when we define syntactic rules constructing $\vdash$ they should be chosen so that $\vdash$ is identical to $\models$. Why bother with making $\vdash$ identical with $\models$, if $\models$ doesn't make any sense? A good question.

On one level Gentzen's rule set can be viewed as a pure effort in classical logic to provide a constructive account of $\vdash$ that is provably coextensive with $\models$. When the theory is presented in this way, the rules used to construct $\vdash$ are classical. They include classical rules like *reductio* and excluded middle that strict intuitionists question. On another level, set theoretic semantics can be rejected and the introduction and elimination rules read as an account of logical meaning. When this is done the construction rule set is modified so that it does not include reductio or excluded middle. Let us use $\vdash_{Int}$ to name the set of deduction produced by the intuitionistic rules set. In summary, then, it is possible to characterize by syntactic different sets of syntactic introduction rules two sets: $\vdash$ (of classical logic) and $\vdash_{Int}$ capturing the deductions intuitionists accept. $\vdash$ exactly coextensive with $\models$ as defined in classical model theoretic semantics. Even without a model theoretic semantics $\vdash_{Int}$ "explains" the meaning of logical signs because in its introduction and elimination rules it provides an account of their use, and does so without appeal to controversial classical rules like *reductio* and excluded middle.

Curiously, despite their doubts about sets intuitionistic logicians have nevertheless advanced modified definitions of model, defined for them a validity relation $\models_{\text{Int}}$, and shown it to be coextensive with their modified $\vdash_{\text{Int}}$. These results are both technically interesting and conceptual puzzling, but they are unfortunately a topic we cannot pursue in this book. (See for example, Michael Dummett, *Elements of Intuitionism.)*

iii. Natural Deduction for Sentential Logic

The rules Gentzen uses to construct the set $\vdash_{\mathbf{SL}}$, which we shall call the set of acceptable deductions, are familiar from elementary logic. What is novel is their theoretical use, and the notation used to formulate them. Instead of set theoretic formulations, it is customary to write $X\vdash P$_ to mean $<X,P>\in\vdash$, and to state the construction rule

If $<X_1,P_1>\in\vdash$ and … and $<X_n,P_n>\in\vdash$, then $<Y,Q>\in\vdash$

in "tree" notation:

$$\frac{X_1\vdash P_1, \ldots X_n\vdash P_n}{Y\vdash Q}$$

The notation allows for an elegant statement of the definition of $\vdash_{\mathbf{SL}}$ from the perspective of graphic design, but the reader should keep the real set theoretic meaning in mind. We are now ready to state the natural deduction syntactic definition for $\vdash_{\mathbf{SL}}$ in classical logic.

A (Classical) Natural Deduction Systems for Sentential Logic

A ***deduction*** is any pair $<X,P>$ such that X is a finite set of formulas and P a formula in a sentential language **SL**.

1. The Basic Elements. The set **BD** of ***basic deductions*** is the set of all deduction $<X,P>$ such that $P\in X$.

2. The Construction Rules. We adopt the following abbreviations:

$X\vdash_{\text{SL-ND}}P$	for	$<X,P>$ is in $\vdash_{\text{SL-ND}}$;
$X,Y\vdash_{\text{SL-ND}}P$	for	$X\cup Y\vdash_{\text{SL-ND}}P$;
$X,P\vdash_{\text{SL-ND}}Q$	for	$X\cup\{P\}\vdash_{\text{SL-ND}}Q$;
$P_1,\dots,P_n\vdash_{\text{SL-ND}}Q$	for	$\{P_1,\dots,P_n\}\vdash_{\text{SL-ND}}Q$;
$\vdash_{\text{SL-ND}}P$	for	$\varnothing\vdash_{\text{SL-ND}}P$.

The set $\mathbf{R}_{\textbf{ND-SL}}$ of ***Natural Deduction*** (construction) ***Rules for Sentence Logic***:

	Introduction (+) Rules	Elimination (-) Rules
$\bot$	$\dfrac{X\vdash_{\text{SL-ND}}P\ \ Y\vdash_{\text{SL-ND}}{\sim}P}{X,Y\vdash_{\text{SL-ND}}\bot}$ [44]	$\dfrac{X\vdash_{\text{S-ND}}\bot}{X\text{-}\{{\sim}P\}\vdash_{\text{SL-ND}}P}$
$\sim$	$\dfrac{X\vdash_{\text{SL-ND}}\bot}{X-\{P\}\vdash_{\text{SL-ND}}{\sim}P}$	Classical Rule $\dfrac{X\vdash_{\text{SL-ND}}{\sim}{\sim}P}{X\vdash_{\text{SL-ND}}P}$ (Intuitionistic Rule $\dfrac{X\vdash_{\text{SL}}P\ \ Y\vdash_{\text{INT-ND}}{\sim}P}{X,Y\vdash_{\text{INT-ND}}Q}$
$\wedge$	$\dfrac{X\vdash_{\text{SL-ND}}P\ \ Y\vdash_{\text{SL-ND}}Q}{X,Y\vdash_{\text{SL-ND}}P\wedge Q}$	$\dfrac{X\vdash_{\text{SL-ND}}P\wedge Q}{X\vdash_{\text{SL-ND}}P}$ $\dfrac{X\vdash_{\text{SL-ND}}P\wedge Q}{X\vdash_{\text{SL-ND}}Q}$
$\vee$	$\dfrac{X\vdash_{\text{SL-ND}}P}{X\vdash_{\text{SL-ND}}P\vee Q}$ $\dfrac{X\vdash_{\text{SL-ND}}Q}{X\vdash_{\text{SL-ND}}P\vee Q}$	$\dfrac{X\vdash_{\text{SL-ND}}P\vee Q\quad Y\vdash_{\text{SL-ND}}R\quad Z\vdash_{\text{SL-ND}}R}{X,Y-\{P\},Z-\{Q\}\vdash_{\text{SL-ND}}R}$
$\rightarrow$	$\dfrac{X\vdash_{\text{SL-ND}}P}{X-\{Q\}\vdash_{\text{SL-ND}}Q\rightarrow P}$	$\dfrac{X\vdash_{\text{SL-ND}}P\quad Y\vdash_{\text{SL-ND}}P\rightarrow Q}{X,Y\vdash_{\text{SL-ND}}Q}$

Thinning $\dfrac{X\vdash_{\text{SL-ND}}P}{X,Y\vdash_{\text{SL-ND}}P}$

3. The Set of (Acceptable) SL Deductions. The relation $\vdash_{\text{SL-ND}}$ is defined inductively:

i. **BD** is a subset of $\vdash_{\text{SL-ND}}$;

ii. If $\mathbf{d}_1,\dots,\mathbf{d}_m$ are in $\vdash_{\text{SL-ND}}$ and $\mathbf{d}_n$ follows from $\mathbf{d}_1,\dots,\mathbf{d}_m$ by one of the above (classical) rules, then $\mathbf{d}_n$ is in $\vdash_{\text{SL-ND}}$;

iii. nothing else is in $\vdash_{\text{SL-ND}}$.

We extend the notion of deduction to possibly infinite sets of premises X by saying $X\vdash_{\text{SL-ND}}Q$ relative to $\vdash_{\text{SL}}$ iff, there is some finite subset $\{P_1,\dots,P_n\}$ of X such that $P_1,\dots,P_n\vdash_{\text{SL-ND}}Q$.

[44] Strictly speaking $\bot$+ is not a true introduction rule because it mentions another connective in its input argument (*viz.* ~). Note, however, that $\bot$+ is directly provable using $\rightarrow$– if ${\sim}P$ is defined as $P\rightarrow\bot$. Depending on which connectives are primitive (e.g. $\bot$ or ~ combined with $\wedge,\vee$, or $\rightarrow$) various of the rules will be come redundant and provable. Here we opt for the full set.

It is helpful to associate with the Gentzen rules their traditional names: $\bot$- is *ex falso quodlibet* (from a falsehood anything follows), $\sim$+ is *reductio ad absurdum*, $\rightarrow$+ is *conditional proof*, and $\rightarrow$ - is *modus ponens* $\sim$- is half of *double negation*, $\wedge$+ is *conjunction*, $\wedge$- is *simplification*, $\vee$+ is *addition*, and $\vee$- is *constructive dilemma*.

The definition of $\vdash_{\mathbf{FOL}}$ requires introduction and elimination rules for the universal quantifier. Below we also include rules for the existential quantifier though they are redundant.

A (Classical) Natural Deduction Systems for First-Order Logic

A ***deduction*** is any pair $<X,P>$ such that X is a finite set and P a sentence in an **FOL-ND** language.

1. The Basic Elements. The set **BD** of ***basic deductions*** is the set of all deduction $<X,P>$ such that $P\in X$.

2. The Construction Rules. We adopt the following abbreviations:

$X\vdash_{\text{FOL-ND}}P$	for	$<X,P>$ is in $\vdash_{\text{FOL-ND}}$;
$X,Y\vdash_{\text{FOL-ND}}P$	for	$X\cup Y\vdash_{\text{FOL-ND}}P$;
$X,P\vdash_{\text{FOL-ND}}Q$	for	$X\cup\{P\}\vdash_{\text{FOL-ND}}Q$;
$P_1,\ldots,P_n\vdash_{\text{FOL-ND}}Q$	for	$\{P_1,\ldots,P_n\}\vdash_{\text{FOL-ND}}Q$;
$\vdash_{\text{FOL-ND}}P$	for	$\varnothing\vdash_{\text{FOL-ND}}P$.

The set $\mathbf{R}_{\textbf{FOL-ND}}$ of ***Natural Deduction*** (construction) ***Rules for First-Order Logic***: the set $\mathbf{R}_{\textbf{ND-SL}}$ plus the four rules:

Introduction (+) Rules **Elimination (-) Rules**

$\forall$ $$\frac{X\vdash_{\text{FOL-ND}}P[v'/v]}{X\vdash_{\text{FOL-ND}}\forall vP} \qquad \frac{X\vdash_{\text{FOL-ND}}\forall vP[}{X\vdash_{\text{FOL-ND}}P[t/v]}$$

where v' is not free in any $P\in X$

In addition we add two redundant rules for the existential quantifier:

$\exists$ $$\frac{X\vdash_{\text{FOL-ND}}P[t//v]}{X\vdash_{\text{FOL-ND}}\exists v'P} \qquad \frac{X\vdash_{\text{FOL-ND}}\exists vP \qquad Y,P[v'/v]\vdash_{\text{FOL-ND}}Q}{X,Y\vdash_{\text{FOL-ND}}Q}$$

(if v' is not free in $X,Y,\exists vP$ or Q)

3. The Set of (Acceptable) FOL-ND Deductions. The relation $\vdash_{\text{FOL-ND}}$ is defined inductively:

i. **BD** is a subset of $\vdash_{\text{FOL-ND}}$;

ii. If $\mathbf{d}_1,\ldots,\mathbf{d}_m$ are in $\vdash_{\text{FOL-ND}}$ and $\mathbf{d}_n$ follows from $\mathbf{d}_1,\ldots,\mathbf{d}_m$ by one of the above rules, then $\mathbf{d}_n$ is in $\vdash_{\text{FOL-ND}}$;

iii. nothing else is in $\vdash_{\text{FOL-ND}}$.

We extend the notion of deduction to possibly infinite sets of premises X by saying $X\vdash_{\text{FOL-ND}}Q$ relative to $\vdash_{\text{FOL-ND}}$ iff, there is some finite subset $\{P_1,\ldots,P_n\}$ of X such that $P_1,\ldots,P_n\vdash_{\text{FOL-ND}}Q$.

Additional natural deduction rules are needed for proofs using identity in **FOL=.** Only two rules are needed, a version of the axiom of self-identity and one for the substitutivity of identity.

Additional Rules for FOL=-ND

Introduction Rule for Identity **Elimination Rule for Identity**

$$\frac{X\vdash P}{X\vdash t=t} \qquad\qquad \frac{X\vdash P \quad Y\vdash t=t'}{X,Y\vdash P[t'//t]}$$

Metatheorem 1-4. **The following deduction is provable: $\varnothing\vdash t=t$**

Examples of Natural Deduction Proofs in Sentential Logic. These metatheorems each assert that a given deduction is provable in the natural deduction system (*i.e.* that the deduction is in the constructive set $\vdash_{\text{SL-ND}}$ of "provable deductions.") The assertion (at the "root" of the tree) is proven by actually presenting the proof tree which describes a construction that shows step by step how the deduction is arrived at from the basic deductions that occupy the "leaves" of the tree).

Metatheorem 1-5. ***P*$\vdash$~~*P*** (That is, the deduction <*P,~~P*> is in the set of pairs $\vdash_{\text{SL-ND}}$.)

$$\dfrac{\dfrac{P\vdash P \quad {\sim}P\vdash{\sim}P}{P,\,{\sim}P\vdash\bot}\,{\sim}\bot+}{P\vdash{\sim}{\sim}P}\,{\sim}+$$

Metatheorem 1-6. ***P,~Q*$\vdash$~(*P*→*Q*)** (That is, the deduction <{*P,~Q*},~(*P*→*Q*)> is in the set of pairs $\vdash_{\text{SL-ND}}$.)

$$\dfrac{\dfrac{\dfrac{P,\,P\to Q\vdash P \quad P,\,P\to Q\vdash P\to Q}{P,\,P\to Q\vdash Q}\to- \qquad {\sim}Q\vdash{\sim}Q}{P,\,P\to Q,\,{\sim}Q\vdash\bot}\,\bot+}{P,{\sim}Q\vdash{\sim}(P\to Q)}\,{\sim}+$$

Metatheorem 1-7. **~*Q*$\vdash$~(*P*∧*Q*)**

$$\dfrac{\dfrac{P\wedge Q\vdash P\wedge Q}{P\wedge Q\vdash Q}\wedge- \qquad {\sim}Q\vdash{\sim}Q}{Q\vdash{\sim}(P\wedge Q)}\,{\sim}+$$

An Intuitionistic Natural Deduction Systems for the Valid Arguments of First-Order Logic

Intuitionistic logic has the same rule as classical logic except for ⊥- and ~-. The intuitionistic versions, which are weaker than their classical counterparts, are:

$$\dfrac{X\vdash_{\text{INT-ND}}\bot}{X\vdash_{\text{INT-ND}}P}\ \bot\text{-I} \qquad\qquad \dfrac{X\vdash_{\text{INT-ND}}P \quad Y\vdash_{\text{INT-ND}}{\sim}P}{X,Y\vdash_{\text{INT-ND}}Q}\ {\sim}\text{-I}$$

Let us call the set constructed from basic deductions by this rule set $\vdash_{\text{Int-SL-ND}}$

Examples of Proving Metatheorems by Constructing Intuitionistic Proof Trees

Metatheorem 1-8. **Intuitionistic ⊥- is a derivable rule using Classical ~ rules:**

Proof:

$$\dfrac{\dfrac{\dfrac{X\vdash\bot}{X\text{-}\{{\sim}P\}\vdash{\sim}{\sim}A}\ {\sim}+}{X\text{-}\{{\sim}P\}\vdash A}\ {\sim}-}{X\vdash A}\ \text{Th}$$

(This tree shows a general method for adding <*X, A*> to $\vdash_{\text{Int-SL-ND}}$ if <*X*,⊥> is already a member.)

Metatheorem 1-9. **Intuitionistic ~- is a derivable rule using Classical ~-:**

Proof:

$$\dfrac{\dfrac{\dfrac{X\vdash P \qquad \dfrac{Y,\vdash{\sim}P}{Y,{\sim}Q\vdash{\sim}P}\ \text{Th}}{X,Y,{\sim}Q\ \vdash\bot}\ {\sim}+}{X,Y\vdash{\sim}{\sim}Q}\ \bot+}{X,Y\vdash Q}\ {\sim}-$$

Metatheorem 1-10. ***~P,~Q*⊢*~*(*P*∨*Q*) in Intuitionistic Logic.** (That is, the deduction <{~*P*,~*Q*>},~(*P*∨*Q*)> is in the set of pairs $\vdash_{\text{Int-SL-ND}}$.)

$$\dfrac{\dfrac{P\vee Q\vdash P\vee Q \quad \dfrac{P\vdash P \quad {\sim}P\vdash{\sim}P}{P,{\sim}P\vdash R} \quad \dfrac{Q\vdash Q \quad {\sim}Q\vdash{\sim}Q}{Q,{\sim}Q\vdash R}\ {\sim}\text{-I}}{P\vee Q,{\sim}P{\sim}Q\vdash R}\ \vee- \qquad \dfrac{P\vee Q\vdash P\vee Q \quad \dfrac{P\vdash P \quad {\sim}P\vdash{\sim}P}{P,{\sim}P\vdash{\sim}R} \quad \dfrac{Q\vdash Q \quad {\sim}Q\vdash{\sim}Q}{Q,{\sim}Q\vdash{\sim}R}\ {\sim}\text{-I}}{P\vee Q,{\sim}P{\sim}Q\vdash{\sim}R}\ \vee-}{{\sim}P,{\sim}Q\vdash{\sim}(P\vee Q)}\ {\sim}+$$

Metatheorem 1-11. In Classical Logic, ⊢***P*∨~*P*** (That is, the deduction <∅,*P*∨~*P*> is in the set of pairs $\vdash_{\text{SL-ND}}$.)

$$\dfrac{\dfrac{{\sim}(P\vee P)\vdash{\sim}(P\vee P) \qquad \dfrac{P\vdash P}{P\vdash P\vee{\sim}P}\ \vee+}{{\sim}(P\vee \mathrm{P})\vdash\bot}\ \bot+}{\vdash P\vee{\sim}P}\ \bot\text{- (the classical rule)}$$

Metatheorem 1-12. In intuitionistic logic, $\vdash\boldsymbol{\sim\sim(P\vee\sim P)}$ **.** $\vdash\boldsymbol{P\vee\sim P}$ (That is, the deduction $<\varnothing,\sim\sim()P\vee\sim P>$ is in the set of pairs $\vdash_{\text{Int-SL-ND}}$ **.**)

$$\dfrac{\dfrac{\sim(P\vee P)\vdash\sim(P\vee P) \qquad \dfrac{P\vdash P}{P\vdash P\vee\sim P}\,\vee+}{\sim(P\vee QP\vdash\bot}\,\bot+}{\vdash\sim\sim(P\vee\sim P)}\,\sim+$$

II. Completeness of First-Order Logic

A. Introduction

i. Strategy

In this section we state and prove the so-called soundness and completeness theorem for first-order logic. The result is fundamental to the entire enterprise of constructing a proof system. It shows that the set of natural deduction rule are successful in their goal. They "capture" all the valid inferences of the language and leave none out. In the "soundness theorem," it is shown that every argument provable by the rules is in fact valid. In the "completeness theorem" the converse is proven – that every valid argument is provable by the rules. (Note that in less precise contexts "soundness and completeness theorem is often shortened to just "completeness".)

Going through the proof in detail is a standard part of logic course introducing the subject to students in specialized fields like philosophy, mathematics, and computer science. The presentation here is designed to facilitate its role as an introduction. The semantics stated earlier in the Chapter follows the original form of Alfred Tarski in its notation and in the two stage development, in which "satisfaction in a model relative to a variable assignment" is defined first and then in terms of it the notion of "truth in a model." The syntax also follows Tarski in allowing for vacuous quantifiers and for well-formed formulas containing free-variables which are viewed as logically equivalent to their universal closure. The completeness proof in this section employs a method that is very elegant and regarded as the standard. It is due to Leon Henkin who developed it originally for axiom systems. It is adapted here to the natural deduction system stated earlier, which is itself a standard version of a natural deduction system for first-order logic based on the ideas of Gerhard Gentzen and Dag Prawitz. Though there are more streamline versions of the syntax and semantics, and other ways to prove completeness, the ideas used here are what most logicians have in the back of their minds as the standard account. It this that students are assumed to be familiar with, and it is to this that the proof theory and semantics of more innovative logics are compared. The various steps in the completeness proof are spelled out in detail, including the necessary background metatheorems. In reading the proofs, there are several things the reader should keep on the lookout for – when you find them, make a note of it:

- the logical manipulation of quantifiers and connectives in the metalanguage to prove facts about the connectives and quantifiers in the object language,
- in proofs by induction how the inductive hypothesis is spelled out and used,
- cases in which specific clauses in the definition of "satisfaction relative to a variable assignment" are used to restate facts about the truth-value of a complex formula in terms of more basic facts about the interpretation of its atomic parts,
- steps in proofs in which logical ideas like validity and satisfiability, proof theoretic ideas like consistency and proof, and syntactic ideas like substitution and alphabetic variance are unpacked by definition.

ii. Background Metatheorems

We begin by stating background metatheorems needed later in the Henkin completeness proof. The first metatheorem established how free variables should be read in the standard Tarski semantics. They are in a sense two-faced. Relative to a model and an interpretation of its variables, the variables function like pronouns. They have a single referent much like proper names, and they are used to state facts about individuals that are not generally true of everything in the domain. When we have abstracted from the variable assignments however, and are evaluating a formula relative to the model as a whole, open sentences are logical equivalent to their universal closures.

Metatheorem 1-13. T**he following are equivalent and mutually interdeducible:**

1. $\mathfrak{A}\models P$
2. for all ***s***, $\mathfrak{A}_{\boldsymbol{s}}\models P$
3. for all ***s***, $\mathfrak{A}_{\boldsymbol{s}}\models\forall vP$

Proof. We show that each entails its successor.

1. That 1 entails 2 follows by the definition of "truth *simpliciter* in a model."

2. Assume for all ***s***, $\mathfrak{A}_{\boldsymbol{s}}\models P$. Consider now an arbitrary assignment ***s′*.** Now consider any *v*-variant ***s″*** of ***s′*.** By universal instantiation, then $\mathfrak{A}_{\boldsymbol{s''}}\models P$. But then $\mathfrak{A}_{\boldsymbol{s'}}\models\forall vP$. Since ***s′*** is typical of all assignments, we may generalize, for all ***s′***, $\mathfrak{A}_{\boldsymbol{s'}}\models\forall vP$.

3. Assume that for all ***s***, $\mathfrak{A}_{\boldsymbol{s}}\models\forall vP$. Then for all ***s*** and all *v*-variant ***s′*** of ***s***, $\mathfrak{A}_{\boldsymbol{s'}}\models P$. Consider now an arbitrary assignment ***s″***. Then by universal instantiation, for all *v*-variant ***s‴*** of ***s″***, $\mathfrak{A}_{\boldsymbol{s'''}}\models P$. But ***s″*** is a *v*-variant of itself. Therefore, again by universal instantiation, $\mathfrak{A}_{\boldsymbol{s''}}\models P$.
But since ***s″*** is arbitrary, we may generalize for all ***s***, $\mathfrak{A}_{\boldsymbol{s}}\models P$, i.e. $\mathfrak{A}\models P$.

In some sense it is obvious that it does not matter what variables you use to state a fact, and that it is possible to state the same fact using different variables. This truth is captured by means of the concept of alphabetic variance. A formula and its alphabetic variant are logically equivalent.

Metatheorem 1-14 (Alphabetic Variance). **If P_σ is an alphabetic variant of *P* relative to some substitution function σ for all terms, then for all $\mathfrak{A}$, $\mathfrak{A}\models P$ iff $\mathfrak{A}\models P_\sigma$.**

Proof. The proof is by induction. The proposition to be proven is really a universal quantification over formulas in the metalanguage:

For any formula *P* in **For$_{\mathbf{FOL}}$**, if P_σ is an alphabetic variant of *P* relative to some full
substitution function σ, then for all $\mathfrak{A}$, $\mathfrak{A}\models P$ iff $\mathfrak{A}\models P_\sigma$.

The "property" that must be shown to hold for all formulas in **For$_{\mathbf{FOL}}$** is sated in the open sentence:

if $\ldots_\sigma$ is an alphabetic variant of … relative to some full
substitution function σ, then for all $\mathfrak{A}$, $\mathfrak{A}\models\ldots$ iff $\mathfrak{A}\models\ldots_\sigma$.

In the proofs basic step we show that the property holds if the blank … is filled by an atomic formula $P^nt_1\ldots t_n$. In the inductive step we much consider each formation rule for formulas. For each rule, we assume in the induction hypothesis that the property holds of the immediate parts of the formula, and then show it holds for the formula itself.

Basic Step. The atomic case follows directly from the previous metatheorem. For let $(P^nt_1\ldots t_n)_\sigma$= $(P^nt_1...t_n)_\sigma = P^n\sigma(t_1)$.:. $\sigma(t_n)$ be an alphabetic variant of $P^nt_1\ldots t_n$ relative to a fill substitution function σ.

Let $\mathfrak{A}$ be arbitrary and assume $\mathfrak{A}\models P^nt_1\ldots t_n$
Let $v_1\ldots v_m$ be the variables from among $t_1\ldots t_n$ and let ***s*** be arbitrary.
By the previous metatheorem, for any $\boldsymbol{s}'$ if $\boldsymbol{s}'$ is a variant of ***s*** with respect to $v_1\ldots v_m$
then $\mathfrak{A}_{\boldsymbol{s}'}\models P^nt_1\ldots t_n$
Define $\boldsymbol{s}''$ such that $\boldsymbol{s}''(v_i)=\boldsymbol{s}(\sigma(v_i))$ for i =1,…,m.
Therefore, by universal instantiation, $\mathfrak{A}_{\boldsymbol{s}''}\models P^nt_1\ldots t_n$
That is, $<\boldsymbol{s}''(t_1),\ldots,\boldsymbol{s}''(t_n)>\in\Im(P^n)$
Hence by substitutivity of identity, $<\boldsymbol{s}(\sigma(t_1)),\ldots,\boldsymbol{s}(\sigma(t_n))>\in\Im(P^n)$
Thus, $\mathfrak{A}_{\boldsymbol{s}}\models P^n\sigma(t_1)\ldots\sigma(t_n)$
But since ***s*** is arbitrary we may generalize, for all ***s*,** $\mathfrak{A}_{\boldsymbol{s}}\models P^n\sigma(t_1)\ldots\sigma(t_n)$
That is, $\mathfrak{A}\models P^n\sigma(t_1)\ldots\sigma(t_n)$
Hence if $\mathfrak{A}\models P^nt_1\ldots t_n$, then $\mathfrak{A}\models P^n\sigma(t_1)\ldots\sigma(t_n)$
The converse is proven similarly.
Hence, $\mathfrak{A}\models P^nt_1\ldots t_n$ iff $\mathfrak{A}\models P^n\sigma(t_1)\ldots\sigma(t_n)$
Since $\mathfrak{A}$ is arbitrary we may universally generalize, for all $\mathfrak{A}$, $\mathfrak{A}\models P^nt_1\ldots t_n$ iff $\mathfrak{A}\models P^n\sigma(t_1)\ldots\sigma(t_n)$

Inductive Step

Connectives. The cases for the connectives follow immediately from the inductive hypothesis and the fact that if the immediate parts of a formula formed by a connective are logically equivalent, so is the whole. (The details are left to the reader as an exercise.)

Universal Quantifier. In the case of the universal quantifier, we assume as the inductive hypothesis:

for all $\mathfrak{A}$, $\mathfrak{A}\models P$ iff $\mathfrak{A}\models P_\sigma$.

We then show that:

for all $\mathfrak{A}$, $\mathfrak{A}\models\forall vP$ iff $\mathfrak{A}\models(\forall vP)_\sigma$.

Let $\mathfrak{A}$ be arbitrary. We note that each of the following lines is equivalent to the line beneath:

$\mathfrak{A}\models\forall vP$	Assumption
for all ***s***, $\mathfrak{A}_{\boldsymbol{s}}\models\forall vP$	(by definition)
for all ***s*** and all *v*-variant $\boldsymbol{s}'$ of ***s***, $\mathfrak{A}_{\boldsymbol{s}'}\models P$	(by definition)
for all ***s*** and all *v*-variants $\boldsymbol{s}'$ of ***s***, $\mathfrak{A}_{\boldsymbol{s}'}\models(P_\sigma)$	(by the induction hypo.)
for all ***s*** and all $\sigma(t)$-variants $\boldsymbol{s}'$ of ***s***, $\mathfrak{A}_{\boldsymbol{s}'}\models(P_\sigma)$	(by sub. of =)
for all ***s***, $\mathfrak{A}_{\boldsymbol{s}}\models\forall\sigma(t)(P_\sigma)$	(by definition)
$\mathfrak{A}\models\forall\sigma(t)(P_\sigma)$	(by definition)
$\mathfrak{A}\models(\forall vP)_\sigma$	(by definition)

Since these are equivalents: $\mathfrak{A}\models\forall vP$ iff $\mathfrak{A}\models(\forall vP)_\sigma$. Since is arbitrary we may universally generalize: for all $\mathfrak{A}$, $\mathfrak{A}\models\forall v\,P$ iff $\mathfrak{A}\models(\forall vP)_\sigma$. **QED.**

The following metatheorem assures us that adding new constants to the syntax does not alone alter the truth-value of any sentence written in the syntax as it was before the introduction of the new names. What was true of everything and something remains the same, and what was true of the individuals we could name remains the same.

Definition. $\mathbf{F}_{\mathbf{FOL}}$ is a ***sublanguage*** of $\mathbf{F'}_{\mathbf{FOL}}$ iff ,$\mathbf{Trms}_{\mathbf{FOL}}\subseteq$ $\mathbf{Trms'}_{\mathbf{FOL}}$,and $\mathbf{Preds}_{\mathbf{FOL}}\subseteq$ $\mathbf{Preds'}_{\mathbf{FOL}}$.

Metatheorem 1-15

If 1. $\mathbf{F}_{\mathbf{FOL}}$ is a sublanguage of $\mathbf{F'}_{\mathbf{FOL}}$,
2. $\mathfrak{A}$=<D,$\Im$> is a model for $\mathbf{F}_{\mathbf{FOL}}$ and
3. $\mathfrak{A}'$=<D,$\Im'$> a model for $\mathbf{F'}_{\mathbf{FOL}}$, $\Im\subseteq\Im'$

then, any P of $\mathbf{F}_{\mathbf{FOL}}$, if ***s*** is a variable assignment for $\mathfrak{A}$, ***s′*** is a variable assignment for $\mathfrak{A}'_{\mathbf{L}}$, and ***s***$\subseteq$***s′***, then $\Im^{\mathfrak{A}}_{s}(P)=\Im'_{\boldsymbol{s}'}(P)$.

Assume the antecedents 1-3 of the theorem.

We must first show a lemma:

If the antecedents 1-3 above hold, then for any *t,* if $t\in\mathbf{Trms}_{\mathbf{FOL}}$, if ***s*** is a variable assignment for $\mathfrak{A}$, ***s′*** is a variable assignment for $\mathfrak{A}'_{\mathbf{L}}$, and ***s***$\subseteq$***s′***, $\Im^{\mathfrak{A}}_{s}(t)$= $\Im^{\mathfrak{A}'}_{s'}(t)$.

We assume the antecedents 1-3 of the theorem, and then show the consequent by induction because the set $\mathbf{Trms}_{\mathbf{FOL}}$ is inductively defined:

Atomic Case. Let *t* be a constant or variable in $\mathbf{Trms}_{\mathbf{FOL.}}$ Assume ***s*** is a variable assignment for $\mathfrak{A}$, ***s′*** is a variable assignment for $\mathfrak{A}'_{\mathbf{L}}$, and ***s***$\subseteq$***s′***. If *t* is a constant in $\mathbf{F}_{\mathbf{FOL}}$, $\Im$ is defined for *t.* Moreover, since $\Im\subseteq\Im'$, $\Im(t)=\Im'(t)$ and hence for any ***s*** and **s′**, $\Im^{\mathfrak{A}}_{s}(t)=\Im^{\mathfrak{A}'}_{s'}(t)$. If *t* is a constant in $\mathbf{F}_{\mathbf{FOL}}$, ***s*** is defined for *t.* Moreover, since ***s***$\subseteq$***s′***, ***s***(*t*)=**s′**(*t*), and hence for any $\Im$ and $\Im'$, $\Im^{\mathfrak{A}}_{s}(t)$= $\Im^{\mathfrak{A}'}_{s'}(t)$.

Molecular Case. Assume as the induction hypothesis that:

if ***s*** is a variable assignment for $\mathfrak{A}$, ***s′*** is a variable assignment for $\mathfrak{A}'_{\mathbf{L}}$, and ***s***$\subseteq$***s′***,
then $\Im^{\mathfrak{A}}_{s}(t_i)$= $\Im^{\mathfrak{A}'}_{s'}(t_i)$ for *i* =1,…,*n.*

We show:

if ***s*** is a variable assignment for $\mathfrak{A}$, ***s′*** is a variable assignment for $\mathfrak{A}'_{\mathbf{L}}$, and ***s***$\subseteq$***s′***,
then $\Im^{\mathfrak{A}}_{s}(f(t_1\ldots\ t_n))$= $\Im^{\mathfrak{A}'}_{s'}(f(\ t_1\ldots\ t_n))$.

Assume ***s*** is a variable assignment for $\mathfrak{A}$, ***s′*** is a variable assignment for $\mathfrak{A}'_{\mathbf{L}}$, and ***s***$\subseteq$***s′***. Now, $\Im^{\mathfrak{A}}_{s}$ $(f(t_1\ldots\ t_n))=\Im(f)(\Im^{\mathfrak{A}}_{s}(t_1),\ldots,\ \Im^{\mathfrak{A}}_{s}(t_1))$ [by the definition of $\Im^{\mathfrak{A}}_{s}$]=$\Im'(f)(\ \Im^{\mathfrak{A}'}_{s'}(t_1),\ldots,\ \Im^{\mathfrak{A}'}_{s'}(t_1))$ [by the induction hypothesis and the substitutivity of identity] = $\Im^{\mathfrak{A}'}_{s'}(f(t_1\ldots\ t_n))$.

We now prove the main theorem. We assume the antecedents 1-3 of the theorem, and then since the set $\mathbf{F}_{\mathbf{FOL}}$ is inductively defined, we show the consequent by induction:

Atomic Case. The result follows directly from the definitions of $\Im^{\mathfrak{A}}_{s}$, $\Im^{\mathfrak{A}}_{s'}$, lemma, and the substitutivity of identity. $\Im^{\mathfrak{A}}_{s}(P^n t_1\ldots\ t_n)$=T iff < $\Im^{\mathfrak{A}}_{s}(t_1),\ldots,\ \Im^{\mathfrak{A}}_{s}(t_n)$>$\in\Im(P^n)$ [by the definition of $\Im^{\mathfrak{A}}_{s}$] iff <$\Im^{\mathfrak{A}'}_{s}(t_1),\ldots,\ \Im^{\mathfrak{A}'}_{s}(t_n)$>$\in\Im(P^n)$ [by the lemma and the substitutivity of identity] iff <$\Im^{\mathfrak{A}'}_{s}(t_1),\ldots,\Im^{\mathfrak{A}'}_{s'}(t_n)$>$\in\Im'(P^n)$ [since $\Im$ is defined for P^n and $\Im\subseteq\Im'$] iff $\Im^{\mathfrak{A}'}_{s'}(P^n t_1\ldots\ t_n\ )$=T [by the definition of $\Im'_{\boldsymbol{s}}$].

Molecular Cases

<u>The Connectives.</u> The cases for the truth-functional connectives follow directly from the definition of $\Im^{\mathfrak{A}}_{s}$, $\Im^{\mathfrak{A}'}_{s'}$, and the inductive hypothesis. (We leave the details as an exercise.)

<u>Universal Quantifier</u>: $\forall vP.$ We assume as the <u>induction hypothesis</u> :

if ***s*** is a variable assignment for $\mathfrak{A}$, ***s′*** is a variable assignment for $\mathfrak{A}'_{\mathbf{L}}$, and ***s***$\subseteq$***s′***,
then $\Im^{\mathfrak{A}}_{s}(P)$= $\Im^{\mathfrak{A}'}_{s'}(P)$.

We show:

if ***s*** is a variable assignment for $\mathfrak{A}$, ***s***′ is a variable assignment for $\mathfrak{A}'_{\mathbf{L}}$, and $\boldsymbol{s}\subseteq\boldsymbol{s}'$,
then $\Im^{\mathfrak{A}}_{s}(\forall vP)= \Im^{\mathfrak{A}'}_{s'}(\forall vP)$.

Assume ***s*** is a variable assignment for $\mathfrak{A}$, ***s***′ is a variable assignment for $\mathfrak{A}'_{\mathbf{L}}$, and $\boldsymbol{s}\subseteq\boldsymbol{s}'$. Since $\boldsymbol{s}\subseteq\boldsymbol{s}'$, any *v*-variant of ***s″*** of ***s*** is a subset of some *v*-variant of ***s‴*** of ***s′***, and conversely any *v*-variant of ***s‴*** of ***s′***, contains as a subset some *v*-variant of ***s″*** of ***s.*** Hence, for any *v*-variant of ***s″*** of ***s***, $\Im^{\mathfrak{A}}_{s''}(P)$=T iff for any *v*-variant of ***s‴*** of ***s′***, $\Im^{\mathfrak{A}'}_{s'''}(P)$=T). Hence $\Im^{\mathfrak{A}}_{s}(\forall vP)$=T iff [by the def of $\Im^{\mathfrak{A}}_{s}$] (for any *v*-variant of ***s″*** of ***s***, $\Im^{\mathfrak{A}'}_{s''}(P)$=T) iff [by what was just proven] (for any *v*-variant of ***s‴*** of ***s′***, $\Im^{\mathfrak{A}'}_{s'''}(P)$=T) iff [by the def of $\Im^{\mathfrak{A}'}_{s}$] $\Im^{\mathfrak{A}'}_{s'}(\forall vP)$=T. **QED.**

Definitions

1. $\mathbf{F'_{FOL}}$ is an ***infinite extension*** of a language $\mathbf{F_{FOL}}$ iff $\mathbf{F_{FOL}}$ is a sublanguage of $\mathbf{F'_{FOL}}$ and $\mathbf{Vbls'_{FOL}}$ and $\mathbf{Cons'_{FOL}}$ each contain denumerably many new expressions not present in $\mathbf{Vbls_{FOL}}$ or $\mathbf{Cons_{FOL}}$, and let $\mathfrak{A}'$, $\mathfrak{B}'$, $\mathfrak{C}'$ range over the models of $\mathbf{F'_{FOL}}$

2. If $\mathbf{F'_{FOL}}$ is an ***infinite extension*** of a language $\mathbf{F_{FOL}}$, there a mapping *h* from the set $\mathbf{M'_{FOL}}$ of models of $\mathbf{F'_{FOL}}$ into the sets $\mathbf{M_{FOL}}$ of models of $\mathbf{F_{FOL}}$ is defined as follows:

   $h(\mathfrak{A}')$ is the unique $\mathfrak{A}$=<D,$\Im$> such that $\mathfrak{A}'$=<D,$\Im'$> and $\Im$ is the restriction of $\Im'$ to $\mathbf{Trms_{FOL}}$ and $\mathbf{Preds_{FOL}}$ of $\mathbf{F_{FOL}}$.

Clearly $h(\mathfrak{A}')$ meets to conditions for being a member of the set $\mathbf{M_{FOL}}$ of models of $\mathbf{F_{FOL}}$, and *h* maps $\mathbf{M'_{FOL}}$ onto $\mathbf{M_{FOL}}$.

3. Let $[\mathfrak{A}']_h$ be $\{\mathfrak{B}' \mid \mathfrak{B}'\in\mathbf{M_{FOL}}$ and $h(\mathfrak{B}')= h(\mathfrak{B}')\}$

4. Let $\equiv_h$ be the relation on the set of models $\mathbf{M'_{FOL}}$ of $\mathbf{F'_{FOL}}$ defined as follows:

$$\mathfrak{A}'\equiv_h\mathfrak{B}' \text{ iff, for some } \mathfrak{C}\in\mathbf{M'_{FOL}},\ \mathfrak{A}'\in[\mathfrak{C}']_h \text{ and } \mathfrak{B}'\in[\mathfrak{C}']_h$$

The results below follow directly from the previous metatheorem.

Corollary. The family $\{ [\mathfrak{A}']_h \mid \mathfrak{A}'$ is a model of $\mathbf{F'_{FOL}}\}$ is a partition of $\mathbf{M'_{FOL}}$ (*i.e.* no two members of the family have non-empty intersections and every model in $\mathbf{M'_{FOL}}$ is in some member of the family). Equivalently, $\equiv_h$ is an equivalence relation (*i.e.* $\equiv_h$ is reflexive, transitive, and symmetric).

Corollary. If *X* is a set of formulas of $\mathbf{F_{FOL}}$ and $\mathbf{F'_{FOL}}$ is an ***infinite extension*** of $\mathbf{F_{FOL}}$, then

X is satisfiable in $\mathbf{F_{FOL}}$ iff *X* is satisfiable in $\mathbf{F'_{FOL}}$.

(Clearly if *X* is satisfied by a model $\mathfrak{A}$ of $\mathbf{F_{FOL}}$, then there is some B′ of $\mathbf{F'_{FOL}}$, such that B′ satisfies *X,* namely any $\mathfrak{B}'$ such that $h(\mathfrak{B}')=\mathfrak{A}$. Conversely, if $\mathfrak{B}'$ of $\mathbf{F'_{FOL}}$ satisfies *X*, there is some of $\mathfrak{A}$ of $\mathbf{F_{FOL}}$, that satisfies *X*, namely that $\mathfrak{A}$ such that $h(\mathfrak{B}')=\mathfrak{A}$.)

The next metatheorem is interesting historically and conceptually. It states in the terminology of modern logic the logical "consequence" that in mediaeval logic was called the decent and assent from a universal to singulars. It also states the precise conditions under which a universally quantified formula is equivalent to the "infinite" conjunction of its instances. This equivalence is historically important in modern logic because it underlies what was ultimately a failed attempt the truth-conditions of the universal quantifier. By an ***instance*** of a universally quantified formula $\forall vP$ let us mean any formula $P[c/v]$ such that c is a constant. Since conjunctions must all be infinite in length there is really no such thing as an infinite conjunction in first-order syntax. But by the ***infinite conjunction of instance of*** $\boldsymbol{\forall vP}$ let us mean the set of all instances $\{ P[c/v] \mid c\in \mathbf{Cons_{FOL}}\}$. The so-called ***substitution interpretation*** of the quantifier is the following proposed statement of the truth-conditions for $\forall vP$:

$$\mathfrak{A}_{s} \models \forall vP \text{ iff, for all } c\in \mathbf{Cons_{FOL}},\ \mathfrak{A}_{s} \models P[c/v]$$

Though proposed by pioneers in formal semantics (*e.g.* by Carnap in *Meaning and Necessity*), it suffers from serious problems. First of all that there are some models with domains too big to be named by constants. There are, for example, domains that are non-countably infinite, like the set of real numbers. But by definition the set $\mathbf{Cons_{FOL}}$ is at most countably infinite. Hence there are some elements of a non-countable domain that would not have a name. Hence everything that does have a name in such a model might fall in the extension of a predicate P, and hence $\forall xPx$ would be true in that model given the substitution interpretation, yet there be elements of the domain that are not in the extension of P. Even in models that are countably infinite, the definition of a model does not require that everything in the domain be assigned by $\mathfrak{I}$ to some constant. Hence, the things named could all be in the extension of P and hence $\forall xPx$ would be true on the substitution interpretation, yet some things would fail to satisfy Px. In the special case in which everything in the domain is named by some constant the substitutional equivalence does hold. It is that fact that is shown in the following metatheorem.

Metatheorem 1-16. "Substitutional" Interpretation of the Quantifiers. If $\mathfrak{A}$=<D,$\mathfrak{I}$> is a model and $\mathfrak{I}$ maps $\mathbf{Cons_{FOL}}$ onto D (i.e. every element in D is assigned some constant from $\mathbf{Cons_{FOL}}$), then

$$\mathfrak{I}^{\mathfrak{A}}_{s}(\forall vP)=\text{T iff, for all } c\in \mathbf{Cons_{FOL}},\ \mathfrak{I}^{\mathfrak{A}}_{s}(P[c/v])=\text{T}.$$

Proof. If-part: assume for conditional proof that $\mathfrak{I}^{\mathfrak{A}}_{s}(\forall vP)$=T. Hence for any v-variant $\boldsymbol{s}'$ of $\boldsymbol{s}$, $\mathfrak{I}^{\mathfrak{A}}_{s'}(P)$=T. Let $\boldsymbol{s}''$ be the v-variant of $\boldsymbol{s}$ such that $\boldsymbol{s}''(v)=\mathfrak{I}(c)$. Then by universal instantiation, $\mathfrak{I}^{\mathfrak{A}}_{s''}(P)$=T. Since $\mathfrak{I}^{\mathfrak{A}}_{s''}(v)=\mathfrak{I}(c)=\mathfrak{I}^{\mathfrak{A}}_{s''}(c)$, by an earlier theorem $\mathfrak{I}^{\mathfrak{A}}_{s''}(P)=\mathfrak{I}^{\mathfrak{A}}_{s''}(P[c/v])$=T. Further since v does not occur in $P[c/v]$, $\mathfrak{I}^{\mathfrak{A}}_{s}(P)=\mathfrak{I}^{\mathfrak{A}}_{s''}(P[c/v])$=T. Then-part: Assume for all $c\in \mathbf{Cons_{FOL}}$, $\mathfrak{I}^{\mathfrak{A}}_{s}(P[c/v])$=T. We show that for any v-variant $\boldsymbol{s}'$ of $\boldsymbol{s}$, $\mathfrak{I}^{\mathfrak{A}}_{s'}(P)$=T. We do so by *reductio.* Assume that there is a v-variant $\boldsymbol{s}''$ of $\boldsymbol{s}$, $\mathfrak{I}^{\mathfrak{A}}_{s''}(P)$=F. Since $\mathfrak{I}$ maps $\mathbf{Cons_{FOL}}$ onto D, there is some constant, call it c′ such that $\mathfrak{I}(c')=\boldsymbol{s}''(v)$. Hence $\mathfrak{I}^{\mathfrak{A}}_{s''}(c')=\mathfrak{I}^{\mathfrak{A}}_{s''}(v)$, and by an earlier theorem since $\mathfrak{I}^{\mathfrak{A}}_{s''}(P)$=F, $\mathfrak{I}^{\mathfrak{A}}_{s''}(P[c'/v])$=F. By an earlier theorem, however, since v is not free in $P[c'/v]$, $\mathfrak{I}^{\mathfrak{A}}_{s''}(P[c'/v])=\mathfrak{I}^{\mathfrak{A}}_{s}(P[c'/v])$, absurd. **QED.**

The following metatheorem establishes the validity of the substitutability of identity.

Metatheorem 1-17 . Substitution of Identity.

For any formula P in $\mathbf{F}_{\mathbf{FOL}}$, if t and t' are terms that contain no occurrences of variables bound in P, $\mathfrak{A}$=<D,$\Im$> is a model in $\mathbf{M}_{\mathbf{FOL}}$, and $\boldsymbol{s}$ a variable assignment such that $\Im^{\mathfrak{A}}_{s}(t)= \Im^{\mathfrak{A}}_{s}(t')$, then $\Im^{\mathfrak{A}}_{s}(P)= \Im^{\mathfrak{A}}_{s}(P[t//t'])$, and hence $\Im^{\mathfrak{A}}_{s}(P)= \Im^{\mathfrak{A}}_{s}(P[t/t'])$,

Proof. Since $\mathbf{F}_{\mathbf{FOL}}$ is inductively defined the theorem is proven by induction.

Atomic Case. Assume the antecedent of the conditional to be proven. For an atomic formula $P^n t_1 \ldots\ t_n$ the consequent follows directly from the definitions of substitution and $\Im^{\mathfrak{A}}_{s}$, and the substitutivity of identity. (The details are left as an exercise.)

Molecular Cases

<u>Sentential Connectives.</u> The cases for the truth-functional connectives follow directly from $\Im^{\mathfrak{A}}_{s}$, the inductive hypothesis and the definition of substitution. (The details are left as an exercise.)

<u>Universal Quantifier:</u> $\forall vP$. Assume the antecedent of the conditional to be proven and the <u>inductive hypothesis</u>:

if the conditions in the antecedent are met,
then for all the immediate parts P of $\forall vP$, $\Im^{\mathfrak{A}}_{s}\ P)= \Im^{\mathfrak{A}}_{s}(P[t//t']$.

Then, $\Im^{\mathfrak{A}}_{s}(\forall vP)$=T iff (for any v-variant of $\boldsymbol{s'}$ of $\boldsymbol{s}$, $\Im^{\mathfrak{A}}_{s'}(P)$=T) [by the def of $\Im^{\mathfrak{A}}_{s}$] iff (for any v-variant $\boldsymbol{s'}$ of $\boldsymbol{s}$, $\Im^{\mathfrak{A}}_{s}(P[t//t'])$=T) [by the induction hypo.] iff $\Im^{\mathfrak{A}}_{s}(\forall vP[t//t'])$=T) [by the def of $\Im^{\mathfrak{A}}_{s}$]. **QED.**

It is by reference to this metatheorem that the standard natural deductions rules for identity are shown to be sound.

The next metatheorem states a somewhat obvious but useful fact. The interpretation of variables that do not occur in a formula are irrelevant to its truth.

Metatheorem 1-18

If $\mathfrak{A}$=<D,$\Im$> is a model in $\mathbf{M_{FOL}}$, $\boldsymbol{s}'$ is a *v*-variant of $\boldsymbol{s}$, and *v* is a variable of $\mathbf{F_{FOL}}$ that does not occur in *P*, then $\Im^{\mathfrak{A}}_{s}(P)=\Im^{\mathfrak{A}}_{s'}(P)$.

Proof. Since the theorem is really a universal quantification over the set of formulas $\mathbf{F_{FOL}}$, which is inductively defined, the proof is by induction.

Atomic Case. Assume $\mathfrak{A}$=<D,$\Im$> is a model, $\boldsymbol{s}'$ is a *v*-variant of $\boldsymbol{s}$, and *v* is a variable of $\mathbf{F_{FOL}}$ that does not occur in $P^n t_1\ldots t_n$. The consequent that $\Im^{\mathfrak{A}}_{s}(P^n t_1\ldots t_n)=\Im^{\mathfrak{A}}_{s'}(P^n t_1\ldots t_n)$ follows directly from the definitions of $\Im^{\mathfrak{A}}_{s}$, the substitutivity of identity, and the fact that since *v* does not occur in *P*, $\boldsymbol{s}$ and $\boldsymbol{s}'$ assign the same values to the terms in *P.*

Molecular Cases

Sentential Connectives. The cases for the connectives follow directly from $\Im^{\mathfrak{A}}_{s}$, the inductive hypothesis and the definition of substitution. (Details are left as an exercise.)

Universal Quantifier: $\forall v' P$. To simply the verbiage of the proof, let us use the abbreviation "$\boldsymbol{s'}=_v\boldsymbol{s}$" for "$\boldsymbol{s}'$ is a *v*-variant of $\boldsymbol{s}$." Assume as the induction hypothesis:

If $\mathfrak{A}$=<D,$\Im$> is in $\mathbf{M_{FOL}}$, $\boldsymbol{s'}=_v\boldsymbol{s}$, and *v* is a variable of $\mathbf{F_{FOL}}$ that does
not occur in *P*, then $\Im^{\mathfrak{A}}_{s}(P)=\Im^{\mathfrak{A}}_{s'}(P)$.

Assume also the antecedent of the conditional to be proven:

$\mathfrak{A}$=<D,$\Im$> is in $\mathbf{M_{FOL}}$, $\boldsymbol{s'}=_v\boldsymbol{s}$, and *v* is a variable of $\mathbf{F_{FOL}}$ that does
not occur in $\forall v' P$,"

By two conditional proofs we show that $\Im^{\mathfrak{A}}_{s}(\forall vP)$=T iff $\Im^{\mathfrak{A}}_{s'}(\forall vP)$=T. For the first assume the antecedent that (1) $\Im^{\mathfrak{A}}_{s}(\forall vP)$=T. Assume further for a *reductio* that the consequent is false, *i.e.* that (2) $\Im^{\mathfrak{A}}_{s'}(\forall vP)$=F. By (1) and the definition of $\Im^{\mathfrak{A}}_{s}$, it follows that for any $\boldsymbol{s''}$, if $\boldsymbol{s'}=_{v'}\boldsymbol{s}$, $\Im^{\mathfrak{A}}_{s''}(P)$=T. Let us consider one such, namely $\boldsymbol{s'''}$. Hence $\Im^{\mathfrak{A}}_{s'''}(P)$=T. By (2), there is some $\boldsymbol{s'''}$, $\boldsymbol{s''}=_{v'}\boldsymbol{s'}$, $\Im^{\mathfrak{A}}_{s'''}(P)$=F. Let us consider this $\boldsymbol{s'''}$. Hence $\Im^{\mathfrak{A}}_{s'''}(P)$=F. Now define $\boldsymbol{s''''}$ as follows: $\boldsymbol{s''''}(v')=\boldsymbol{s''}(v')$, $\boldsymbol{s''''}(v)=\boldsymbol{s'}(v)=\boldsymbol{s'''}(v)$, and for all v'' other than *v* and *v′*, $\boldsymbol{s''''}(v'')=\boldsymbol{s}(v'')=\boldsymbol{s'}(v'')=\boldsymbol{s'''}(v'')$. Now, $\boldsymbol{s''''}=_v\boldsymbol{s''}$ because $\boldsymbol{s''''}(v)=\boldsymbol{s'}(v)$, $\boldsymbol{s''}(v)=\boldsymbol{s}(v)$ and $\boldsymbol{s'}=_v\boldsymbol{s}$. Since $\boldsymbol{s''''}=_v\boldsymbol{s''}$ and $\Im^{\mathfrak{A}}_{s''}(P)$=T, it follows by the induction hypothesis that $\Im^{\mathfrak{A}}_{s''''}(P)$=T. On the other hand, $\boldsymbol{s''''}=_v\boldsymbol{s'''}$ because $\boldsymbol{s''''}(v)=\boldsymbol{s'}(v)=\boldsymbol{s'''}(v)$, and an assignment is a *v*-variant of itself. Since $\boldsymbol{s''''}=_v\boldsymbol{s'''}$ and $\Im^{\mathfrak{A}}_{s'''}(P)$=F, it follows by the induction hypothesis that $\Im^{\mathfrak{A}}_{s''''}(P)$=F. But $\Im^{\mathfrak{A}}_{s''''}(P)$=T and $\Im^{\mathfrak{A}}_{s''''}(P)$=F is a contradiction. Hence by *reductio*
$\Im^{\mathfrak{A}}_{s'}(\forall vP)\neq$F, and thus $\Im^{\mathfrak{A}}_{s'}(\forall vP)$=T. Thus by conditional proof, if $\Im^{\mathfrak{A}}_{s}(\forall vP)$=T then $\Im^{\mathfrak{A}}_{s'}(\forall vP)$=T. The converse is proven similarly. Hence $\Im^{\mathfrak{A}}_{s}(\forall vP)=\Im^{\mathfrak{A}}_{s'}(\forall vP)$. **QED.**

B. Soundness

We are now ready to proceed to the proof of the soundness and completeness results themselves. First we establish soundness by a straightforward inductive argument.

Metatheorem 1 -19 (Soundness). $\vdash_{\text{FOL=-ND}}\subseteq\vDash_{\text{FOL=}}$. Or equivalently:

For any *X* and *P*, if *X* $\vdash_{\text{FOL-ND}}$ *P* then *X*$\vDash_{\text{FOL =}}$*P*.

We show first that $\vdash\subseteq\vDash$:

for any <*X,P*>, if <*X,P*>$\in\vdash$, then <*X,P*>$\in\vDash$,

or equivalently,

for any <*X,P*>, if $X\vdash P$, then $X\vDash P$

Let us first review the general form of an inductively defined set and an inductive definition. We show that the open metalinguistic property "__$\in\vDash$" is true of every element of the constructive set $\vDash$. Recall that if C is a constructive set, it is defined by induction in terms of set BE of basic elements BE and some rules R_i:

Basis Clause. BE$\subseteq$C

Inductive Clause. If $e_1\in$C,…,$e_n\in$C and $R_i(e_1,…,e_n)=e_m$, then $e_m\in$C

In generally, to show some condition "$Q(x)$" in the metalangauge olds of every element in C, we give an inductive proof that falls into two parts.

Basis Step. Show: for all *e, if e*$\in$BE, then $Q(e)$.

Inductive Step. Assume (<u>induction hypothesis</u>): Q(e_1,…,Qe_n).
Show: $Q(e_m)$ where $R_i(e_1,…,e_n)=e_m$

Proof. We show the property "__$\in\vDash$" holds for every element <*X,P*> of the set $\vdash$.

Basis Step. We show "__$\in\vDash$" holds first for all elements <*X,P*> of the set **BD** of basic elements of $\vdash$. That is, we show:

for any <*X,P*>, if <*X,P*> in **BD**, then <*X,P*> $\in\vDash P$,

or in equivalent notation,

for any <*X,P*>, if <*X,P*> in **BD**, then $X\vDash P$.

It is clearly the case that if <*X,P*> in **BD**, then $X\vDash P$ because by the definition of **BP** if <*X,P*> is a basic deduction, then $X\in P$. Thus, if all the formulas in *X* are satisfied in a model, *P* will also be satisfied.

Inductive Step. For each rule R_i, assuming (as an induction hypothesis) that "__$\in\vDash$" is true of all the inputs of R_i, we must show that "__$\in\vDash$" is true of the output of R_i. Let <X_1,P_1>, …,<X_n,P_n> be the inputs of R_i, and <*Y,Q*> its output. We must show the conditional:

If "__$\in\vDash$" is true of each of <X_1,P_1>, …,<X_n,P_n>, then "__$\in\vDash$" is true of <*Y,Q*>.

This formulation of what is to be proven may be rephrased in two equivalent ways:

If <X_1,P_1>$\in\vDash$ and … and <X_n,P_n>$\in\vDash$, then <*Y,Q*>$\in\vDash$

If $X_1\vDash P_1$ and … and $X_n\vDash P_n$, then $Y\vDash Q$.

It is the latter of these formulations that is more customary. But the steps leading to it show that it is merely a way of stating the conditional that if the inputs deduction of the construction rule of $\vdash$ have the property in question (validity), the output deduction does to. It is this conditional that must be proven for each rule. Fortunately each such conditional has an antecedent that we can assume to be true (its induction hypothesis). In each case, the conditional is proven by an appeal to the induction hypothesis, the satisfaction conditions for the formulas in the deductions, and the definition of validity. Let us break the proof down, listing for each construction (inference) rules, the conditional that must be proven:

<u>Rule</u>:		<u>Inductive Hypotheses</u>		<u>To Prove</u>:
$\perp$-Introduction:	if	$X\vDash P$ and $X\vDash\sim P$,	then	$X\vDash\perp$
$\perp$-Elimination:	if	$X\vDash\perp$	then	$X\vDash P$
$\sim$-Introduction:	if	$X\vDash\perp$	then	X–{P}$\vDash\sim P$
$\sim$-Elimination:	if	$X\vDash\sim\sim P$	then	$X\vDash P$
$\wedge$-Introduction:	if	$X\vDash P$ and $Y\vDash Q$	then	$X\vDash P\wedge Q$
$\wedge$-Elimination:	if	$X\vDash P\wedge Q$	then	$X\vDash P$

	if	$X\models P\wedge Q$	then	$X\models Q$
$\vee$-Introduction:	if	$X\models P$	then	$X\models P\vee Q$
	if	$X\models Q$	then	$X\models P\vee Q$
$\vee$-Elimination:	if	$X\models P\vee Q,\ \ Y\cup\{P\}\models R,\ \ Z\cup\{Q\}\models R$	then	$X,Y,Z\models R$
$\rightarrow$-Introduction:	if	$X\models P$	then	$X-\{Q\}\models Q\rightarrow P$
$\rightarrow$-Elimination:	if	$X\models P\rightarrow Q,\quad Y\models P$	then	$X,Y\models Q$
$\forall$-Introduction:	if	$X\models P[t/v]$ & v is not free in any $P\in$X	then	$X\models\forall vP$
$\forall$-Elimination:	if	$X\models\forall v$P	then	$X\models P[t/v]$
$\exists$-Introduction:	if	$X\models P[t//v]$	then	$X\models\exists vP$
$\exists$-Elimination:	if	$X\models\exists vP,\ Y,P[t/v]\vdash Q$ &		
		v is not free in X$\cup Y\cup\{\exists vP,Q\}$	then	$X,Y\models Q$
=-Introduction	if	$X\models P$	then	$X\models t=t$
=-Elimination	of	$X\models P$ and $Y\models t=t'$	then	$X,Y\models P[t'//t]$

We prove one case as an example, $\perp$-Introduction. In set theoretic notation we must show:

If $<X,P>\in\models$ and $<X,\sim P>\in\models$ then $<X,\perp>\in\models$

Let us assume $X\models P$ and $X\models\sim P$. If both $X\models P$ and $X\models\sim P$, then X cannot every be satisfied. Hence the conditional if X is satisfied, then $\perp$ is satisfied is true because its antecedent is false. This holds for any model. Hence for any model if X is true in it so is $\perp$. But that means $X\models\perp$. QED.

C. Completeness

The first step in establishing the converse of soundness is the introduction of several proof theoretic concepts. The key idea of Henkin's proof is to define a set of sentences that is a proxy for a model in the sense that it contains all and only the sentences true in that model. In the early days of formal semantics before Tarski had introduced the idea of model in the sense we are using it, logicians actually used these sets, known then as ***state descriptions*** or ***model sets*** as the semantic proxies for possible worlds. As "worlds" however they have many of the drawbacks of the substitutional interpretation of the quantifier. They assume that all the objects in a world can be represented by a constant. In general, however, this assumption is not true – think of the "world" of the real numbers. There are however some worlds in which every entity is named by a unique constant. In the proof Henkin uses the sentences in the "state description" to describe one such model. The model is particularly interesting because its domain, the set of entities that exist in that world, are syntactic entities. They are the terms that appear in the formula of $\mathbf{F_{FOL}}$ itself used in the state description. Not only is the set of terms countable, and hence open to the possibility of having each of its elements assigned a name, but it is also ready to hand and clear. We know by construction that the set $\mathbf{Trms_{FOL}}$ exists. Moreover, since it is a set of syntactic entities, the nature and properties of its elements are particularly accessible epistemically. To a logician, referring to sets of syntactic entities is much preferable to referring to sets of entities that exist outside mathematics or that are drawn from more controversial parts of mathematics itself, like transfinite or non-constructible sets. Later we shall see that similar "syntactic" models, *i.e.* models in which the domain of objects consists of expressions of the syntax itself, figure prominently in Herbrand's techniques for proving first-order logic is undecidable, and in the metatheory underlying logic programming and computer languages like Prolog.

The proof progresses in steps. The relevant set of formulas is first shown to be consistent. It is then shown to be maximal in the sense that it contains every formula or its negation, Lastly, it is shown to be saturated in the sense that it contains a universal formula iff it contains all its instances, much as the substitutional interpretation would have it.

Definition. X is ***consistent*** iff not($X \vdash_{\textbf{FOL-ND}} \perp$)
iff not(for some finite subset Y of X, $Y \vdash \perp$)
iff for all finite subsets Y of X, not($Y \vdash \perp$)

Metatheorem 1-20. X is consistent iff not(for all P, $X \vdash P$), and for some P, not($X \vdash P$).
X is inconsistent iff for all P, $X \vdash P$.

Definition. X is ***maximally consistent*** iff, X is consistent and for any P, either $P \in X$ or $\sim P \in X$.

Metatheorem 1-21. If X is maximally consistent, then X is closed under $\vdash$ in the following sense:
for any P, if $X \vdash P$ then $P \in X$.

.

Corollary. If P and Q are alphabetic variants and X is maximally consistent,
then $P \in X$ iff $Q \in X$.

(The proof presupposes the fact that if $P \equiv Q$ then $P \vdash Q$ *and* $Q \vdash P$, which follows by induction from the definition of $\equiv$ and the rules for universal quantifier introduction and elimination.)

Definition. X is ***saturated*** iff, X is maximally consistent and
for any $\forall vP$ in $\mathbf{F_{FOL}}$, $\forall vP \in X$ iff (for all $c \in \mathbf{Cons_{FOL}}$, $P[c/v] \in X$).

Metatheorem 1-22. If X is saturated, then for any P, $P \in X$ iff $X \vdash P$.

Metatheorem 1-23. X is saturated iff, X is maximally consistent and for any $\exists vP$ in $\mathbf{F_{FOL}}$.,
$\exists vP \notin X$ iff (for some $c \in \mathbf{Cons_{FOL}}$, $\sim P[c/v] \in X$).

Henkin provides a general technique for expanding a consistent set to a saturated one. The construction consists of defining an inductive set from an initial consistent set taken as the set of basic elements of the construction. Once the set is constructed, it must be progressively shown that it is consistent, that it is maximal, and that it is saturated.

Metatheorem 1-24 (Expansion of Consistent to Saturated Sets.). If *Y* is consistent set of formulas of $\mathbf{F_{FOL}}$, then there is an infinite extension $\mathbf{F'_{FOL}}$ of $\mathbf{F_{FOL}}$ such that *Y* may be extended to (i.e. is a subset of) a saturated set Y^* in $\mathbf{F'_{FOL}}$..

Proof. We first construct some infinite extension $\mathbf{F'_{FOL}}$ of $\mathbf{F_{FOL}}$. We first specify two denumerable series: $\{v_i\}$ of variables and $\{c_i\}$ and constants that do not appear in formulas in $\mathbf{F_{FOL}}$, and add these to the terms of $\mathbf{F_{FOL}}$ to construct the set of formulas $\mathbf{F'_{FOL}}$. We assume the set $\mathbf{F'_{FOL}}$ may be placed in ordered as a denumerable series $\{P_i\}$. We now define a denumerable series $\{Y_i\}$ of subsets of $\mathbf{F'_{FOL}}$ by induction:

Basis Step: $Y_0=Y$.

Inductive Step: Assume Y_n is defined. We now define Y_{n+1} by first defining an intermediate set A_n as follows:

if $\{Y_n,\forall v_nP_n\}$ is consistent, then $A_n=Y_n\cup\{\forall v_nP_n\}$;
if $\{Y_n,\forall v_nP_n\}$ is inconsistent, then $A_n=Y_n\cup\{\sim P_n[c_n/v_n]\}$.

Note that c_n does not occur in Y_n. We now define Y_{n+1} in terms of Y_n and A_n.

if $A_n\cup\{P_n\}$ is consistent, then $Y_{n+1}=A_n\cup\{ P_n \}$;
if $A_n\cup\{P_n\}$ is inconsistent, then $Y_{n+1}=A_n$.

Note that this construction is unique, *i.e.* each $\{Y_n,\forall v_nP_n\}$ determines one and only one A_n and each $A_n\cup\{P_n\}$ one and only one Y_{n+1}.

The desired set Y^* is then defined as $\mathrm{U}\{Y_i\}$, i.e. as $\{ P_i|$ for some Y_i, $P_i\in Y_i\}$.

Claim 1: each Y_i is consistent.

Assume for a conditional proof that $Y_{n+1}\vdash\bot$. We show first that $A_n\vdash\bot$. By excluded middle, either $A_n,P_n\vdash\bot$ or not($A_n,P_n\vdash\bot$). If the latter, then $A_n\cup\{P_n\}$ is consistent and $Y_{n+1}=A_n\cup\{ P_n \}$. But then by the sub. of =, Y_{n+1} is consistent, contrary to our original assumption. Hence, $A_n,P_n\vdash\bot$ and $A_n\cup\{P_n\}$ is inconsistent. Thus, then $Y_{n+1}=A_n$. Hence, by sub. of = into the original assumption, $A_n\vdash\bot$. Now again by excluded middle, either $Y_n\cup\{\forall v_nP_n\}\vdash\bot$ or not($Y_n\cup\{\forall v_nP_n\}\vdash\bot$). If the latter $Y_n\cup\{\forall v_nP_n\}$ is consistent and $A_n=Y_n\cup\{\forall v_nP_n\}$. But then by $A_n\vdash\bot$ and sub., $Y_n\cup\{\forall v_nP_n\}\vdash\bot$ and $Y_n\cup\{\forall v_nP_n\}$ is inconsistent, which is absurd. Hence $Y_n\cup\{\forall v_nP_n\}\vdash\bot$ and $A_n=Y_n\cup\{\sim P_n[c_n/v_n]\}$. We show that $Y_n\vdash\bot$ by the following proof tree:

$$\dfrac{\dfrac{Y_n\cup\{\forall v_nP_n\}\vdash\bot\text{ (given)}}{Y_n\vdash\sim\forall v_nP_n}\sim+\qquad\dfrac{\dfrac{\dfrac{Y_n\cup\{\sim P_n[c_n/v_n]\}\vdash\bot\text{ (given)}}{Y_n\vdash\sim\sim P_n[c_n/v_n]}\sim-}{Y_n\vdash P_n[c_n/v_n]}\sim-}{Y_n\vdash\forall vP_n}\forall+}{Y_n\vdash\bot}\bot+$$

By conditional proof, then, if $Y_{n+1}\vdash\bot$, then $Y_n\vdash\bot$, for any *n*.

Assume now for a *reductio* that there is some set in Y^*, call it Y_{i+1}, that is inconsistent. But then Y_i is inconsistent, and similarly for all Y_k for k<i, including Y_0. But by the proof's original assumptionY_0 is consistent, an absurdity. Hence all Y_i are consistent.

Claim 2: Y^* is consistent. Suppose for *reductio* that $Y^*\vdash\bot$. Then for some finite subset *X* of Y^*, $X\vdash\bot$. Moreover, since *X* is finite, there is some $Y_i\subseteq Y^*$ such that $X\subseteq Y_i$. But since Y_i is consistent, so is *X*, which is absurd.

Claim 3: Y^* is maximal. Suppose for *reductio* that Y^* is not maximal. Then there is some *Q* such that neither $Q\in Y^*$ nor $Q\notin Y^*$. Since $Q\in\mathbf{F'_{FOL}}$, there are some *i* and *j* such that $Q=P_i$ and $\sim Q=P_i$. We show that both $Y_i,P_i\vdash\bot$ and $Y_j,P_j\vdash\bot$. Consider the first. Since $Q=P_i$ and $Q\notin Y^*$, $P_i\notin Y_{n+1}$. Now for a *reductio* suppose that $\{Y_i, P_i \}$ is consistent, and $Y_{n+1}=A_n\cup\{P_n\}$. Hence $P_n\in Y_{n+1}$, $P_n\in Y^*$ or in other words $P_n\in Y^*$ contrary to our assumption. Similarly, it is shown that $Y_j,P_j\vdash\bot$. Now, let *k* be the greater of *i* and *j*. Then both $Y_j\subseteq Y_k$ and$Y_j\subseteq Y_k$. Hence since both $\{Y_i, P_i\}$ and $\{Y_j,P_j\}$ are inconsistent so are $\{Y_k,P_i\}$ and $\{Y_k,P_j\}$, otherwise known as $\{Y_k,Q\}$ and $\{Y_k,\sim Q\}$,. We construct a tree:

$$\dfrac{\dfrac{Y_k,Q\vdash\bot\text{ (given)}}{Y_k\vdash{\sim}Q} \qquad \dfrac{\dfrac{Y_k,{\sim}Q\vdash\bot\text{ (given)}}{Y_k\vdash{\sim}{\sim}Q}\;{\sim}+}{Y_k\vdash Q}\;{\sim}-}{Y_k\vdash\bot}$$

But we have previously shown that Y_k is consistent. Hence, contrary to our assumption, Y^* is maximal.

Claim 4: Y^* is saturated. We show that for any $\forall v_nP$ in $\mathbf{F'_{FOL}}$, $\forall v_nP\in Y^*$ iff (for all $c\in\mathbf{Cons'_{FOL}}$, $P[c/v]\in Y^*$). Let $\forall v_nP$ be an arbitrary universal quantification in $\mathbf{F'_{FOL}}$. If Part: Assume $\forall v_nP\in Y^*$. Since Y^* is maximal, an earlier theorem assures that it is closed under $\vdash$. Therefore, $Y^*\vdash\forall vP$. Let c be an arbitrary constant in $\mathbf{Cons'_{FOL}}$. By $\forall$- it follows that $Y^*\vdash P[c/v]$. Since d is arbitrary this holds for all $c\in\mathbf{Cons'_{FOL}}$. Only-If Part: Let P be the n-th member of $\{P_i\}$, i.e. $P=P_n$, and assume for all $c\in\mathbf{Cons'_{FOL}}$, $P[c/v]\in Y^*$. For a *reductio* assume $\{Y^*,\forall v_nP_n\}$ is inconsistent. Then, $A_n= Y_n\cup\{{\sim}P_n[c_n/v_n]\}$, and ${\sim}P_n[c_n/v_n]\in Y^*$. But by universal instantiation, $P[c_n/v]\in Y^*$, and thus Y^* is inconsistent, contrary to what has previously been proven. Therefore, $\{Y^*,\forall v_nP_n\}$ is consistent. Hence, $A_n=Y_n\cup\{\forall v_nP_n\}$. But then $\forall v_nP_n\in A_n$ and thus $\forall v_nP_n\in Y^*$. **QED.**

Henkin now links the syntactically defined saturated set, to the semantics, showing that it is in fact the "state description" of some model. Proof is by construction of the appropriate model, *i.e.* by defining it. Once the model is defined, it is shown step by step that it satisfies the set. The model is defined in reference to a consistent set X. The domain of the model consists of the set $\mathbf{Trms_{FOL}}$ of terms, in the interpretation function $\mathfrak{I}$ of the model terms name themselves, and an n-place predicate P^n includes in its extension any n-tuple $<t_1,...,t_n>$ of terms that occur in some formula $P^nt_1,...,t_n$ in the reference set X and or in any alphabetic variant of a formula in X. The structure defined will be clearly meet the conditions for being a model. It will have a non-empty domain and an interpretation function assigning referents to constants and predicates. Moreover, it will be shown that the reference set is true in this model.

Metatheorem 1-25 (Satisfiability). If X is a saturated set of $\mathbf{F_{FOL}}$., then X is satisfiable in $\mathbf{F_{FOL}}$..

Proof. We construct a model $\mathfrak{A}$=<D,$\mathfrak{I}$> in $\mathbf{M_{FOL}}$ as follows: D=$\mathbf{Trms_{FOL}}$ and $\mathfrak{I}$ is defined as follows:

For any $c\in\mathbf{Cons_{FOL}}$, $\mathfrak{I}(c)$ = c;
For any $f^n\in\mathbf{Funcs_{FOL}}$, $\mathfrak{I}(f^n)$ = $\{<t_1,...,t_n>|\ t_{n+1}= f^n(t_1...t_n)\}$;
For any $P^n\in$ $\mathbf{Preds_{FOL}}$, $\mathfrak{I}(P^n)$ = $\{<t_1,...,t_n>|$ all alphabetic variants Q of $P^nt_1,...,t_n\in X$ such that $Q\in X\}$

Lemma. For all P [(for all variable assignments of $\boldsymbol{s}$, $\mathfrak{I}^{\mathfrak{A}}_{s}(P)$=T) iff (for some variable assignments of $\boldsymbol{s}$, $\mathfrak{I}^{\mathfrak{A}}_{s}(P)$=T)]. [The lemma asserts that for this particular statement the universal case is equivalent to the existential.]

Atomic Case. If Part: trivial. Only-If Part: Assume for some $\boldsymbol{s}$, call it $\boldsymbol{s'}$, that $\mathfrak{I}^{\mathfrak{A}}_{s'}(P^nt_1,...,t_n)$=T and assume further that $\boldsymbol{s''}$ is an arbitrary variable assignment. Observe that for any variable assignments of $\boldsymbol{s'}$ and $\boldsymbol{s''}$, $P^n\mathfrak{I}^{\mathfrak{A}}_{s'}(t_1),...,\ \mathfrak{I}^{\mathfrak{A}}_{s'}(t_n)$ and $P^n\ \mathfrak{I}^{\mathfrak{A}}_{s''}(t_1),...,\ \mathfrak{I}^{\mathfrak{A}}_{s''}(t_n)$ are alphabetic variants, hence since X is maximally consistent we know by an earlier metatheorem that $P^n\ \mathfrak{I}^{\mathfrak{A}}_{s'}(t_1),...,\ \mathfrak{I}^{\mathfrak{A}}_{s'}(t_n)\in X$ iff
$P^n\mathfrak{I}^{\mathfrak{A}}_{s''}(t_1),...,\ \mathfrak{I}^{\mathfrak{A}}_{s''}(t_n)\in X$. Therefore,

$\mathfrak{I}^{\mathfrak{A}}_{s'}(P^nt_1,...,t_n)$=T only if $<\mathfrak{I}^{\mathfrak{A}}_{s'}(t_1),...,\ \mathfrak{I}^{\mathfrak{A}}_{s'}(t_n)>\in\ \mathfrak{I}^{\mathfrak{A}}_{s'}(P^n)$
only if $<\mathfrak{I}^{\mathfrak{A}}_{s'}(t_1),...,\ \mathfrak{I}^{\mathfrak{A}}_{s'}(t_n)>\in\{<t_1,...,t_n>|\ P^nt_1,...,t_n\in X\}$
only if $P^n\mathfrak{I}^{\mathfrak{A}}_{s'}(t_1),...,\ \mathfrak{I}^{\mathfrak{A}}_{s'}(t_n)\in X$
only if $P^n\mathfrak{I}^{\mathfrak{A}}_{s''}(t_1),...,\ \mathfrak{I}^{\mathfrak{A}}_{s''}(t_n)\in X$

Since s'' is an arbitrary variable assignment, we may generalize for all s, $\mathfrak{I}^{\mathfrak{A}}_{s}(P^n t_1,...,t_n)$=T.

Molecular Case. Inductive Hypothesis: for any immediate part Q of $P\in\mathbf{F}_{\mathbf{FOL}}$,
for all variable assignments of s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T iff for some
variable assignments of s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T

Negation. If Part: trivial. Only-If Part: Assume for some s, $\mathfrak{I}^{\mathfrak{A}}_{s}(\sim Q)$=T iff for some s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=F iff [by the induction hypo.] for all s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=F iff for all s, $\mathfrak{I}^{\mathfrak{A}}_{s}(\sim Q)$=T.
Conjunction. If Part: trivial. Only-If Part: Assume for some s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q\wedge R)$=T. Then for some s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T, and for some s, $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=T. Hence, by the induction hypo., for all s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T, and for all s, $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=T. Hence, for all s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T, and $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=T, *i.e.* $\mathfrak{I}^{\mathfrak{A}}_{s}(Q\wedge R)$=T.
Disjunction. If Part: trivial. Only-If Part: Assume for some s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q\vee R)$=T. Then for some s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T or $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=T. Assume it is s' and Q such that $\mathfrak{I}^{\mathfrak{A}}_{s'}(Q)$=T. Thus, $\mathfrak{I}^{\mathfrak{A}}_{s'}(Q\vee R)$=T. Then by the induction hypo., for all s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q\vee R)$=T. Likewise if it is it is s' and R such that $\mathfrak{I}^{\mathfrak{A}}_{s'}(R)$=T, it follows that $\mathfrak{I}^{\mathfrak{A}}_{s}(Q\vee R)$=T.
Conditional. If Part: trivial. Only-If Part: Assume for some s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q\rightarrow R)$=T. Then for some s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=F or $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=T. . Assume it is s' and Q such that $\mathfrak{I}^{\mathfrak{A}}_{s'}(Q)$=F. Thus, $\mathfrak{I}^{\mathfrak{A}}_{s'}(Q\rightarrow R)$=T. Then by the hypo., for all s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q\rightarrow R)$=T. Likewise if it is s' and R such that $\mathfrak{I}^{\mathfrak{A}}_{s'}(R)$=T, it follows that $\mathfrak{I}^{\mathfrak{A}}_{s}(Q\rightarrow R)$=T.
Universal Quantifier. If Part: trivial. Only-If Part: Assume for some s, $\mathfrak{I}^{\mathfrak{A}}_{s}(\forall vQ)$=T. Then by an earlier theorem, for all s, $\mathfrak{I}^{\mathfrak{A}}_{s}(\forall vQ)$=T. **QED.**

The main theorem will follow directly from the following lemma, which we are now ready to show by induction:

Lemma. For any $P\in\mathbf{F}_{\mathbf{FOL}}$, $\mathfrak{A}\models P$ iff $P\in X$.

Atomic Case.

$\mathfrak{A}\models P^n t_1,...,t_n$ iff for all variable assignments s, $\mathfrak{I}^{\mathfrak{A}}_{s}(P^n t_1,...,t_n)$=T [def of $\mathfrak{A}\models P^n t_1,...,t_n$]
iff for all variable assignments s, $<\mathfrak{I}^{\mathfrak{A}}_{s}(t_1),..., \mathfrak{I}^{\mathfrak{A}}_{s}(t_n)>\in \mathfrak{I}^{\mathfrak{A}}_{s}(P^n)$
[def of $\mathfrak{I}^{\mathfrak{A}}_{s}(P^n t_1,...,t_n)$=T]
iff for all variable assignments s,
$<\mathfrak{I}^{\mathfrak{A}}_{s}(t_1),..., \mathfrak{I}^{\mathfrak{A}}_{s}(t_n)> \in \{<t_1,...,t_n>|\ P^n t_1,...,t_n\in X$ [def of $\mathfrak{I}^{\mathfrak{A}}_{s}$]
iff for all variable assignments s, $P^n\mathfrak{I}^{\mathfrak{A}}_{s}(t_1),..., \mathfrak{I}^{\mathfrak{A}}_{s}(t_n)\in X$ [Principle of Abstraction]
iff for the variable assignments s' such that for any variable v, $s'(v)=v$,
$P^n\mathfrak{I}^{\mathfrak{A}}_{s'}(t_1),..., \mathfrak{I}^{\mathfrak{A}}_{s'}(t_n)\in X$ [by universal instantiation and the lemma]
iff $P^n t_1,...,t_n\in X$ [by the def of s' for any t_i of $t_1,...,t_n$, $\mathfrak{I}^{\mathfrak{A}}_{s'}(t_i)= t_i$]

Molecular Case. Inductive Hypothesis: for any immediate part Q of $P\in\mathbf{F}_{\mathbf{FOL}}$,
$\mathfrak{A}\models Q$ iff $Q\in X$.
for all variable assignments s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T iff $Q\in X$.
for some variable assignments s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=F iff $Q\notin X$.

Negation. $\mathfrak{A}\models\sim Q$ iff all s, $\mathfrak{I}^{\mathfrak{A}}_{s}(\sim Q)$=T iff for all s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=F iff [by lemma] some s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=F iff [by hypo.] $Q\notin X$ iff $\sim Q\in X$.
Conjunction. $\mathfrak{A}\models Q\wedge R$ iff all s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q\wedge R)$=T iff, for all s, ($\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T and $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=T) iff, (for all s, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T) and (for all s, $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=T) iff ,$Q\in X$ and $R\in X$ iff [by closure of X under $\vdash$] $Q\wedge R\in X$.

<u>Disjunction.</u> $\mathfrak{A}\models Q\lor R$ iff all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q\land R)$=T iff, for all ***s***, ($\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T or $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=T) iff, (for some ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T) or (for some ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=T) iff [by lemma] (for all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T) or (for all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=T) iff, [by hypo] $Q\in X$ or $R\in X$. <u>If Part</u>: Assume $\mathfrak{A}\models Q\lor R$. But then $Q\in X$ or $R\in X$. in either case [by closure of X under $\vdash$] $Q\lor R\in X$. <u>Only-If Part</u>: Assume: $Q\lor R\in X$. Hence $X\vdash Q\lor R$. Assume for a *reductio* that neither $Q\in X$ nor $R\in X$. Hence ${\sim}Q\in X$ and ${\sim}R\in X$, and $X\vdash{\sim}Q$ and $X\vdash{\sim}R$. But then X is inconsistent as the following proof tree demonstrates:

$$\dfrac{\dfrac{Q\vdash Q\text{ (basic)}\qquad X\vdash{\sim}Q\text{ (given)}}{X,Q\vdash\bot}\;{+\bot}\qquad\dfrac{R\vdash R\text{ (basic)}\qquad X\vdash{\sim}R\text{ (given)}}{X,R\vdash\bot}\;{+\neg}\qquad X\vdash Q\lor R}{X\vdash\bot}\;{\lor-}$$

But since X is consistent by assumption, we have a contradiction. Thus, either $Q\notin X$ or $R\notin X$. That is, by the hypo. either for some ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T or for some ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=T. Hence by the lemma for all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T or, for all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=T. Thus, by quantifier logic, for all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=T or $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=T, or in other words, there is no $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=F and $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=F. Suppose now for a *reductio* that it is not the case that $\mathfrak{A}\models Q\lor R$. That is, there is some ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q\lor R)$=F. Thus, for some ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=F and $\mathfrak{I}^{\mathfrak{A}}_{s}(R)$=F. But this contradicts what we have just shown. Hence, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q\lor R)$=T.

<u>Conditional.</u> We show first: $X\vdash{\sim}Q$ or $X\vdash S$, iff $X\vdash Q\to S$. Suppose first that $X\vdash{\sim}Q$ or $X\vdash S$. Now either $X\vdash S$, $X\vdash{\sim}Q\to S$ by →+. On the first alternative $X\vdash Q\to S$ follows as the following tree establishes:

$$\dfrac{\dfrac{\dfrac{X\vdash{\sim}Q\text{ (given)}}{X\vdash{\sim}Q\lor S}\;{\lor+}\qquad S\vdash S\qquad\dfrac{Q\vdash Q\text{ (basic)}\qquad{\sim}Q\vdash{\sim}Q\text{ (basic)}}{Q,{\sim}Q\vdash S)}\;{\bot+}}{X,Q\vdash S)}\;{\lor-}}{X\vdash Q\to S}\;{\to+}$$

On the other alternative $X\vdash Q\to S$ follows by →+. Conversely, if neither $X\vdash{\sim}Q$ or $X\vdash S$ in a maximally consistent set X, then ${\sim}Q\notin X$ and $S\notin X$. But then ${\sim}S\in X$ and hence $X\vdash{\sim}S$. But then X is inconsistent. Hence, $X\vdash{\sim}Q$ or $X\vdash S$, iff $X\vdash Q\to S$. Now, $Q\to S\in X$ iff $X\vdash Q\to S$ [by closure] iff ($X\vdash{\sim}Q$ or $X\vdash S$) [by what was shown above] iff (${\sim}Q\in X$ or $S\in X$) [by closure] iff ($Q\notin X$ or $S\in X$) [by maximality] iff (for some ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=F) or (for all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(S)$=T) [by induction hypo.] iff (for all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=F) or (for all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(S)$=T) [by lemma] iff (for all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q)$=F or $\mathfrak{I}^{\mathfrak{A}}_{s}(S)$=T) [by quantifier logic] iff (for all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q\to S)$=T) [by definition] iff $\mathfrak{A}\models Q\to R$.

<u>Universal Quantifier</u>. $\mathfrak{A}\models\forall vQ$ iff for all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(\forall vQ)$=T [by definition] iff for all ***s***, for all $c\in$ $\mathbf{Cons}_{\mathbf{FOL}}$, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q[c/v])$=T [by an earlier theorem] iff for all $c\in$ $\mathbf{Cons}_{\mathbf{FOL}}$, for all ***s***, $\mathfrak{I}^{\mathfrak{A}}_{s}(Q[c/v])$=T [by quantifier logic] iff for all $c\in$ $\mathbf{Cons}_{\mathbf{FOL}}$, $Q[c/v]\in X$ [by induction hyp.] iff $\forall vQ\in X$ [by saturation]. **QED.**

Corollary. If X is consistent, then X is satisfiable.

Proof. Let X be a consistent set of $\mathbf{F}_{\mathbf{FOL}}$. Let X be expanded to a saturated superset Y^* of $\mathbf{F'}_{\mathbf{FOL}}$ as in the next to last theorem. By the previous theorem, then, Y^* is satisfiable. By a yet earlier theorem, then, X of $\mathbf{F}_{\mathbf{FOL}}$ is also satisfiable. **QED.**

Lastly, the pieces of the proof are brought together. It is shown by appeal to the definitions of satisfiability, valid argument and consistency that the two metatheorems previously proven establish the completeness result.

Metatheorem 1-26

For any X and P of $\mathbf{F_{FOL}}$, $X \vdash_{\mathbf{FOL}} P$ iff $X \vDash P$.

Proof. The <u>If Part</u> follows by the soundness metatheorem. <u>Only-If Part</u>. We note first the following consequences of the definitions:

$X \vdash_{\mathbf{FOL}} P$ iff for some finite subset Y of X, $Y \vdash P$
iff for some finite subset Y of X, $Y, {\sim}P \vdash \perp$
iff for some finite subset Y of $X \cup \{{\sim}P\}$, $Y \vdash \perp$
iff some finite subset Y of $X \cup \{{\sim}P\}$ is (finitely) inconsistent
iff $X \cup \{{\sim}P\}$ is inconsistent

Assume $X \vDash P$. Then,

$X \vDash P$ iff $X \cup \{{\sim}P\}$ is not satisfiable
only if $X \cup \{{\sim}P\}$ is inconsistent (by previous theorem)
iff $X \vdash_{\mathbf{FOL}} P$ **QED.**

Metatheorem 1-27 (Finite Deductibility).

$X \vdash_{\mathbf{FOL\text{-}ND}} Q$, iff there is some finite subset $\{P_1, \ldots P_n\}$ of X such that $P_1, \ldots P_n \vdash_{\mathbf{FOL\text{-}ND}} Q$.

Metatheorem 1-28

Equivalence of Axiomatic and Natural Deduction Theory.

$X \vdash_{\mathbf{FOL}} Q$, iff $X \vdash_{\mathbf{FOL\text{-}ND}} Q$

The result follows trivially from the definition of $X \vdash_{\mathbf{FOL}} Q$.

D. Further Metatheorems

The semantic fact corresponding to finite deducibility is called finitary semantic entailment or compactness. Proving it directly without appealing to facts about syntax is quite non-trivial. Here however it will sufficed to remark that it follows directly from the soundness and completeness theorems together with finite deducibility,

Metatheorem 1-29 (Compactness)

Entailment Formulation:

X $\vDash$ Q iff, there is some finite subset $\{P_1, \ldots P_n\}$ of X such that $P_1, \ldots P_n \vDash$ Q.

Satisfiability Formulation:

X is satisfiable iff every finite subset of X is satisfiable.

Lastly we state a metatheorem that is an intriguing “inexpressibility result,” the proof of which will not be given here. It states that, in certain respects, we cannot tell from the sentences true in a model how many entities exist in the model.

First, if a model is infinite, we cannot tell how big an infinite set its domain is. Although infinite sets may vary in size -- because as Cantor taught us there are some infinite sets that are bigger than others -- what is called the "downward" part of the theorem tells us we cannot distinguish among these in worlds if we are writing about them only in the language of first-order logic. If a set of sentences (*e.g.* the axioms of some science) are true in a countably infinite world, then it is true in an infinite world of any size whatever. We call a domain ***countably infinite*** iff it may be put in one to one correspondences with the natural numbers. We call it ***non-countable*** iff one of its subsets may be put in one to one correspondence with the natural numbers but the set as a whole cannot. An example of a non-countably infinite set is the set of real numbers.

Second, if our syntax lacks the identity predicate, we cannot using formulas written in the syntax of first-order logic, even limit the number of objects in the domain to a finite number. We cannot make up a formula equivalent to saying "there are only 20 entities" in such a way that that formula is true iff the domain has only twenty entities in it. The "upward" part of the theorem tells us that no matter what formula we write there is some infinite world that makes it true.

Metatheorem 1-30 (Skolem-Löwenheim)

Downward Skolem-Löwenheim Theorem. *X* is satisfiable in some model with a countably infinite domain if, and only if, *X* is satisfiable in some model with a non-countably infinite domain.

Upward Skolem-Löwenheim Theorem. If *X* is satisfiable in some model with a finite domain, then *X* is satisfiable in some model with a countably infinite domain.

Note that once we introduce the identity predicate into the language, the upward part of the theorem fails to hold. Using identity we can express the cardinality of the domain. For example,

$$\exists x\exists y\exists z(x\neq y\wedge y\neq z\wedge x\neq z\wedge\forall w(w=x\vee w=y\vee w=z))$$

says there are exactly three things in the domain in the sense that one can prove the following:

Metatheorem 1-31. For, $\mathfrak{A}$=<D,$\mathfrak{I}$>

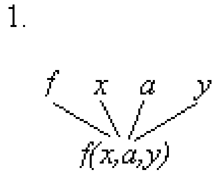
$\mathfrak{A}\models\exists x\exists y\exists z(x\neq y\wedge y\neq z\wedge x\neq z\wedge\forall w(w=x\vee w=y\vee w=z))$ iff the cardinality of D is 3

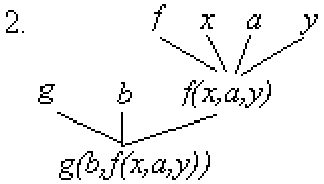
III. Exercises

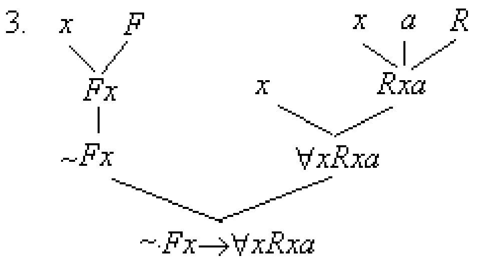
A. Skills

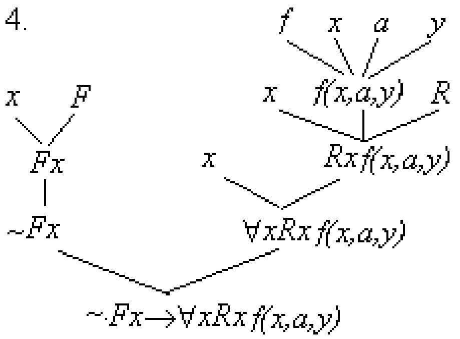
i. Syntactic Construction Trees.

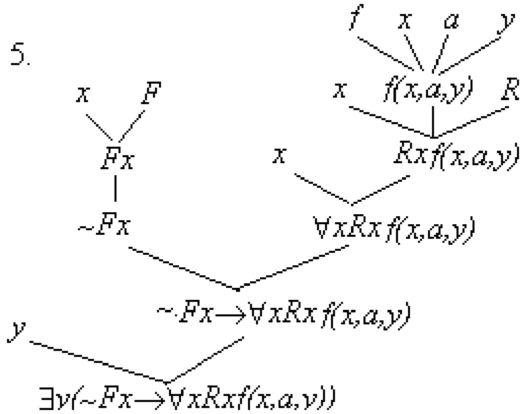
<u>*Examples of ConstructionTrees*</u>











Above are examples of construction trees. The "leaf" nodes are occupied by atomic descriptive expressions (constants, variables, or predicates). Each descendent node is occupied by a non-logical expression (term or formula). Each descendant consists of the application of a formation rule from the syntax. The inputs to the rule are the expressions in left to right order occupying the immediately superior nodes and the output of the rule is the expression occupying the descendent node. The tree thus records in a step by step process the stages by which the expression on the root (bottom) node is "constructed" and added to the set of formulas.

<u>*Bound and Free Occurrences.*</u> Exercise 1. In the trees above:

a. Circle each ***variable*** on a leaf (node) that is ***free*** (*i.e.* is above no formula beginning with ∀*x* or ∃*x*).
b. Put a square around each ***variable*** on a leaf that is ***free for y*** (*i.e.* is above no formula beginning with ∀*y* or ∃*y*).
c. Circle each ***constant*** on a leaf that is ***free for x.***
d. Put a square around each ***constant*** on a leaf that is ***free for y***.

<u>Drawing Trees</u>. Exercise 2. Below are formulas *P*[*e*/*e*′] in which and expression *e* is substituted for another *e*′ in an original formula *P.* For each draw a construction tree for the formula *P* as it is before any substitution is made. Then, carefully draw into the tree *each* occasion of substitution. You do this by altering the expressions on the nodes. Start systematically at each top (leaf) node of the tree. If the substitution of *e* for *e*′ is well defined for the expression at that node, cross out *e*′ on that node and carefully write above it *e.* Proceed down the tree's branches, replacing *e*′ by *e* at each node (crossing out *e*′ and writing above it *e*) in the increasingly complex formulas that occupy the tree's nodes, but do not complete a replacement at a node if the substitution of *e* for *e*′ is not well defined for the expression at that node. If you arrive at a node at which the substitution is not well defined, circle that node and halt any further substitution beneath that node.

a. $(\sim Fx\rightarrow\forall xRxa)[a/y]$
b. $(\sim Fx\rightarrow\forall xRxa)[a/x]$
c. $(\sim Fx\rightarrow\forall xRxf(x,a,y))[\ f(x,a,y)/a]$
d. $\exists y(\sim Fx\rightarrow\forall xRxf(x,a,y))[x/y]$
e. $\exists y(\sim Fx\rightarrow\forall xRxf(x,a,y))[\ f(x,a,y)/x]$
f. $\exists y(\sim Fx\rightarrow\forall xRxf(x,a,y))[\ f(b,a,z)/a]$

<u>*3. Partial and Total Substitution in Trees*</u>. Exercise 3. There are two notions of substitution of *e* for *e*′ in *P: e*ither an expression *e* replaces *every* occurrence of an expression *e*′ and the result is written *P*[*e*/*e*′], or *e* replaces *some* (*i.e.* one or more) occurrence of *e*′ in *P* and the result is *P*[*e*//*e*′]. For each of the formulas below draw a construction tree exhibiting the difference in their syntactic structure. Draw the construction tree for the formula before the substitutions are made and then indicate on the tree each substitution-at-a-node by crossing out the expression replaced and writing above it the expression that replaces it. If at some node the substitution is undefined, circle that node and write in a short explanation of why the substitution is undefined.

a. $\exists y(\sim Fg(x,a)\rightarrow\forall xRxf(x,a,y))[b/a]$
$\exists y(\sim Fg(x,a)\rightarrow\forall xRxf(x,a,y))[b//a]$

b. $(\sim Fg(x,a)\wedge Rzxx)\rightarrow\forall xRxf(x,a,y))[\ g(x,a)/x]$
$(\sim Fg(x,a)\wedge Rzxx)\rightarrow\forall xRxf(x,a,y))\ [\ g(x,a)//x]$

<u>Alphabetic Variation in Trees</u>. Exercise 4.

a. Draw a tree for $\exists y(\sim Fx\rightarrow\forall xRxf(x,a,y))[z/y]$.
b. Why is it an alphabetic variant of $\exists y(\sim Fx\rightarrow\forall xRxf(x,a,y))$
but $\exists y(\sim Fx\rightarrow\forall xRxf(x,a,y))[x/y]$ is not?

ii. Proof Theory.

Constructing Classical Proof Trees. Exercise 5. Using the rules for classical natural deduction in first-order logic (pages 125 and 126), construct a proof tree (like those on page 127) for each of the following:

a. $P{\wedge}{\sim}Q \vdash {\sim}(P{\rightarrow}Q)$

b. $P{\rightarrow}Q \vdash {\sim}P{\vee}Q$

Intuitionistic Proofs. Exercise 6. By referring to the classical proof tree for $P{\rightarrow}Q \vdash {\sim}P{\vee}Q$ (above), explain why it cannot be proven using the rule set for intuitionistic logic.

iii. Semantic Entailment and Validity

Semantic Metatheorems on Validity. Give informal proofs in the metalanguage of the following. Set up the proofs when applicable as conditional proofs or reductions to the absurd. Reformulate assertions by applying definitions (*e.g.* of $X \vDash_{\mathbf{FOL}} P$, $\mathfrak{A} \vDash P$, $\Im^{\mathfrak{A}}_{s}$ **,** ***s***, and $\Im$). In particular, reformulate assertions of the form $\Im^{\mathfrak{A}}_{s}(P)$=T into facts about the referents, under $\Im$ and ***s***, of the atomic parts of *P.*

Exercise 7

a. $\forall x(Fx{\rightarrow}Gx) \vDash_{\mathbf{FOL}} \forall xFx{\rightarrow}\forall xGx$

b. $\forall xFx{\rightarrow}\forall xGx \nvDash_{\mathbf{FOL}} \forall x(Fx{\rightarrow}Gx)$

c. $\forall x(Fx{\rightarrow}Gx), \exists x\neg Gx \vDash_{\mathbf{FOL}} \exists \mathrm{x}\neg Fx$

d. $\vDash_{\mathbf{FOL}} \forall x\exists y(x{=}y)$

e. $Fa \vDash_{\mathbf{FOL}} \exists xFx$

f. If $\vDash_{\mathbf{FOL}}$ $P{\rightarrow} Q$ then $P \vDash_{\mathbf{FOL}}$ $Q.$
(Note: the converse holds if *P* and *Q* are sentences.)

g. $\forall x(Hx{\rightarrow}Gx), \forall x(Fx{\rightarrow}Gx) \nvDash_{\mathbf{FOL}} \forall x(Fx{\rightarrow}Hx)$

iv. Inductive Proofs

The general form of an inductive definition of a set C. One or more basic sets $B_1,\ldots,B_n$ and rules (relations) $R_1,\ldots,R_m$ defined. Then C is defined by three clauses:

Basis Clause. All basic sets $B_1,\ldots,B_n$ are subsets of C;

Inductive Clause. For any rule R_i, if R_i is an n+1-place relation, then for any $x_1,\ldots,x_n,x_{n+1}$, if $x_1,\ldots,x_n$ are all in C and $<x_1,\ldots,x_n,x_{n+1}>\in R_i$, then $x_{n+1}\in$C.

Closure Clause. Nothing else is in C.

General Form of Steps in an Inductive Proof

To Prove: For all *x,* if $x\in$C then $P[x]$.

Basis Steps (one step for each i=1,…,n).

Step *i.* Assume x is arbitrary and that $x\in B_i$.

Show: $P[x]$.

Inductive Steps (one step for each i=1,…,m).

Step *i* **Induction Hypothesis**. Let $x_1,\ldots,x_n,x_{n+1}$ be arbitrary. Assume: $P[x_1],\ldots, P[x_n]$

Show: If $<x_1,\ldots,x_n,x_{n+1}>\in R_i$ then $P[x_{n+1}]$.

Proofs by Induction Exercise 8. Complete the inductive proof for both the atomic case and the cases of the sentential connectives in Metatheorem 1-17 . Substitution of Identity. Complete the proof also for the cases of the sentential connectives in Metatheorem 1-18.

Do so methodically. For each metatheorem write down on a page the numbers 1-6 and complete the following steps. (This exercise will take up paper, but it is worth taking the case to spell out the details):

1. Name the inductive set C that is at issue.
2. Name the set of basic elements of C.
3. State carefully the open metalinguistic sentence *P*[*x*] that is being shown to hold for all *x* in C.
4. List all the assumptions that given the formulation of the metatheorem you are allowed to assume before you start to prove “for all *x* in C, *P*[*x*]”.
5. For Metatheorem 1-17 . Substitution of Identity. P$^{\tau}$rove the atomic case.
6. For each sentential construction rule used to define C:
    a. Write out the inductive hypothesis that you may assume for that rule. That is, write out what it means to say that the inputs of that rule have *P*[*x*].
    b. Prove the inductive step. That is, given the inductive hypothesis (that the inputs of the rule satisfy *P*[*x*]), prove that the output of the rule satisfies *P*[*x*]). You will need to refer to

several thighs: (1) the definition of the construction rule itself (*i.e.* to what the rule does to the inputs), (2) the inductive hypothesis for that rule, and (3) other facts that are relevant.

v. Metatheorems

Exercise 9. The metatheorems pages 130-139 are only of marginal interest in themselves. They are proven in the text, however, because each is crucial later at some point in the proofs of the main completeness theorems (page 139 and later). As you read later proofs, note (page and line number) where each of the earlier metatheorems (note its number) is used.

B. Theory and Ideas

i. Existence

Exercise 10. In formal languages a distinction is drawn between descriptive (categorematic) and logical (syncategorematic) terms. Descriptive terms are assigned referents or truth-values. Logical signs do not directly have referents or truth-values. Rather each is represented in the syntax by its own grammar rule and in the semantics by a corresponding clause the definition of $\Im$ (in sentential logic) or $\Im^{\mathfrak{A}}_{s}$ (in first-order logic). The grammatical clause explains how the logical sign is used to build up a "whole" expression from immediate parts, and its semantic clause states the conditions that determine the referent or truth-value of the whole given the referents or truth-values of its immediate parts. But it is hard to decide sometimes whether an expression in natural language is descriptive or logical. Consider the verb *to exist* in English. Write a short discussion on whether it is descriptive or logical. Here are some points to consider.

- Suppose an "existence predicate" E (read *exists*) were added to first-order syntax as, say, $P^1{}_1$. What would $\Im(E)$ be? How might it be defined? Would it have the same definition in every model?
- Are there any valid "logical" arguments in English that turn on the syntax of *exists, i.e.* ones that can be described in purely syntactic terms or that hold because of the shape of the formulas regardless of which descriptive constants, functors, or predicates occur?
- If *exists* occur is the syntax, it is used to combine with immediate parts to form a whole. What part of speech is this "whole" and what are the parts of speech of the immediate parts it combines with? What sort of entities do the whole and parts refer to? Things? Sets? Relations? Truth-values? Can you formulate a semantic rule that would tell you what the whole (made up using *exists*) refers to give what the parts refer to?
- Translate $\models_{\mathbf{FOL=}} \exists y(x{=}y)$ into English. Discuss whether it mean the same as *x exists?*

ii. Truths of Logic and Valid Arguments

Exercise 11. Presumably we have some idea from real life and natural language what arguments are valid and what sentences are trivially true before we ever study logic. Indeed, these "intuitions" are used to critically evaluate logical systems, both syntactic proof theory (axiom and natural deduction systems) and semantic theories with their "truth-conditions" and definitions of valid argument. Here are some controversial validities of first-order logic:

1. $\models_{\mathbf{FOL}} P\vee{\sim}P$
2. $P, {\sim}P \models_{\mathbf{FOL}} Q$
3. $\models_{\mathbf{FOL}} \exists x(Fx\vee{\sim}Fx)$
4. $Fa \models_{\mathbf{FOL}} \exists xFx$
5. $\forall x(Gx\rightarrow Hx), \forall x(Fx\rightarrow Gx) \not\models_{\mathbf{FOL}} \exists x(Fx\wedge Hx)$

Select one or two of these that your intuitions suggest may in fact not be really valid, and discuss whether and why your intuitions should have any bearing on the acceptability of first-order logic as a "scientific theory."

iii. Intuitionistic Proof Theory as Semantics: Meaning as Use

Exercise 12*.* Intuitionistic logicians and, more generally, constructivist mathematicians reject classical semantics because of its use of excluded middle, *reductio* and non-constructive sets. Gentzen's natural deduction proof system (of introduction and elimination rules) has been proposed as an alternative "semantic" theory of "meaning" even though it is really a syntactic theory that is formulated completely in terms of the spatial arrangement of symbols on the page. Explain the argument for the thesis that this syntactic proof theory could be considered a theory of meaning, and offer some critical remarks (for or against) its success as a theory of meaning.

iv. Henkin's Proof of Soundness and Completeness

Exercise 13. In a short essay spell out for yourself in terms that you will be able to understand later (say, in twenty years):

1. what the soundness and completeness theorem says,
2. why it is interesting,
3. the general strategy of Henkin's proof, and

Chapter 3

Effective Process and Undecidability

I. CALCULATION, ALGORITHMS AND DECIDABLE SETS[45]

A. Introduction

It was logicians or philosophers with an interest in logic that first envisioned and actually built the earliest computers. Ancient Greek mathematicians had discovered simple algorithms for some basic calculation in arithmetic, like Euclid's algorithm for greatest common devisor (defined below). We learned similar procedures in elementary school, like the technique for long division. Such procedures turn a human being into a paper and pencil, flesh and blood calculating machine. Numbers are feed in, the procedure is applied by rote, and it terminates in a sort time with a set of final numbers. No thinking or judgment is required beyond careful rule following.

In the Middle Ages and again in the sixteenth century the philosophers extended such calculation beyond simple arithmetic. An intriguing but notoriously unsuccessful example was the calculation procedure proposed by the 14th-century Spanish philosopher Raymond Llull. Llull constructed various mechanical devices for predicting the future. These consisted of concentric disks of increasing diameter mounted on a central axis so that they could rotate one on top of anther. Mystical symbols representing forces governing the world are written around their outer edges. A row of symbols one from each disk is then created by their alignment one above the other along a "spoke" of the disk. These combinations of symbols had special meaning, expressing a proposition of some sort. By rotating the disks, a "proposition" along one spoke describing one state of affairs was automatically correlated with another on a different spoke. By this means its was believed the future could be predicted. Kings and queens, including Elizabeth I of England, who hired practitioners of this art for advice, esteemed the system.[46]

Though Llull's device was essentially superstitious, it inspired the seventeenth century German philosopher Gottfried Wilhelm Leibniz (1646-1716) to more serious work. What was important about Llull's invention was that it was a machine. It provided a purely mechanical manipulation of crude, highly visible physical objects, in a finite and prescribed procedure, one with clear beginning and ending points, and clear steps in between. In a series of essays Leibniz designed paper and pencil calculation procedures that likewise had clear

[45] I am indebted to Jennifer Seitzer for reading over sections 1-4 and for helping me in the correct usage of Computer Science.

[46] On Llull's machines see Martin Gardner, *Logic Machines, Diagrams, and Boolean Algebra* [1958] (N.Y.: Dover, 1968).

beginning and end points, and a finite number of simple and transparent intermediate steps. His procedures were directed to topics in logic, the validity of various syllogistic inferences. Leibniz' procedures were different in an important way from Aristotle's "reductions" of the valid syllogistic moods to Barbara and Celarent. Proofs, even Aristotle's reductions, are not automatic. They require ingenuity on the part of the prover. Leibniz' method, in contrast, required no thought or originality, and could be applied by a dunce. So long as the steps were correctly followed, it was sure to produce an answer in a short period of time.[47] Leibniz went further in his experiments with calculation and actually constructed a machine that would do addition.[48] At roughly the same time in France Blaise Pascal (1623-1662) constructed an even more elaborate machine for simple mathematical calculations. Though proposals to mechanize logic did not resurface until the twentieth century, experiments in arithmetical calculators continued. By the start of twentieth century large offices were equipped with extremely bulky devices that clerical staff used to add long strings of numbers. These evolved into the mechanical office adding machine. In recent decades, however, there has been a quantum leap in calculation with the invention of the computer, a device that depended on advances in theoretical logic.

B. The Concept of Calculation

You will recall that one of the major projects in the foundations of mathematics in the last century was to axiomatize arithmetic, and that in the thirties Kurt Gödel advanced the subject in several ways. He brought the class of simple arithmetical calculations like addition and multiplication within a larger class of numerical calculations in general – called the (primitive) recursive functions – and provided an inductive definition of this class.

The mathematician wants to understand this wider class of numerical calculations because he wants to understand the general properties of arithmetic. Philosophers too are interested in calculation, but not because they are specifically interested in arithmetic. What interests philosophers about addition, multiplication and other automatic calculating procedures is clarity of knowledge they provide. Philosophers have always been concerned to explain how we know anything – the subject is called epistemology. Knowledge that possesses certainty is particularly interesting. Over the centuries philosophers have become more and more skeptical of the existence of any certainties at all, but mathematical calculation remains one serious candidate. What is calculation and

[47] The mnemonic names for the valid moods given to them in the Middle Ages did allow for the automatic construction of a "proof", but they do so because the syllogistic is trivial. There are only a small finite number of valid moods. One could simply memorize all of them. Because it is finite, this list itself would count as a calculable function. It is only when the set of validities becomes infinite that the problem of defining a calculable decision function becomes interesting.

[48] See especially his paper *De arte combinatoria*, translated and explained in G.H.R.Parkinson, *Leibniz, Logical Papers* (1966: Clarendon Press, 1966). His techniques were designed to yield a proof for a true propsition in a finite number of steps and to terminate without a proof for a false proposition. Hence, in the later language of this section, they were algorithms intended to both produced proofs and to serve as a decision procedure.

what about it makes it certain, if in fact it is certain? These are philosopher's questions.

Both mathematicians and philosophers, then, want to know what calculation is. What is unclear as they start out is what sort of answer they are looking for. How is calculation to be explained? What would an explanation look like? How would we know if we had a good one?

Gödel provided an answer of a sort in his inductive definition of the (primitive) recursive functions. What is a calculation? His answer is that it is a member of **PRF**. It is the model of an inductive definition that Gödel used to "explain" calculation. There is a good deal to be said about this definition as an "explanation." It was the first of a series of attempts find an inductive accounts of calculable functions in the period from 1930-1950. These attempts, which were independent and based on quite different ideas, all ended up defining exactly the same set. Gödel's work was completed in 1931. In 1936 Alan Turing, an English logician, defined the notion of a ***Turing machine*** and a ***Turing computable function***. Turing's abstract machines are simple computers that when combined can compute any computable function. In 1947 the Russian logician A. A. Markov defined the notion of ***algorithm***, a set of directions for calculating an answer from given numerical inputs. Markov's algorithms are the abstract versions of computer programs, and they define the set of "program calculable" functions. These and other definitions are equivalent in a rigorous sense: it can be shown by mathematical proofs that a function is in one of these sets iff it is in another. In this case the Axiom of Extensionality in set theory says the two sets are identical. The coincidence of the different approaches suggests very strongly that there is a genuine phenomenon in the world that the different approaches have all captured with different ropes. This is the set of calculable functions.

It must be admitted, however, that there is something unsatisfactory about inductive definitions as "explanations," especially to a philosopher who wants to see into the "essence" of things. Recall that Socrates and Aristotle set the standard of philosophical insight when they required that explanations take the form of definitions. Aristotle insisted that a definition lay bare a species' nature or essence – understood as its genus restricted by its characteristic difference. In general philosophers still look askance at "definitions" that fail to list the necessary and sufficient conditions for a term's application.

Axioms are all right if they are the best explanation available. Such, for example, is the philosopher's typical attitude to set theory. Sets cannot be defined directly, so we must be satisfied with their characterization by a set of axioms, which are sometimes said to "implicitly" define the primitive terms they contain. In the case of set theory the main primitive term is that for set membership. Another example of a primitive term not directly definable is *time*. What is time? It would be most satisfying to have a definition. The best physics has to offer, however, are basic laws of nature in which the concept of time occur as primitive terms. (Consider Newton's laws f=ma, where a (acceleration) is d/t^2. Here a *time* variable appears as an undefined primitive.)

Inductive definitions have the same drawback as axioms. They tell you how to construct a set but they do not give you its “conceptual content.” They do not say what the objects in the set have in common. Perhaps some important sets just do not have common features, and the traditional quest for necessary and sufficient conditions is a misguided scientific enterprise. Maybe so. But an attempt at a straightforward traditional definition of a calculable function is revealing.

If this idea is to be defined, the definition must include as a minimum numerical calculations. In principle however there is no reason to limit such processes to numbers. It seems we could apply the same sort of mechanical procedure to other sorts of entities, like symbols or any manageable hunks of matter. An adding machine, for example, is entirely physical, as is a computer. The inputs and outputs are material objects, but the process is like calculations on numbers in that it works quickly and reliably. The formal name for the more general phenomenon not restricted to numbers is ***effective process***. This is the idea we want to define. Before attempting to do so, however, let us have some more examples of effective processes. These we need to guide our later abstraction.

We shall define a series of what are technically called ***algorithms***, a set of directions for an effective process. Computer programs generally fall in this class. Each has a definite starting point, and a finite series of rules in a set order. The process begins by applying the first rule that fits to the starting point. It process proceeds to a new step to it the first applicable. In this way the process produces more steps by proceeding down the rule list, skipping any rule that has a condition of application that does not fit the current step. If the procedure does not stop because a rule says to stop or because no rule is applicable, the procedure returns to the top of the rule list and starts applying them all again, in their prescribed order. If the definition has no “glitches,” the process will not go in circles, or run on forever. Rather, after completing a finite number of steps, there will be a step to which no rule is applicable or there will be a rule that says at that step the process is to “stop”.

Example 1. The Addition Operation.

We begin with a recursive definition of the *addition operation* + in terms of the successor operation ***S***:

$$x+0=x$$
$$x+\boldsymbol{S}(y)=\boldsymbol{S}(x+y)$$

We can use + to define an algorithm for calculating *n*+*m*, for any numbers *n* and *m*, on the assumption that we can apply the successor operation to any number. Accordingly we can write an algorithm to calculate *n*+*m* as follows:

Starting Point:

Write *m* imbedded *n* ***S***'s as follows:

$$\underbrace{\boldsymbol{S}(\boldsymbol{S}(\ldots(\boldsymbol{S}(\boldsymbol{S}(m)))))}_{n\ \text{S's}}$$

Step 1. If there is a numeral *c*, and it is inside the expression ***S***( ). Calculate ***S***(*c*) and rewrite the expression on the line beneath replacing ***S***(*c*) with the numeral for its successor. Go to Step 1.

Step 2. If there is a numeral *c*, and it is not inside the expression ***S***( ), stop.

The numeral you finish with is the name for the sum of *m*+*n*.

Example. 4+3

$$\overbrace{\boldsymbol{S}(\boldsymbol{S}(\boldsymbol{S}}^{3\ \boldsymbol{S}\text{'s}}(4)))$$
$$\boldsymbol{S}(\boldsymbol{S}(5))$$
$$\boldsymbol{S}(6)$$

Example 2. Euclid's Algorithm: the Greatest Common Divisor of m and n.

Starting point: write the numerals for any pair of natural numbers *m,n.*

Step 1. If *n*≤*m*, and n is on the left, move to the line below and write *m,n*.

Step 2. If *n* ≠ 0, divide *m* by *n* and move to the line below and write *n*,*r* where *r* is the remainder. Go to Step 1.

Step 3. if *n* = 0, move to the line below, write *m*, and stop.

Examples

```
3,17        8,12
17,3        12,8
3,2          8,4
2,1          4,0
1,0          4
  1
```

The number on the lowest line is the GCD of *m* and *n*.

Example 3. Formulas in Polish Notation.

Starting Point:

Write down a (non-empty) string of symbols.
Append the symbol *e* for the empty string at the right-hand end of the string.
Set the counter *c* so that *c*=1, and write that value for *c* beneath the string to the left.
Go to the leftmost sign in the string.

Step 1. If the sign is *P*,*Q*, or *R*, set counter to *c*=*c*-1, write the new value of c under the sign, and move to the sign on the right. Go to Step 1.

Step 2. If the sign is **N** move to the sign on the right. Go to Step 1.

Step 3. If the sign is **K**,**A**,**C**,**E**, set counter to *c*+1, write the new value of *c* under the sign, and move to the sign on the right. Go to Step 1.

Step 4. If the sign is other that *A*,*B*,*C*,**N**,**K**,**A**,**C**, or **E**, stop.

Step 5. If the sign is the empty string e, stop.

The procedure produces a final (rightmost) value for c. If *c*=0, the string is a wff in Polish Notation, and if **c**≠0, it is not. (It is the possibility of such simple algorithms that makes Polish Notation a favorite of computer programmers.)

<u>*Examples*</u>

```
 CANNABKBNAe     NAEANBKCAe     NAEANBKCKe
1 2 3   2 121  0    1   232  1210     1  2 32  1 21
```

Example 4. A Test for Tautologies, Contradictions, and Valid Arguments (with a Finite Number of Premises) in Sentence Logic

Starting point. Convert the item to be tested into a single wff *P.* (An argument $P_1,...,P_n$ to Q is converted into the conditional $(P_1\wedge...\wedge P_n)\rightarrow$Q.) Next determine the 2^n possible assignments of T or F to the atomic sentences of *P*. Put these assignments into some fixed order, and write in order a copy of the grammatical tree of *P*, one for each assignment. In each tree writing next to each atom in the tree the truth-value it has in that assignment. Move to the leftmost tree.

Step 1. If some node in the tree has no value assigned to it, find the most north, then west (highest, then leftmost) node whose predecessor nodes all have truth-values assigned to them. Assign to this node the truth-value determined by the truth-table for the connective introduced at that node. Go to Step 1.

Step 2. If every node in the tree has no value assigned to it and there is a tree to the right, go to that tree and then go to step 1.

Step 3. If every node in the tree has no value assigned to it and there is no tree to the right, stop.

If at the end the roots of all the trees are assigned T, the sentence is a tautology. If they are all F, it is a contradiction. If some are T and some are F, it is neither.

<u>*Example*</u>. Testing $P\vee\sim P$

There is one atomic sentence *P* in $P\vee\sim P$ and therefore 2^1=2 assignments of truth-values: *P* is T of *P* is F. There are accordingly two trees:

```
                P,T                              P,F
                 |                                |
    P,T        ~P,F                P,F          ~P,T
      \         /                    \           /
       P∨~P,T                         P∨~P,T
```

The root of all trees is marked with T. Therefore the sentence is a tautology .

Given these examples, it is possible to abstract the general features characteristic of calculable functions in general. First of all there is a clear entity with which the process starts (the "starting point"). There are a finite number of manipulations or steps, commencing with the starting point, such that it is clear at each step what entity ("input") is given at the beginning of the step, what it is that must be done to it during the step, and what entity ("output") results from applying the step. It is clear when the finite process terminates and what entity is finally produced (the "end point"). What is especially interesting to philosophers is the clarity at each stage. It is possible to *know* at each point what is going on, and to know it with what approaches certainty.

Definition 1-1

1. An entity is ***epistemically transparent*** iff it is possible to tell with a high degree of certainty what it is.
2. An application of a rule (a step) is ***epistemically transparent*** iff it starts with an epistemically transparent entity, it proceeds by associating with it a second epistemically transparent entity and it is possible to tell with a high degree of certainty that the entity associated with it is the correct one.

Symbols as marks on a page or even as spoken sounds – whether they be symbols logic, mathematics, or ordinary language – are epistemically transparent in this sense because we can tell what they are with a high degree of certainty by simple sense perception. We just look at them. If they are the right size, neatly written or printed, the lighting is right, and our vision is normal, we can be as certain about what written symbol is in front of us as we are about almost anything else. Likewise the manipulations of symbols-on-a-page typical of logic and mathematics are epistemically transparent. They consist of adding and subtracting symbols and of moving symbols about on a page.

It is to obtain this epistemic transparency that syntax limits itself to talking about just signs and their properties. The strings and properties in question are supposed to be transparent in this sense. Signs accordingly are limited to a set of highly perceptible crude physical objects. In logic and mathematics these are marks on paper. In linguistics studies of the syntax of natural languages, they are limited to certain perceptibly different sounds. Similarly, the processes allowed in syntax are crudely physical manipulations of "signs." The adding, subtracting, and moving of written symbols in logic and mathematics is a physical process the correct application of which is immediately evident.

We are now ready to summarize the discussion with an attempt at a philosopher's traditional definition, in terms of necessary and sufficient conditions, of an *effective process*.

A Traditional Definition ("Analysis") of Effective Process

A process is ***effective*** iff it consists of a finite series of elements such that there is some epistemically transparent element *e* and some finite list of epistemically transparent rules such that the *e* is the first element of the series each subsequent element is produced by the ordered application of the first applicable rule.

This sort of definition, if successful, would have the virtue that it sees into what is common to all elements of the class it defines. But traditional definitions only work if they succeed at illuminating the concept. They cannot do so if the terms that are used in the definition are themselves not well understood: *you can't explain the obscure by the obscure*. The definition above, however, suffers from exactly this defect. It explains the idea of effective process in terms of other ideas, including the philosophically loaded idea of epistemic transparency. What is it to know something with certainty? To give a deeper explanation, we would have to know what knowledge and certainty are, and these are deep still unresolved issues in epistemology. Hence, the direct philosophical analysis of effective process is not of much use or interest outside philosophy. It is for this reason that the alternative definition of effective process found in mathematics and logic is preferred in mathematics and logic circles. The inductive definitions of Gödel, Turing, Markov and others do not lay bare the "essence" of the notion. They do, however, explain how to construct the set. Moreover, as we saw in set theory, the whole idea of a direct definition in terms of necessary and sufficient conditions may well be problematical. Indeed, every abstract science, philosophy included, might do well to abandon direct traditional definitions in favor of indirect constructions. It might be said that the reason epistemology has been unsuccessful for 2,000 years is due to its fixation on Aristotelian definitions.

On the other hand, it must be admitted that the omission of any reference to certainty or knowledge misses in an important way what effective processes are all about. The definitions of recursive function, Turing machine, Markov algorithm all fail to mention knowledge and certainty. But it is in large part because effective process as there defined is the abstract version of the intuitive epistemically transparent notion that they are theoretically interesting. We employ calculation and computers generally precisely because they are quick and dependable. What kind of "science" is it that would explain such things without addressing their most important feature?

There is a middle ground. It is to recognize that the same idea may be approached in different ways. Philosophers may attempt direct definitions of effective process, admitting that their best definitions suffer from defining terms that remain deeply unclear. Logicians and mathematicians can likewise offer their inductive definitions of effective process recognizing that they are offering mere constructions that fail to mention what is perhaps the most significant feature of effective processes, their epistemic transparency. The constructive definitions, nevertheless, are mathematically precise and appear to capture the right set of operations. That the two approaches to effective process, the philosopher's and the mathematicians, are in fact explaining the same phenomenon is not something that can be proved in mathematics itself. It consists of a determination that the set of effective processes as defined by philosophers in epistemic terms is in fact co-extensive with the set of effective processes defined inductively in recursion theory. Whether the two sets do in fact coincide is, however, a super-mathematical question involving non-mathematical ideas. The thesis that the two coincide falls in the philosophy of logic, and it is known as *Church's thesis*, after the American logician Alonzo Church. After advancing his own inductive definition of effective process in terms of the lambda calculus – provably equivalent to those of Gödel, Turing, Markov, *et al*. – it was Church who first clearly stated the conceptual link between the mathematician's inductive definition and the philosopher's epistemic concept.

Church's Thesis

The set of effective processes that is defined by induction by Gödel (and others) is in fact the same set as the set of effective processes defined by the direct definition in terms of knowledge and certainty.

Among logicians and philosophers there is a consensus that the thesis is true. Everyone admits that the operations captured by the inductive definitions are effective in the epistemic sense because all the individual basic operations and the rules of construction used in the inductive definition clearly have the required epistemic transparency. It is the converse direction that may be questioned. Might there be some effective process that is not captured by Gödel's inductive definition? The evidence against this possibility, and it is strong evidence indeed, is the amazing convergence of the various attempts to give an inductive definition of effective process. The fact that the definitions of Gödel, Turing, Markov, Church, and others use quite different sets of ideas but all provably pick out the same set suggest that they have captured the real animal.

A decision procedure is an effective process that determines whether an entity is a member of a set. Given any entity the procedure provides an answer "yes, it is in the set" or "no, it is not in the set." It is what is called the *characteristic function* of the set. If **C** is the set in question, then the procedure is a function *f* such that *f*(*x*) is 1 if *x* is in **C** and is 0 if *x* is not in **C**. Moreover, given any *x*, it produces its unique judgment in a finite number of perfectly prescribed and epistemically transparent steps.

Definition 3-2

The ***characteristic function*** of a set **C** is a function *f* defined for any value of *x* such that *f*(*x*)=1 if *x*∈**C** and *f*(*x*)=0 if *x*∉**C**.

Definition 3-3

A ***decision procedure*** for **C** is any characteristic function of **C** that is an effective process.

A set **C** is said to be ***decidable*** iff there is a decision procedure for **C**.

Let us finish our discussion of constructive (inductively defined) and decidable sets by noting that they are not the same thing. There are some constructive sets that are not decidable, and there are some decidable sets that are not constructive.

An example of the former is the set **Th$_{\textbf{FOL}}$** of theorems of first-order (predicate) logic. This set is inductively defined. It is the closure or the axioms 1-6 of the axiom system **PM** under the rule modus ponens. Let us call the axiom system based on these axioms **FOL**. This set of theorems is non-decidable, and the proof of that fact is a major result to which we shall turn shortly. The fact that it is undecidable, however, is exceedingly interesting. It shows that even though it may in principle be possible to construct a set, we may not be able to tell for a given element whether the construction process puts it into the set. This is a limitation on human knowledge. Certainty does not extend to the membership of all constructive sets.

Similarly, there are some sets for which we have a decision procedure, but which we cannot construct. The set of prime numbers is a good example. It is easy to test whether a number *n* is a prime. Merely start dividing all numbers less than *n* into *n*. If any of them (other than *n* and 1) divide *n* without a remainder then *n* is not a prime; otherwise it is. Nobody, however, knows how to construct the primes. If we did then it would always be possible to construct the next prime. Finding higher primes, however, is a hit and miss proposition in modern mathematics, and the discovery of a new prime is a major and unpredictable event.

Examples

	Constructible	Decidable
Th$_{\textbf{FOL}}$ (Theorems of First-Order Logic)	yes	no
The Prime Numbers	no	yes

C. Logic and Artificial Intelligence

One way to understand a human being is to cut him open and look, and then try to describe and explain what one sees. Another was is to “model” him, to construct something that behaves just like he does. A perfect robot, in this second approach, would shed light on how humans work. Such at any rate is one strategy of the discipline called ***artificial intelligence***. It tries to “understand” human thinking by constructing machines (computers) that reproduce the reasoning processes of humans. Any such approach is open to the objection that merely reproducing the same behavior does not guarantee that it is being produced by the same mechanism. Both birds and airplanes fly, but there are differences in how they do it. Despite such obvious problems artificial intelligence plows on. Even if it does not ultimately shed light on how humans work, it would be quite remarkable if it succeeds in making machines that replicate human reasoning. With any luck we could then edit out the errors human tends to make, and end up with a machine that thinks better than we do. We could put it charge of things, like train systems, the economy, and teenagers.

In one sense logicians have been trying to achieve this end. Logicians are not psychologists. They do not look at how people reason. They do not collect questionnaires interviewing thousands of subjects on whether *modus ponens* is right. That is, logic is not a descriptive science. Rather logic is interested in what really follows from a premise, regardless of whatever erroneous judgments humans are apt to make. In this sense logic is often said to be normative or prescriptive. This terminology is slightly misleading because it might suggest that logical rules are a matter of values or precepts laid down by some authority or sage. On the contrary, the facts of logical validity are facts of nature, just as much as those of arithmetic. We discover and elucidate them by the techniques of logic, just as mathematicians discover and explain mathematical facts. Of course, a given logical system, like an axiomatization of geometry, may not accurately describe the world we live in. Like a geometry, a logical system must be subjected to empirical verification. The verification, however, will concern what facts follow from what, not what facts human beings think follow. In a similar manner a geometry is verified by taking physical measurements, rather than by interviewing humans about their personal views on geometry, errors and all.

Once the “right” logic has been discovered, moralists may enter the picture. They can urge humans to be logical in the sense of using the rules of the right logic when they do their personal reasoning. In a similar way somebody having a bridge built might urge the engineers to use the right geometry when they make their calculations. But whether you or I use *modus ponens*, or the engineer building the bridge uses the Reimannian parallel postulate, reflects not at all on the validity of the first and the truth of the second.

Logicians then have a rather aloof stance towards psychology. Psychologists, on the other hand, have a very difficult job. The psyche does not lead itself easily to scientific explanation. This is especially true of the process of human reasoning. In their desperation for ideas psychologists have been know to turn to logic. Perhaps they hypothesize, humans do in fact reason logically, or at least approximate logical reasoning. After all if they didn't, wouldn't they have been eliminated long ago in the struggles of natural selection? If so, then logic may provide models for human psychology. Such in general is the idea behind artificial intelligence.

In artificial intelligence investigators have attempted to model in computers or other machines all sorts of human mental powers, from walking across the irregular terrain's, to speech recognition, to visual discrimination. One area that draws heavily from traditional logic is that of experts systems. An expert knows her field. She makes judgments drawing on her experience and theoretical background. The challenge is to make a machine that replicates the judgments of this expert.

The proposed way to do so is a straightforward application of traditional logic. First you axiomatize the theory. You produce an axiom system. There are two parts to your axiomatization. First you collect the general laws and principles of the subject matter you are thinking about. You translate these into sentences in logical notation. These are called the ***laws*** of the system. Next you collect all the relevant details of fact. These too are represented in logical notation, each fact by its own sentence. These are called the ***database*** of the system, and they are added to the general rules to make a large axiom system called a ***logic program*** that spells out all the assumptions "the expert" will be using in the reasoning process. The expert's "knowledge" is then represented as the set of conclusions (theorems) that can be deduced logically from the laws and database.

An expert's judgment on a particular case is more complicated. It is represent by a determination on the "truth" of a sentence given the axioms. It is judged to be true if it follows and false otherwise. To capture this additional feature of *judgment* on the truth of given sentence, the system must go beyond a usual axiom system. An axiom system alone, though constructive, does not provide a decision procedure allowing one to tell whether a sentence is a theorem of the system. If the set of theorems is in fact decidable, then there is a decision procedure that can be used to make such judgments. Suppose a set $\mathbf{Th}_{\boldsymbol{S}}$ of theorems of an axiom system $\boldsymbol{S}$ has an effectively decidable characteristic function $\boldsymbol{f_S}$. The we can decide if a sentence Q follows form a finite program $\{P_1,...P_n\}$ if $\boldsymbol{f_S}(P_1,...,P_n\rightarrow Q)=1$. Accordingly, we restrict programs to finite sets and expert systems to sets of theorems that are decidable. To represent the expert's judgment on particular cases, we augment an axiom system with a decision procedure for its set of theorems. In computer jargon the decision procedure is called the a ***query mechanism*** or ***handler,*** i.e. a kind of hidden meta-program built into the computer language that "runs" individual programs. A sentence that is submitted to the decision procedure for a judgment as to whether it is "true in the program" is called a ***query***.

Definition 3-4

An ***expert system*** in language $\mathbf{L}_S$ consists of the following:

1. an axiom system ***S*** in $\mathbf{L}_S$ consisting of
   a. a set $\mathbf{Ax}_S$ of axioms divided into two parts:[49]
      i. a set of molecular sentences, called the system's ***program***
      ii. a set of atomic sentences, called the system's ***database***
   b. a set $\mathbf{R}_S$ of rules of inference,
   c. the set $\mathbf{Th}_S$ of theorems defined as the closure of $\mathbf{Ax}_S$ under the rules in $\mathbf{R}_S$
2. a decision procedure $\boldsymbol{f}_S$ for $\mathbf{Th}_S$, called the system's ***query handler***.

Then, a ***query*** is defined as a sentence from $\mathbf{L}_S$.

We have already met several axioms systems. Some of these will be suitable for expert systems, namely those that are decidable. Let us first review the axiom systems we have encountered. We axiomatize sentence logic (using axioms 1-3 of ***PM***), and for its set of theorems $\mathbf{Th}_{SL}$ we actually defined a decision procedure earlier in this chapter. We went on in ***FOL*** to axiomatize all of first-order logic (axioms 1-6 of ***PM***), but this system is (as we shall prove) undecidable – there is no decision procedure for $\mathbf{Th}_{FOL}$. In addition, we axiomatized all of set theory in the full axiom set (axioms 1-10) of ***PM***, but this system is also undecidable since it includes as a subset the set of theorems of predicate logic, which is undecidable. Finally, we learned from Gödel that arithmetic is not even axiomatizable. Let us summarize this information.

[49] A logic program is sometimes called an *intensional database* (IDB), and the database is then distinguished by being called an *extensional database* (EDB). This bit of jargon is unfortunate in that it has very little in common with more traditional uses of *intension* and *extension* in logic or philosophy

Systems	Definition	Set of Theorems	Decidable
Let ***P*** be a set of formula of the language (a *program*)			
SL Sentence Logic consequences of ***P***	***P*** and Axioms 1-3 closed under *modus ponens*	$\mathbf{Th}_{P\cup SL}$	yes
FOL Predicate Logic consequences of ***P***	***P*** and Axioms 1-6 closed under *modus ponens*	$\mathbf{Th}_{P\cup FOL}$	no
PM . Type Theory consequences of ***P***	***P*** and Axioms 1-8, 9*,10* closed under *modus pones*	$\mathbf{Th}_{P\cup PM}$	no

As this table indicates, the only "logics" we have met that would be suitable for genuine expert systems is sentence logic because only it is decidable. Any finite set $\{P_1,...,P_n\}$ of ***SL***-sentences could serve as a program and any individual sentence Q as a query. The truth-table decision method (defined above) could then be used to test the conditional $(P_1\wedge...\wedge P_n)\rightarrow Q$.

Example. An **Expert System $\mathbf{E}_{SL}$ Based on Sentence Logic**, for sentences in language $\mathbf{L}_{SL}$ of sentence logic, consists of:

1. An axiom system made up of:
    - a. finite set of sentences $\mathbf{P}_{E\text{-}SL}$ of $\mathbf{L}_{S}$ which serve the role of premises of the system and are called the ***program*** of $\mathbf{E}_{SL}$, such that:
        - i. the axioms of sentence logic $\mathbf{Ax}_{SL}$ are all included in $\mathbf{P}_{E\text{-}L}$ (i.e. $\mathbf{Ax}_{SL}\subseteq\mathbf{P}_{E\text{-}SL}$), and
        - ii. $\mathbf{P}_{E\text{-}SL}$ divided into two parts:
          a set of molecular sentences, called ***laws***,
          a set of atomic sentences, called the ***database***,
    - b. a set $\mathbf{R}_{E\text{-}SL}$ of rules of inference containing just *modus pones*,
    - c. the set $\mathbf{Th}_{E\text{-}SL}$ of theorems constructed from $\mathbf{P}_{E\text{-}SL}$ and $\mathbf{R}_{E\text{-}SL}$,
2. the decision procedure $\boldsymbol{f}_{E\text{-}SL}$ defined earlier for $\mathbf{Th}_{E\text{-}SL}$ (i.e. for tautologies in sentence logic).
3. a ***query*** is any sentence form $\mathbf{L}_{SL}$.

First-order (predicate) logic is undecidable. However, many laws and arguments that cannot be expressed in sentence logic can be expressed in predicate logic. It would be highly desirable to somehow extend decision procedures to embrace major parts of predicate logic even if we could not decide on all its theorems. Are there subsets of predicate logic theorems that are decidable? This is a major question for computer science, and its various partial and qualified answers make up a good part of the literature on expert systems.

One important sublangauge of first-order logic that is decidable is so-called *monadic quantification theory*, the set of all first order formulas constructed from atomic formulas that contain at most one variable. A decision procedure that essentially consists of the truth-table test for validty can be defined for this fragment.[50]

Another important approximation of a decision procedure for predicate logic is incorporated in the computer language **Prolog**, designed for writing expert systems. By limiting the syntax of its programs and queries a partial decision procedure is possible. Prolog is practical and commercially successful. It is also an excellent example of how AI programming languages are really branches of symbolic logic in disguise. In what remains of this lecture I shall explore Prolog to illustrate the details of how logic is built into an important language for expert systems.

A program in Prolog is simple to write for somebody familiar with symbolic logic. It is just a set of quantified formulas of the right syntax. A program is "run" by proposing a query. Then the resolution decision procedure hidden in the query handler of the computer language takes over to test whether the sentence proposed in the query in fact follows logically from the program. The general strategy is to assume the opposite of the query in question and then attempt to deduce a contradiction by an inference rule called *resolution*. If a contradiction follows the proposition queried is true. If no contradiction is reached the proposition is generally, but not always false. The syntax of Prolog is limited to facilitate and increase the likelihood that such a strategy will work. Laws are limited to universally quantified conditionals of a single variable with antecedents that are either atomic or negations of atomic sentences, and the data base is limited to atomic sentences. Queries are limited to atomic sentences. It is generally possible to test propostions by using just the two logical rules universal instantiation and *modus tollens*. In most cases the deduction procedure either terminates in a contradiction, in which case the sentence queried is true ("in the world of the program") or the procedure terminates without a contradiction, in which case the queried sentence is false ("in the world of the program"). The decision method of Prolog is imperfect, however, and not a genuine decision procedure, because sometimes it reaches no result, because it runs in circles or in infinite chains. Thus, Prolog is really just a practical approximation of a genuine expert system. It is more useful than a system in sentence logic, however, because its syntax is richer and allows for the formulation of some complex reasoning patterns that depend on quantifiers.

[50] See W.V.O. Quine, *Methods of Logic* (N.Y.: Holt, Rinehart, and Winston, 1950), p. 192 *ff.*

D. Systems in the Language *Prolog*

The language Prolog works by building into it query handler, which is hidden from the view of the user. This is a program that is a quasi--theorem tester, an algorithm that, more or less well, tests to see whether a formula follows from a set of premises in first-order logic. In this section we will see how Prolog works. Here Prolog notation and procedures will be explained in the notation and terminology associated with the literature on Prolog. At the same time what is really going on in the logic will be explained in the notation and concepts of symbolic logic. To keep the two straight, we shall present in the text the ideas in Prolog using its regular terminology. As we go along, we place in large displayed shaded boxes the reformulations into logic.

By a ***literal*** is meant any atomic formula or its negations. We shall use the letters *L* and *M,* and later *P* and *R* as well*,* to range over literals . If a formula does not contain free variables it is said to be ***grounded***. Grounded literals will be very important to Prolog, especially as they occur in conjunctions and disjunctions. Since first-order logic obeys the rules of association and commutation for conjunction and disjunction, the order and arrangement of pure conjunctions and pure disjunctions is irrelevant to their truth. Thus to simplify matters we shall assume that all pure conjunctions and disjunctions of literals are written in one standard form. (For example, we may arrange them as follows: write first from left to right the non-negated formulas in alphabetic order, followed by the negated literals again in alphabetic order, all grouped by increasingly larger nestings from left to right.)

i. Programs as Axioms

A ***(generalized) Prolog program statement*** is any disjunction of literals.[51] Any non-trivial (non-tautologous) statement $L_1\vee...\vee L_n$ in which there is more than one disjunct is called a ***law*** or, more commonly, a ***rule***. Often the positive and negative literals are separated as in $M_1\vee...\vee M_m\vee\sim L_1\vee...\vee\sim L_n$. This disjunction in turn is logically equivalent to the conditional $(L_1\wedge...\wedge L_n)\rightarrow(M_1\vee...\vee M_m)$ which has more the form of a traditional "law:' If one thing happens, then so does another. Often in the literature on Prolog laws are written as conditionals but with the arrow in the reverse direction $(M_1\vee...\vee M_m)\leftarrow(L_1\wedge...\wedge L_n)$.

A ***first-order logic law*** or ***rule*** is any $\forall v_1...\forall v_n(L_1\vee...\vee L_n)[v_1,...,v_n]$, where $v_1,...,v_n$ are all the free variables in $L_1\vee...\vee L_n$. That is, a law is a universal generalization of a disjunction of literals. A set $\mathrm{L}_{\mathbf{FOL}}$ of rules is called a ***first-order law set***.

Any statement *L* consisting of a single literal, i.e. a disjunction of literals containing only a single disjunct, is called a ***datum*** or, more commonly, a ***fact***.

Definition 3-5

A ***first-order logic datum*** or ***fact*** is any universal closure of a literal: i.e. any $\forall v_1...\forall v_n L\ [v_1,...,v_n]$, where $v_1,...,v_n$ are all the free variables in *L*. A set $\mathrm{D}_{\mathbf{FOL}}$ of such data is called a ***first-order database***.

In this discussion I shall use *law* instead of *rule* and *datum* instead of *fact* in order to emphasize that a Prolog expert system is a straightforward implementation of a covering law model of "explanation" of the sort common in the philosophy of science.

Note that in Prolog a datum is sometimes written $L\leftarrow$ (This unorthodox notation derives from reading a single literal *L* as a "conditional with an empty antecedent:" i.e. as some $P\rightarrow L$ (equivalent to $\sim P\vee L$) in which the *P* is "empty". Thus $\leftarrow L$, or equivalently $L\rightarrow$, really means $\sim L$.)

A ***Prolog program*** **P** is the union of some law set L with some data base D: P=L∪D

Definition 3-6

A ***first-order logic program*** is any $\mathrm{P}_{\mathbf{FOL}}=\mathrm{L}_{\mathbf{FOL}}\cup\mathrm{D}_{\mathbf{FOL}}$ for some first-order law set $\mathrm{L}_{\mathbf{FOL}}$ and some first-order data base $\mathrm{D}_{\mathbf{FOL}}$.

[51] Though possible in principle, Prolog languages that have actually been implemented do not permit disjuctive statements with negative litterals. Likewise, in practice consequents of rules are limited to single literals. The generalized forms here are designed to show the more general power that is theoretically possible for languages based on the ideas underlying actual Prolog that have been implemented to run on computers.

A ***query*** is any conjunction ($M_1\wedge...\wedge M_m$) of literals. It is a question in the sense that running the program for the query seeks a "yes" or "no" answer to the query, "Is ($M_1\wedge...\wedge M_m$) true, for some values of its variables, given the laws and the database of the program?" Logically a query amounts to the question, "Is ($M_1\wedge...\wedge M_m$) a theorem (for some value of its variables) in the inductive set with the program P =R∪D as axioms?"

> Viewed from the perspective of first-order logic a ***query*** is the question, "Is $\exists v_1...\exists v_n(M_1\wedge...\wedge M_m)[v_1,...,v_n]$ provable as a theorem in the axiom system that has $P_{\mathbf{FOL}}$ as its axioms".

ii. The Resolution Inference Rule

We shall now use logic to test whether certain formulas called "goals" follow from the "program." One of the advantages of the regimentation programs and (as we shall see) goals of Prolog into a restricted syntax is that the restriction permits the "logic" of the inference procedure to be very simple, employing only one rule, called ***Resolution***. The rule in sentential logic is easy to verify by truth tables. Here and later since we will always be dealing with the same set of premises, viz. the program that states rules and facts, and since we shall not be using discharge rules that shrink the premise set or thinning that enlarges it, we need not mention the premise set in stating logical rules or in describing the nodes on a proof tree. Thus we may abbriviate **P**├*P* to just *P.* We may then state the rule in proof tree notation as[52]:

Sentent*i*al Resolution $$\frac{P\vee Q \qquad\qquad R\vee{\sim}Q}{P\vee R}$$

It is important to pause for a minute to understand what the rule says. It says that if we know two disjunctive facts, which have parts that contradict each other, we can ignore their parts. They do not bear on the truth of what we know. What is important about this rule is that it provides a way to simplify what we know, to recast it in its essentials. We shall see that in some cases by repeated applications of the rule we may derive two nodes on a proof tree that jintly descend to a contradiction. We represent the contradiction by the ***contradiction symbol*** ⊥. It is does not matter what contradiction ⊥ represents so long as it is understood that in all interpretations $\mathfrak{I}$, $\mathfrak{I}$ assigns false to ⊥. Then resolution for this case reads as follows.

Sentential Resolution $$\frac{P \qquad\qquad {\sim}P}{\bot}$$

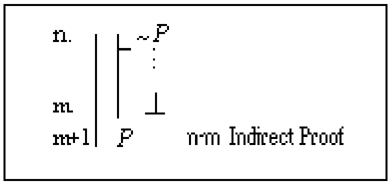
We shall also use the contradiction symbol in our statement of the inference rule ***indirect proof***. This is a version of reduction to the absurd in

[52] In full notation the rule would read: if X├$P\vee Q$ and X├$R\vee{\sim}Q$ then X├$P\vee R$.

which we prove Q indirectly by showing its negation is absurd. We assume $\sim Q$ temporally. If $\sim Q$ together with our original premises $P_1,\ldots,P_n$ lead to a contradiction, which represent by $\perp$, then $\sim Q$ is false and Q is true. (It is this last step, deriving from Q from the falsity of $\sim Q$, that is rejected in intuitionistic logic.) In the notation of natural deduction theory, the general form of indirect proof is:

$$\text{If } \{ P_1,\ldots,P_n, \sim Q \} \vdash \perp \text{ then If } \{ P_1,\ldots,P_n \} \vdash Q$$

In the discussion below we shall use the following informal format for laying out proofs using reduction to the absurd:



Here $\sim P$ is adopted as a "temporary premise" on line n. If it entails a contradiction on line m, from the premise on line n together with whatever has been assumed prior to line n, then P follows on line m+1 by indirect proof from the prior premises alone.

In Prolog resolution (this sort of *reductio*) is used together with rules for the quantifiers and variables. Since the only quantifiers Prolog uses in its derivations are universal, they may be instantiated for free variables that literally represent everything in the universe. By progressive applications of Universal Generalization and Instantiation, these variables may in turn be replaced by any other term, whether constant, variable or composite of a functor with other terms. Often variables from different formulas may be resolved into (instantiates to) the same term, in which case they are said to be ***unified***, and resolution is said to be ***uniform***.

Example. The resolution of unification $Fx\vee Gx$ and $Hy\vee\sim Gy$ into $Ft\vee Ht$:

$$\frac{Fx\vee Gx \qquad\qquad Hy\vee\sim Gy}{Ft\vee Ht}$$

This is accomplished by unifying x and y into t through steps of Resolution, and Universal Generalization and Instantiation as follows:

1.	$Fx\vee Gx$	Premise
2.	$Hy\vee\sim Gy$	Premise
3.	$(\forall x)(Fx\vee Gx)$	1 Universal Generalization
4.	$(\forall y)(Hy\vee\sim Gy)$	2 Universal Generalization
5.	$Ft\vee Gt$	3 Universal Instantiation
6.	$Ht\vee\sim Gt$	4 Universal Instantiation
7.	$Ft\vee Ht$	5,6 Resolution, x and y unified as t

Since Prolog only deals with general free variables, it employs a more general version of resolution, one that allows for a short cut that deletes intermediate steps:

Uniform Resolution $$\frac{P[v_1,...,v_n]\vee L[v_1,...,v_m] \qquad R[w_1,...,w_n]\vee \sim L[w_1,...,w_m]}{P[t_1,...,t_n]\vee R[t_1,...,t_n]}$$

In the rule the notation $P[t_1,...,t_n]$ is the result of replacing all occurrences of v_i in $P[v_1,...,v_n]$ by free t_i. That is, t_i replace v_i only if none of the occurrences of t_i nor of any variables in t_i become bound in $P[v_1,...,v_n]$.[53]

In the "degenerate case" of resolution in which the premises $P[v_1,...,v_n]$ and $R[w_1,...,w_n]$ are both empty, the two premises above the line are $L[v_1,...,v_n]$ and $\sim L[w_1,...,w_n\}$, and these are contradictory. In that case we understand the rule to allow us to deduce the contradiction symbol $\perp$:"[54]

$$\frac{L[v_1,...,v_n] \qquad \sim L[w_1,...,w_n]}{\perp}$$

iii. Running a Prolog Program: Deduction within an Axiom System

By a ***Prolog Axiom System*** let us mean any logistic system constructed from an axiom set P=L∪D composed of some set L Prolog laws and some Prolog D by the Resolution rule.

Definition 3-7

In first-order logic we preserve the full logical form of laws and goals, and accordingly require a few more rules of logic. As above $P[t_1,...,t_n]$ is the result of replacing all free occurrences of v_i in $P[v_1,...,v_n]$ by occurrences free for t_i and its contained variables (if any). In addition, $\perp$ stands for any contradiction such that for all interpretations $\Im$, $\Im$ of $\perp$ is F. We assume in each of the rules that disjunctions and conjunctions are written in the preferred order, and in the rule quantifier negation that all occurrences of double negations are eliminated.

Universal Instantiation $$\frac{\forall v_1...\forall v_n P[v_1,...,v_n]}{P[t_1,...,t_n]}$$

Quantifier Negation $$\frac{\sim\exists v_1...\exists v_n(P_1\wedge...\wedge P_m)\ [v_1,...,v_n]}{\forall v_1...\forall v)(\sim P_1\vee...\vee\sim P_m)\ [v_1,...,v_n]}$$

Indirect Proof If $\dfrac{P_1,...,P_n,\sim Q}{\perp}$ then $\dfrac{P_1,...,P_n}{Q}$

Uniform Resolution $$\frac{P[v_1,...,v_n]\vee L[v_1,...,v_n] \qquad R[w_1,...,w_n]\vee\sim L[w_1,...,w_n]}{P[t_1,...,t_n]\vee R[t_1,...,t_n]}$$

$$\frac{L[v_1,...,v_n] \qquad \sim L[w_1,...,w_n]}{\perp}$$

Definition 3-8

Then ***a first-order Prolog axiom system*** is any logistic system with an axiom set $P_{\mathbf{FOL}}=L_{\mathbf{FOL}}\cup D_{\mathbf{FOL}}$ and the four rules above.

[53] In the more precise notation of natural deduction from Pat II, the rule would be written: If $X\vdash P[v_1,...,v_n]\vee L[v_1,...,v_m]$ and $Y\vdash R[w_1,...,w_n]\vee\sim L[w_1,...,w_m]$, then $X,Y\vdash P[t_1,...,t_n]\vee R[t_1,...,t_n]$.

[54] More precisely, the rule reads: If $X\vdash L[v_1,...,v_n]$ and $Y\vdash\sim L[w_1,...,w_n]$, then $X,Y\vdash\perp$.

A ***query*** is any question: Is ($M_1 \wedge ... \wedge M_m$) true, for some values of its variables, given the laws and the database of the program? Logically a query becomes the question: Does ($M_1 \wedge ... \wedge M_m$) follow by resolution from a program P =R∪D?

Definition 3-9

In first-order logic the question may be posed as follows. In first-order logic whether $M_1 \wedge ... \wedge M_m$ follows from $\wedge \mathrm{P}_{\mathbf{FOL}}$ is then a question of whether $\exists v_1 ... \exists v_n(M_1 \wedge ... \wedge M_m)$ is provable in the axiom system that has $\mathrm{P}_{\mathbf{FOL}}$ as its axioms. That is, ***a first-order query*** is the question whether $\vdash_{\mathbf{P}_{\mathbf{FOL}}} \exists v_1 ... \exists v_n(M_1 \wedge ... \wedge M_m)$. Equivalently, since $\mathrm{P}_{\mathbf{FOL}}$ is finite, the issue is equivalent to an issue in first-order logic, namely whether $\mathrm{P}_{\mathbf{FOL}} \vdash_{\mathbf{FOL}} \exists v_1 ... \exists v_n(M_1 \wedge ... \wedge M_m)$.

Moreover, since $\mathrm{P}_{\mathbf{FOL}}$ is finite, we may view it as a long conjunction, which we may write $\wedge \mathrm{P}_{\mathbf{FOL}}$. The issue then becomes whether
$\vdash_{\mathbf{FOL}} \wedge \mathrm{P}_{\mathbf{FOL}} \rightarrow \exists v_1 ... \exists v_n(M_1 \wedge ... \wedge M_m)$.

The strategy in a Prolog theorem test is essentially indirect proof. The negation of the query is assumed and then it is taken as a goal of the procedure to refute this negation by indirect proof. A ***goal*** in Prolog is therefore the negation $\sim(M_1 \wedge ... \wedge M_m)$ of a query, i.e. the negation of any conjunction $M_1 \wedge ... \wedge M_m$ of literals.

In the literature on Prolog a goal is often written $\leftarrow(M_1 \wedge ... \wedge M_m)$. Again this notation comes from logic and the conceit that every statement in Prolog is some sort of conditional. Then a negated literal $\sim L$ is viewed as a "conditional with an empty consequent: i.e. as some $(M_1 \wedge ... \wedge M_m) \rightarrow P$ (equivalent to $(\sim M_1 \vee ... \vee \sim M_m) \vee P$) in which the P is "empty". That is, $\leftarrow(M_1 \wedge ... \wedge M_m)$, or equivalently $(M_1 \wedge ... \wedge M_m) \rightarrow$, is really $\sim(M_1 \wedge ... \wedge M_m)$ or its equivalent $\sim M_1 \vee ... \vee \sim M_m$.

Using the ← notation, resolution may be reformulated as a kind of *modus tollens*:

Sentential Resolution

$$\frac{\leftarrow(\underline{L_1} \wedge ... \wedge \underline{L_m} \wedge L) \quad L \leftarrow(M_{\underline{1}} \wedge ... \wedge M_{\underline{m'}})}{\leftarrow(L_1 \wedge ... \wedge L_m \wedge M_1 \wedge ... \wedge M_{m'})} \qquad \frac{\leftarrow L \qquad L \leftarrow}{\perp}$$

,

Uniform Resolution

$$\frac{\leftarrow(L_{\underline{1}} \wedge ... \wedge L_{\underline{m}} \wedge L)[v_{\underline{1}},...,v_{\underline{n}}] \quad L \leftarrow(M_{\underline{1}} \wedge ... \wedge M_{\underline{m'}})\ [w_{\underline{1}},...,w_{\underline{n}}]}{\leftarrow(L_1 \wedge ... \wedge L_m \wedge M_1 \wedge ... \wedge M_{m'})[t_1,...,t_n]} \qquad \frac{\leftarrow L[v_{\underline{1}},...,v_{\underline{n}}] \quad L \leftarrow [w_{\underline{1}},...,w_{\underline{n}}]}{\perp}$$

Definition 3-10

A ***first-order logic goal*** is any existential quantification of a conjunction of literals: i.e. any $\sim\exists v_1...\exists v_n(M_1\wedge...\wedge M_m)[v_1,...,v_n]$, where $v_1,...,v_n$ are all the free variables in $M_1\wedge...\wedge M_m$. It is for such formulas that the Prolog theorem prover is defined to take as starting points of its procedure. Note that

$$\sim\exists v_1...\exists v_n(M_1\wedge...\wedge M_m)[v_1,...,v_n]$$

is logically equivalent to

$$\forall v_1...\forall v_n(\sim M_1\vee...\vee\sim M_m)[v_1,...,v_n].$$

To test $\leftarrow(M_1\wedge...\wedge M_m)$ in program P =R∪D we attempt to deduce a contradiction. We follow a simple intuitive procedure. We construe the goal as a disjunction of negated literals and draw inferences from it by means of the rule resolution. We do so in a way that attempt to "resolve" it with the various laws and data in the program. We do so in a way that usually reaches a contradiction if there is one to be found. Notice that resolution produces a sentence that is shorter than its two inputs but which incorporates all the relevant information of the premises that has not been shown to be false. This fact invites the following recursive procedure:

Step 1. Resolve the goal with some sentence of the program (if possible) and replace the goal with this resolution.

Step n+1. Resolve the goal generated by Step n with the program and replace that goal with this resolution.

If the process turns up a resolution that is a contradiction (⊥), then stop. The original goal has been refuted and the answer to the query is "yes." If there is no resolution, the answer is "no." Sometimes the procedure runs on forever and there is no answer one way or the other.

Definition 3-11

A Prolog Theorem Prover for a Program P**.**

The Prolog decision procedure $\mathbf{D_P}$ for a program P for formulas of the form $M_1\wedge\ldots\wedge M_m$, producing (for most of these) an answer ***Yes*** or ***No***, is defined as follows. Given a formula $M_1\wedge\ldots\wedge M_m$, the procedure answers ***Yes*** if it succeeds in refuting the opposite formula $\leftarrow(M_1\wedge\ldots\wedge M_m)$, and answers ***No*** in (some) cases in which no refutation is possible.

Step 1. We arrange the program P in some finite order. We identify what we shall call the ***goal value*** as $\leftarrow(M_1\wedge\ldots\wedge M_m)$ (i.e. $\sim M_1\vee\ldots\vee\sim M_m$). For convenience we shall stipulate that the program lists laws (rules) before data (facts). Let P_i be the i-th formula in P, and let the program length k be the cardinality of (*i.e.* the number of formulas in) P. We set up a program counter that keeps track of which formula in the program we are considering and fix its initial value as c=1. We now begin the construction of an annotated proof (a series of lines down th page, each with its "justification"). We use the letter G to represent the line we are constructing. Now set up a counter $\boldsymbol{line_G{=}n}$ indicating the number in the series of the "active line" G and set it at 1. We set G equal to $\leftarrow(M_1\wedge\ldots\wedge M_m)$, and start the proof by writing down as its first line the following:

1. G Temporary Premise

Step 2. If $P_c=(M_1\wedge\ldots\wedge M_m)\leftarrow$, $\boldsymbol{n}$=line$_G$, and n=proof length, **then** write as the next three lines of the proof:

$\boldsymbol{n}$+1. P_c axiom
$\boldsymbol{n}$+2. $\perp$ $\boldsymbol{n}$, $\boldsymbol{n}$+1 Resolution
QED

Write: **Answer to query is *Yes*.** Set $\mathbf{D_P}(M_1\wedge\ldots\wedge M_m)$=***Yes***. Stop.

Step 3. If $\boldsymbol{n}$=line$_G$, $P_c = (M_1\vee\ldots\vee M_m)\leftarrow(P_1\wedge\ldots\wedge P_{m'}\wedge R_1\wedge\ldots\wedge R_{m''})$, and
$G = \leftarrow(M_1\wedge\ldots\wedge M_m\wedge R_1\wedge\ldots\wedge R_{''})$, **then** write as the next two lines of the proof:

$\boldsymbol{n}$+1. P_c axiom
$\boldsymbol{n}$+2. $(P_1\wedge\ldots\wedge P_{m'}\wedge R_1\wedge\ldots\wedge R_{m''})\leftarrow$ $\boldsymbol{n}$, $\boldsymbol{n}$+1 Resolution

Set $G=(P_1\wedge\ldots\wedge P_{m'}\wedge R_1\wedge\ldots\wedge R_{m''})\leftarrow$, line$_G$=$\boldsymbol{n}$+2, c=1, and go to Step 2.

Step 4. If there is no P_c identical to $\leftarrow(M_1\wedge\ldots\wedge M_m)$ or to
$(M_1\vee\ldots\vee M_m)\leftarrow(P_1\wedge\ldots\wedge P_{m'}\wedge R_1\wedge\ldots\wedge R_{m''})$, **then**

If c is less than the program length k, **then** set c=c+1, go to Step 2.
If c= k, **then** write: **No Proof. Answer to query is *No*.**
Set $\mathbf{D_P}(M_1\wedge\ldots\wedge M_m)$=***No***. Stop.

If the program and goal were written in first order logic the full procedure would require the introduction of several steps at the beginning and end of the process. At the start, formulas would be rewritten into logically equivalent prenex normal forms – the negations moved to literals and the quantifiers moved to the outside. At the end the reduction to any contradiction must be converted into a proof of the "goal." Laws must also be universally instantiated.

Definition 3-12

The Prolog Query Handler Viewed as a Theorem Prover in First-Order Logic

Relative to a program $P_{\mathbf{FOL}}=L_{\mathbf{FOL}}\cup D_{\mathbf{FOL}}$, the calculable function **D** is defined mapping a subset of formulas of the form $\exists v_1...\exists v_n(L_1\wedge...\wedge L_m)[v_1,...,v_n]$ into the values {***Yes***,***No***} as follows.

Step 1. Arrange $P_{\mathbf{FOL}}$ in a finite series starting with its laws. Let P_i be the i-th formula in $P_{\mathbf{FOL}}$, and let the following parameter values: set the program length k at the cardinality of $P_{\mathbf{FOL}}$; set the program counter c so that c=1, and write as the first lines of the proof:

1. $\sim\exists v_1...\exists v_n(L_1\wedge...\wedge L_m)\ [v_1,...,v_n]$ — Temporary Premise for Indirect Proof
2. $\forall v_1...\forall v_n(\sim L_1\vee...\vee\sim L_m)[v_1,...,v_n]$ — 1 Quantifier Negation
3. $(\sim L_1\vee...\vee\sim L_m)[v_1,...,v_n]$ — 2 Universal Instantiation

Set $G=(\sim L_1\vee...\vee\sim L_m)\ [v_1,...,v_n]$, set $\text{line}_G=3$.

Step 2. **If** $\text{line}_G=\boldsymbol{n}$, $G=\sim M[w_1,...,w_n]$, and there is some universal instantiation $M[v_1,...,v_n]$ of $P_c=\forall v_1...\forall v_k M[v_1,...,v_n]$, for $0\leq k\leq n$, **then**

If k<0, **then** we write as the next lines of the proof:

- ***n***+1. $\forall v_1...\forall v_n L[v_1,...,v_n]$ — Axiom
- ***n***+2 $L[v_1,...,v_n]$ — ***n***+2 Universal Instantiation
- ***n***+3. $\perp$ — ***n, n***+2 Uniform Resolution
- ***n***+4 $\sim\exists v_1...\exists v_n(L_1\wedge...\wedge L_m)[v_1,...,v_n]$ — 1– ***n***+4 Reduction to Absurd

QED

Write: **Answer to query is *Yes***. Set $\mathbf{D}(M_1\wedge...\wedge M_m)=$***Yes***. Stop.

If k=0 (i.e. when $P_c=M[v_1,...,v_n]$), **then** we write as the next lines of the proof:

- ***n***+1 $L[v_1,...,v_n]$ — Axiom
- ***n***+2. $\perp$ — ***n, n***+1 Uniform Resolution
- ***n***+3 $\sim\exists v_1...\exists v_n)(L_1\wedge...\wedge L_m)[v_1,...,v_n]$ — 1– ***n***+3 Reduction to Absurd

QED

Write: **Answer to query is *Yes***. Set $\mathbf{D}(M_1\wedge...\wedge M_m)=$***Yes***. Stop.

Step 3. **If** $\text{line}_G=\boldsymbol{n}$, $G=R[w_1,...,w_n]\vee\sim M[w_1,...,w_n]$, there is for some universal instantiation $P[v_1,...,v_n]\vee L[v_1,...,v_n]$ of $P_c=\forall v_1...\forall v_k P[v_1,...,v_n]\vee M[v_1,...,v_n]$, for $k\leq n$ and there are some terms $t_1,...,t_n$ free respectively for $v_1,...,v_n$ and $w_1,...,w_n$, **then**

If k<0, **then** we write as the next two lines of the proof:

- ***n***+1. $\forall v_1...\forall v_n\ P[v_1,...,v_n]\vee M[v_1,...,v_n]$ — Axiom
- ***n***+2. $P[v_1,...,v_n]\vee M[v_1,...,v_n]$ — ***n***+1 Universal Instantiation
- ***n***+3 $P[t_1,...,t_n]\vee R[t_1,...,t_n]$ — ***n, n***+2 Resolution

If k=0 (i.e. when $Pc=P[v_1,...,v_n]\vee M[v_1,...,v_n]$, **then** we write as the next two lines of the proof:

- n+1. $P[v_1,...,v_n]\vee M[v_1,...,v_n]$ — Axiom
- n+2 $P[t_1,...,t_n]\vee R[t_1,...,t_n]$ — ***n, n***+1 Uniform Resolution

Set $G=P[t_1,...,t_n]\vee R[t_1,...,t_n]$, $\text{line}_G=n+2$, c=1 and goto Step 2.

Step 4. **If** $P_c=\forall v_1...\forall v_n P[v_1,...,v_n]\vee L[v_1,...,v_n]$ and there is no resolution of $P[v_1,...,v_n]\vee L[v_1,...,v_n]$ and G), **then**

If c is less than the program length k, set c=c+1, goto Step 2.

If c = k, write: **No Proof, answer to query is *No*.** Set $\mathbf{D}(M_1\wedge...\wedge M_m)=$***No***. Stop.

iv. Expert Systems in Prolog.

We are now in a position to draw together the various elements into a complete expert system.

An ***Expert System in Prolog*** (a subsystem of Predicate Logic), for sentences in language $\mathbf{L}_{\mathbf{\mathit{PL}}}$ of predicate logic (with regularized conjunctions, disjunctions, and double negations) consists of

1. A axiom system consisting of
    a. Prolog program P=L∪D
    b. a set **R** of rules of inference containing indirect proof and *modus ponens*,
    c. the set **Th** of theorems constructed from P and **R,**
2. the resolution decision procedure $\mathbf{D}_{\mathbf{P}}$
3. a ***query*** is any Prolog query of $\mathbf{L}_{\mathbf{\mathit{PL}}}$

Definition 3-13
From the perspective of First-Order Logic the intended expert system is defined relative to a first-order Prolog program $P_{\mathbf{FOL}}=L_{\mathbf{FOL}}\cup D_{\mathbf{FOL}}$; its set of theorems **Th** generated by the rule set comprised of Uniform Resolution, Reduction to the Absurd, and Universal Instantiation; and the decision procedure **D** defined relative to $P_{\mathbf{FOL}}$**.**

We will now illustrate Prolog and its logic by working through some programs and queries. We shall see the strengths and weakness of the approach. We shall find successful cases in which the testing procedure produces proofs of theorems and correctly returns the answer ***Yes***; and others in which it fails to produce proofs of non-theorems and correctly returns an answer ***No***. But we shall also find cases in which it is unable to prove theorems or correctly produce a ***Yes*** answer, and others in which it is unable to produce an answer ***No*** for non-theorems. That is, we shall see that when the procedure works and produces an answer, it is correct. But we shall also see that as it stands, the procedure sometimes produces no answer at all. All implementations of Prolog, and indeed all expert systems that approach the syntactic complexity of first-order logic, have similar limitations.

***Example.* A universal law & one concrete datum. A single concrete query. Refutation successful.**

Laws: $Gx \leftarrow Fx$
Database: $Fa \leftarrow$

Query: $Ga?$
Goal: $\leftarrow Ga$
In five steps the query handler write the following proof (the more usual logical notation is given in square brackets):

1.	$\leftarrow Ga$	$[\sim Ga]$	Temporary Premise for a *reductio*
2.	$Ga \leftarrow Fa$	$[Ga \rightarrow Fa]$	Axiom
3.	$\leftarrow Fa$	$[\sim Fa]$	1,2 Resolution
4.	$Fa \leftarrow$	$[Fa]$	Axiom
5.	$\bot$	$\bot$	3,4 Resolution
	QED		
	Answer to query is *Yes.*		

The Example within First-Order Logic

Laws: $\forall x[\sim Fx \vee Gx]$
Database: Fa
Query: $Ga?$
Goal: $\sim Ga$
In seven steps the query handler writes the following proof:

1.	$\sim Ga$	Temporary Premise for a *reductio*
2.	$\forall x(\sim Fx \vee Gx)$	Axiom
3.	$\sim Fa \vee Ga$	2 Universal Instantiation
4.	$\sim Fa$	1,3 Uniform Resolution
6.	Fa	Axiom
7.	$\bot$	4,6 Uniform Resolution
	QED	
	Answer to query is *Yes.*	

Example. **Universal law and datum. An existential query. Refutation successful.**

Laws: $Gx \leftarrow Fx$
Database: $Fx \leftarrow$

Query: *Gx?*
Goal: $\leftarrow Gx$

In five steps the query handler write the following proof (the more usual logical notation is given in square brackets):

1.	$\leftarrow Gx$	$[{\sim}Gx]$	Temporary Premise for a *reductio*
2.	$Fx \leftarrow Gx$	$[Gx \rightarrow Fx]$	Axiom
3.	$\leftarrow Fx$	$[{\sim}Fx]$	1,2 Resolution
4.	$Fx \leftarrow$	$[Fx]$	Axiom
5.	$\perp$	$\perp$	3,4 Resolution

QED
Answer to query is *Yes.*

The Example within First-Order Logic

Laws: $\forall x({\sim}Fx \vee Gx)$
Database: *Fx*
Query: *Gx?*
Goal: ${\sim}Gx$

In seven steps the query handler writes the following proof:

1.	${\sim}Gx$	Temporary Premise for a *reductio*
2.	$\forall x({\sim}Fx \vee Gx)$	Axiom
3.	${\sim}Fx \vee Gx$	2 Universal Instantiation
4.	${\sim}Fx$	1,3 Uniform Resolution
6.	Fx	Axiom
7.	$\perp$	4,6 Uniform Resolution

QED
Answer to query is *Yes.*

***Example.* A complex universal law & multiple concrete data. A complex existential query. Refutation successful.**

Laws: $Hx\leftarrow(Fx\wedge Gx)$
Database: $Fa\leftarrow$
$Ga\leftarrow$
Query: $Hx\wedge Gx?$
Goal: $\leftarrow(Hx\wedge Gx)$

In seven steps the query handler write the following proof (the more usual logical notation is given in square brackets):

1.	$\leftarrow(Hx\wedge Gx)$	$[\sim(Hx\wedge Ga)]$	Temporary Premise
2.	$Hx\leftarrow(Fx\wedge Gx)$	$[(Fx\wedge Gx)\rightarrow Hx$	*A*xiom
3.	$\leftarrow(Fx\wedge Gx)$	$[Fa\rightarrow Ga]$	1,2 Resolution
4.	$Fa\leftarrow$	$[Fa]$	Axiom
5.	$\leftarrow Ga$	$[\sim Ga]$	Resolution
6.	$Ga\leftarrow$	$[Ga]$	Axiom
7.	$\perp$	$\perp$	3,4 Resolution
	QED		
	Answer to query is *Yes.*		

The Example within First-Order Logic

Laws: $\forall x[(\sim Fx\vee\sim Gx)\rightarrow Hx]$
Database: Fa
Ga
Query: $Hx\wedge Gx?$
Goal: $\sim(Hx\wedge Gx)$

In eight steps the query handler writes the following proof:

1.	$\sim(Hx\wedge Gx)$	Temporary Premise
2.	$\forall x[\sim Fx\vee\sim Gx\vee Hx]$	Axiom
3.	$\sim Fx\vee\sim Gx\vee Hx$	2 Universal Instantiation
4.	$\sim Fx\vee\sim Gx$	1,3 Uniform Resolution
5.	Fa	Axiom
6.	$\sim Ga$	4,5 Uniform Resolution
7.	Ga	Axiom
8.	$\perp$	6,7 Uniform Resolution
	QED	
	Answer to query is *Yes.*	

***Example*. Complex universal law & single concrete *datum*. Single concrete query. Refutation unsuccessful.**

Laws: $Hx \leftarrow (Fx \wedge Gx)$
Database: $Fb \leftarrow$

Query: $Hb?$
Goal: $\leftarrow Hb$

In five steps the query handler write the following proof (the more usual logical notation is given in square brackets):

1.	$\leftarrow Hb$	$[\sim Hb]$	Temporary Premise
2.	$Hx \leftarrow (Fx \wedge Gx)$	$[(Fx \wedge Gx) \rightarrow Hx]$	*A*xiom
3.	$\leftarrow (Fb \wedge Gb)$	$[Fb \rightarrow Gb]$	1,2 Resolution
4.	$Fb \leftarrow$	$[Fb]$	Axiom
5.	$\leftarrow Gb$	$[\sim Gb]$	Resolution

No proof. Answer to query is *No.*

The Example within First-Order Logic

Laws: $\forall x[(\sim Fx \vee \sim Gx) \rightarrow Hx]$
Database: Fb
Query: $Hb?$
Goal: $\sim Hb$

In six steps the query handler writes the following proof:

1.	$\sim Hb$	Temporary Premise
2.	$\forall x[\sim Fx \vee \sim Gx \vee Hx]$	Axiom
3.	$\sim Fb \vee \sim Gb \vee Hb$	2 Universal Instantiation
4.	$\sim Fb \vee \sim Gb$	1,3 Uniform Resolution
5.	Fb	Axiom
6.	$\sim Gb$	4,5 Uniform Resolution

No proof. Answer to query is *No.*

***Example*. Multiple universal laws, universal datum. Existential query. A refutation is possible but the procedure goes into an infinite loop and fails to returns an answer.**

Laws:	$\forall x(Gx\leftarrow Fx)$
	$\forall x(Fx\leftarrow Gx)$
Database:	$Fx\leftarrow$
Query:	$Gx?$
Goal:	$\leftarrow Gx$

Note*:** Here the program in fact entails a ***Yes answer because it is easy to prove an affirmative to the query:

1.	$\forall x(Fx\rightarrow Gx)$	Axiom
2.	$\sim Fx$	Temporary Premise
3.	$Fx\rightarrow Gx$	1 Universal Instantiation
4.	$\sim Gx$	2,3 *Modus Tollens*, or Resolution
5.	Fx	2–4 Reductio

The query handler (as defined) however fails to find a proof. It attempts becomes caught in a loop and produces for eternity the following unending series of lines (the more usual logical notation is given in square brackets):

1.	$\leftarrow Gx$	[$\sim Gx$]	Temporary Premise for indirect proof
2.	$Fx\leftarrow Gx$	[$Gx\rightarrow Fx$]	Axiom
3.	$\leftarrow Fx$	[$\sim Fx$]	1,2 Resolution
4.	$Gx\leftarrow Fx$	[$Fx\rightarrow Gx$]	Axiom
5.	$\leftarrow Gx$	[$\sim Gx$]	3,4 Resolution
⋮	⋮	⋮	⋮

The Example within First-Order Logic

Laws:	$\forall x[\sim Fx\vee Gx]$
	$\forall x[\sim Gx\vee Fx]$
Database:	Fx
Query:	$Gx?$
Goal:	$\sim Gx$

The query handler proceeds to write the following infinite series:

1.	$\sim Gx$	Temporary Premise for indirect proof
2.	$\forall x[\sim Fx\vee Gx]$	Axiom
3.	$\sim Fx\vee Gx$	2 Universal Instantiation
4.	$\sim Fx$	1,3 Uniform Resolution
5.	$\forall x[\sim Gx\vee Fx]$	Axiom
6.	$\sim Gx\vee Fx$	5 Universal Instantiation
7.	$\sim Gx$	4,6 Uniform Resolution
8.	$\forall x[\sim Fx\vee Gx]$	Axiom
9.	$\sim Fx\vee Gx$	8 Universal Instantiation
10.	$\sim Fx$	7,9 Uniform Resolution
⋮	⋮	⋮

***Example*. Multiple universal laws, no data. Existential query. No refutation is possible, but the procedure goes into an infinite loop and fails to return an answer.**

Laws: $\forall x[Gx \leftarrow Fx]$
$\forall x[Fx \leftarrow Gx]$
Database: empty
Query: *Gx?*
Goal: $\leftarrow Gx$

Note*:** Here the answer to the query should be ***No because it is easy to find a world in which the program premises are true but the query disconfirmed. For example, in our world the sentences

All dinosaurs are dodoes	$\forall x[Fx \rightarrow Gx]$
All dodoes are dinosaurs	$\forall x[Gx \rightarrow Fx]$

are both true because there are no dinosaurs nor dodoes. Hence a conditional asserting *x* is such would have a false antecedent and therefore be true, for all values of *x* in our world.

But the Prolog query handler for this program produces exactly the same infinite series as in the last example:

1.	$\leftarrow Gx$	$[\sim Gx]$	Temporary Premise for indirect proof
2.	$Fx \leftarrow Gx$	$[Gx \rightarrow Fx]$	Axiom
3.	$\leftarrow Fx$	$[\sim Fx]$	1,2 Resolution
4.	$Gx \leftarrow Fx$	$[Fx \rightarrow Gx]$	Axiom
5.	$\leftarrow Gx$	$[\sim Gx]$	3,4 Resolution
⋮	⋮	⋮	⋮

The Example within First-Order Logic

Laws: $\forall x[\sim Fx \vee Gx]$
$\forall x[\sim Gx \vee Fx]$
Database: empty
Query: *Gx?*
Goal: $\sim Gx$

The query handler proceeds to write the following infinite series:

1.	$\sim Gx$	Temporary Premise for indirect proof
2.	$\forall x[\sim Fx \vee Gx]$	Axiom
3.	$\sim Fx \vee Gx$	2 Universal Instantiation
4.	$\sim Fx$	1,3 Uniform Resolution
5.	$\forall x[\sim Gx \vee Fx]$	Axiom
6.	$\sim Gx \vee Fx$	5 Universal Instantiation
7.	$\sim Gx$	4,6 Uniform Resolution
8.	$\forall x[\sim Fx \vee Gx]$	Axiom
9.	$\sim Fx \vee Gx$	8 Universal Instantiation
10.	$\sim Fx$	7,9 Uniform Resolution
⋮	⋮	⋮

Example. A Kinship System with Database.

Data Base:

	Male?	**Children?**	**Brothers?**
Cain	*yes*	*Henoch*	*Able*
Able	*yes*		*Cain*
Henoch	*yes*	*Irad*	

Data in Logical Notation:

Mc	Pch	Bca
Ma	Phi	Bac
Mh		

Kinship Laws:

A male parent is a father	$\forall x\forall y[\sim Pxy\vee\sim Mx\vee Fx]$
All brothers are male	$\forall x\forall y[\sim Bxy\vee Mx]$
The brother of one's parent is one's uncle	$\forall x\forall y\forall z[\sim Bxy\vee\sim Pxz\vee Uyz]$

Query: Uah?	**Query:** *Uih*?
$\sim Uah$	$\sim Uih$
$\sim Bxa\vee\sim \mathrm{Pxh}$	$\sim Bxi\vee\sim Pxh$
$\sim Bca$	$\sim Bia$
Bca	***No Proof***
$\perp$	***No***

QED. ***Yes***

II. DECIDABILITY AND EXPERT SYSTEMS

A. Conceptual Introduction

This section presents results that show an extremely interesting restriction on human knowledge. In the completeness proof we learned that the set of valid arguments of first-order logic is identical to an inductive set defined in purely syntactic terms. This result captures one of the special epistemic features of logic: its claims about validity are open to a "syntactic proof" in the sense that any valid argument can be shown to be so by constructing a proof tree in natural deduction. These proof trees lend a degree of certainty to claims about validity because the correctness of a proof tree is a mater of simple inspection of shapes on a page.

In this section we show that the validities of first-order logic though constructively characterizable in this sense are not decidable; there is no mechanical decision procedure for testing whether an argument is valid. Another way of saying this is that there is no automatic way to construct a proof tree for a valid formula. Hitting proofs must remain forever, as a mater of mathematical truth, a mater of creativity rather than rote method.

The result shows something about ideas that may not have been evident to start with, namely that the ideas of a constructible set and a decidable set are not the same. There are examples of the one that are not the other. This section show a constructible set that is not decidable.

We have already met the idea of an effective process and a decision procedure. A decidable set is one for which the characteristic function is a decision procedure.

In some cases, however, even though a set is not decidable, we determine in an effective way whether an object is in the set, though there is not effective way to determine whether it is not.

Definition 3-14

A set *C* is ***decidable*** iff it has a characteristic function that is an effective process.

A set *C* is ***recursively enumerable*** iff there is an effective process *f* such that
if $x \in C$, then $f(x)=1$, and
if $x \notin C$, then $f(x)=0$ or the calculation of $f(x)$ does not terminate if $x \notin C$.

The material in this section shows the very interesting result that the relation $\models_{\mathbf{FOL}}$ of valid argument in first-order logic is not decidable but recursively enumerable.

B. Normal Forms and Skolimization.

In this section we do some "house cleaning" in the sense of explaining how to convert any formula into a logical equivalent that has a very useful form: all its quantifiers are at the outside at it’s beginning, and the formula within it is a truth-function.

Important Equivalents:
The following are pairs of logical equivalents with their traditional names:

Association	$(P\wedge Q)\wedge R$	$\Dashv\vDash$	$P\wedge(Q\wedge R)$
	$(P\vee Q)\vee R$	$\Dashv\vDash$	$P\vee(Q\vee R)$
Commutation	$P\wedge Q$	$\Dashv\vDash$	$Q\wedge P$
	$P\vee Q$	$\Dashv\vDash$	$Q\vee P$
Double Negation	P	$\Dashv\vDash$	$\sim\sim P$
DeMorgan’s Laws	$\sim(P\wedge Q)$	$\Dashv\vDash$	$\sim P\vee\sim Q$
	$\sim(P\vee Q)$	$\Dashv\vDash$	$\sim P\wedge\sim Q$
Distribution	$P\wedge(Q\vee R)$	$\Dashv\vDash$	$(P\wedge Q)\vee(P\wedge R)$
	$P\vee(Q\wedge R)$	$\Dashv\vDash$	$(P\vee Q)\wedge(P\vee R)$
Implication	$P\rightarrow Q$	$\Dashv\vDash$	$\sim P\vee Q$
Equivalence	$P\leftrightarrow Q$	$\Dashv\vDash$	$(P\rightarrow Q)\wedge(Q\rightarrow P)$
Quantifier Rules	$\forall vP$	$\Dashv\vDash$	$\sim\exists v\sim P$
	$\exists vP$	$\Dashv\vDash$	$\sim\forall v\sim P$
If v is not free in Q:	$\forall vP\wedge Q$	$\Dashv\vDash$	$\forall v(P\wedge Q)$
If v is not free in Q:	$\exists vP\wedge Q$	$\Dashv\vDash$	$\exists v(P\wedge Q)$
If v is not free in Q:	$\forall vP\vee Q$	$\Dashv\vDash$	$\forall v(P\vee Q)$
If v is not free in Q:	$\forall vP\vee Q$	$\Dashv\vDash$	$\forall v(P\vee Q)$

Metatheorem 3-1. Normal Forms.

- A formula in which $\rightarrow$ and $\leftrightarrow$ occur is equivalent to a formula that use only the sentential connectives $\sim$,$\wedge$, and $\vee$.
- A formula in which negations occur on parts other than literals is equivalent to another formula in which “negation has been driven inside” in the sense that $\sim$ occurs only in literals.
- The quantifiers of a formula may be “driven outside” in the sense that for every formula there is some general formula equivalent to it.
- Any truth-functional formula P is equivalent to one that is a conjunction of disjuncts of literals (called ***the conjunctive normal form*** of P, briefly CNF(P)).
- Any truth-functional formula P is equivalent to one that is a disjunction of conjuncts of literals (called ***the disjunctive normal form*** of P, briefly DNF(P)).

Definition 3-15

A formula is said to be in ***prenex conjunctive/disjunctive normal form*** iff it is some general formula $E_1v_1...E_1v_nQ$ such that Q is a truth-function and in conjunctive/disjunctive normal form.

Metatheorem 3-2. Skolem's Theroem.

The theorem has two parts:

1. If $\forall v_1 \ldots \forall v_n \exists wP$ is satisfiable relative to some variable assignment $\boldsymbol{s}$, then there is some n-place functor f such that $\forall v_1 \ldots \forall v_n P[f(v_1,\ldots,v_n)/w]$ is satisfiable relative to some variable assignment $\boldsymbol{s}'$.
2. (The result in the other direction is stronger.)
   For any model $\mathfrak{A}=<\mathrm{D},\mathfrak{I}>$,
   if $\forall v_1 \ldots \forall v_n P[f(v_1,\ldots,v_n)/w]$ is satisfied relative to a variable assignment $\boldsymbol{s}$,
   then so is $\forall v_1 \ldots \forall v_n \exists wP$.

The theorem may be state more succinctly in symbols:

1. If $\mathfrak{A}_{\boldsymbol{s}} \models \forall v_1 \ldots \forall v_n \exists wP$, then for some f and $\boldsymbol{s}'$, $\mathfrak{A}_{\boldsymbol{s}'} \models \forall v_1 \ldots \forall v_n \exists wP[f(v_1,\ldots,v_n)/w]$
2. $\forall v_1 \ldots \forall v_n P[f(v_1,\ldots,v_n)/w] \models \forall v_1 \ldots \forall v_n \exists wP$

Sketch of Proof. Part 2 is a simple case of existential generalization. For part 1, let $\forall v_1 \ldots \forall v_n \exists wP$ be a formula of $\mathbf{F_{FOL}}$ in which the functor f does not occur, and let $\mathfrak{I}^{\mathfrak{A}}_{s}(\forall v_1 \ldots \forall v_n \exists wP)$=T.
Define the function φ as follows:

$\varphi(\mathrm{d}_1,\ldots,\mathrm{d}_n)$=e iff for some $v_i,\ldots,v_n$-variant $\boldsymbol{s}'$ of $\boldsymbol{s}$, $\boldsymbol{s}'(v_i)=\mathrm{d}_i$, and
for some w-variant $\boldsymbol{s}''$ of $\boldsymbol{s}'$, $\boldsymbol{s}''(w)$=e

We define an interpretation $\mathfrak{I}'$ to be like $\mathfrak{I}$ except that $\mathfrak{I}'(f)=\varphi$. Note that $\mathfrak{I}$ and $\mathfrak{I}'$ have the same variable assignments. By our original assumption, we know that for all $v_i,\ldots,v_n$-variant $\boldsymbol{s}'$ of $\boldsymbol{s}$, $\mathfrak{I}^{\mathfrak{A}}_{s'}(\exists wP)$=T. Hence for all $v_i,\ldots,v_n$-variant $\boldsymbol{s}'$ of $\boldsymbol{s}$, and some w-variant $\boldsymbol{s}''$ of $\boldsymbol{s}'$, $\mathfrak{I}^{\mathfrak{A}}_{s''}(P)$=T. Suppose that $\boldsymbol{s}'$ is a $v_i,\ldots,v_n$-variant of $\boldsymbol{s}$, and $\boldsymbol{s}''$ is a w-variant of $\boldsymbol{s}'$, such that $\mathfrak{I}^{\mathfrak{A}}_{s''}(P)$=T. Since f does not appear in $\exists wP$, neither does $f(v_1,\ldots,v_n)$. By an earlier theorem then $\mathfrak{I}'_{\boldsymbol{s}''}(P)$=T. Moreover, $\mathfrak{I}'_{\boldsymbol{s}''}(f(v_1,\ldots,v_n))= \varphi(\boldsymbol{s}''(v_1),\ldots, \boldsymbol{s}''(v_n))= \boldsymbol{s}''(w)$. Since $\boldsymbol{s}''$ is a w-variant and w does not occur in $P[f(v_1,\ldots,v_n)/w]$ an earlier theorem assures us that $\mathfrak{I}'_{\boldsymbol{s}''}(P)=\mathfrak{I}'_{\boldsymbol{s}''}(P[f(v_1,\ldots,v_n)/w])$. Hence $\mathfrak{I}'_{\boldsymbol{s}''}(P[f(v_1,\ldots,v_n)/w])$=T. **QED.**

Definition 3-16

Skolemization

A formula $\forall v_1...\forall v_n P[f(v_1,...,v_n)/w]$ is called a ***primary Skolemization*** of $\forall v_1...\forall v_n\exists wP$ if $\forall v_1...\forall v_n\exists wP$ is in prenex normal form and *f* does not occur in *P.* To replace other existential quantifiers within *P* we define ***the set of all Skolemizations*** of *P.* The definition is inductive: all primary Skolemizations of *P* are Skolemizations; if *Q* is a Skolemization of *P* and *R* is a Skolemization of *Q*, the *R* is a Skolemization of *P*; nothing else is a Skolemization of *P*. The previous results may be generalized:

Metatheorem 3-3

1. A formula is satisfiable only if its Skolemizations are.
2. The Skolemizations of a formula entail that formula.

Definition 3-17

By a ***complete*** Skolemization of *P* will be meant a Skolemization $\forall v_1...\forall v_n Q$ in prenex universal normal form in which $v_1,...,v_n$ are the free variables that occur in *Q, Q* contains no quantifiers (i.e. is a truth-functional formula) and is in conjunctive normal form. In this case we refer to $\forall v_1...\forall v_n Q$ as $\forall v_1...\forall v_n Q[v_1,...,v_n]$; and we call $Q[v_1,...,v_n]$ a ***bare*** Skolemization of *P.*

C. Herbrand Models

Like the model constructed in Henkin's completeness proof for first-order logic, the entities in the domain of Herbrand models (the "objects' that exist in that model) are terms of the syntax itself. Herbrand models are dissimilar to those of the completeness proof, however, in that they omit terms containing variables. As in the Henkin model, in a Herbrand model a term refers to itself, and predicates refer to a set or relations of terms that satifie terms that appear (in order) in formulas in the set. It then follows that the terms of an atomic predication fall in the interpretation of the predicate iff they, in order, fall within the relation that the predicate stands for. Herbrand models have two special properties.

First, because in Herbrand models the universal quantifier conforms to the substitution rule (it is true iff all its substitution instances for terms are true) and because first-order logic is compact (a set of formulas is satisfiable iff all its finite subsets are), a universal quantification is satisfiable iff all finite subsets of its instances are. But these instances are atomic or negations of atomic sentences without variables in them, and they may be made into conjunction, which may be tested for satisfiability by truth-tables. In this way, testing for satisfiability, by truth-tables, all conjunctions of instances of a universal quantifiers provide a way to test the formula itself. Since issues about validity may be reformulated in terms of satisfiability (because $\{P_1,...P_n\}\models Q$ iff $\{P_1,...P_n,\sim Q\}$ is unsatisfiable), tests of key validities may be approached through testing these instances by truth-tables.

Second, Herbrand models may be made to replicate the truths of non-Herbrand models. Indeed a set of sentences is true a model iff it is true in a Herbrand model. This equivalence also allows for the translation of issues about satisfiability (and validity) relative to all models into issues about Herbrand models, where they are approachable by the truth-table techniques of the last paragraph. Recall that all that it is sufficient for definining a model to specify its domain and define for its interpretation function the values it assigns to constants, functors, and predicates.

Definition 3-18

Let $\mathbf{L_{FOL=}}$ be a language and X a set of formulas of $\mathbf{L_{FOL=}}$ such that every constant, predicate and functor of $\mathbf{L_{FOL=}}$ is contained in some formula of X. Then, a ***Herbrand model*** relative to X is that $<\mathrm{D}_X, \mathfrak{I}^{\mathrm{H}}>$ such that D_X is the set of all the (grounded) terms generated by the constants and functors that occur in X . If no constant occurs in any P of X, then the first constant c_1 of $\mathbf{L_{FOL=}}$ is in D_X. That is, D_X is the inductive set such that:

- if a constant c occurs in some P of X, then c is in D_X, or $c_1 \in \mathrm{D}_X$;
- if an n-place functor f occurs in some P in X and $t_1,...,t_n$ are all in D_X, then $f(t_1,...,t_n)$ is in D_X;
- nothing else is in D_X.

Further, $\mathfrak{I}^{\mathrm{H}}$ on D_X is that interpretation defined as follows:

- $\mathfrak{I}^{\mathrm{H}}(c)=c$ (if there is some c that occurs in X)
- $\mathfrak{I}^{\mathrm{H}}(f^n)= \{<t_1,\ldots,t_n, t_{n+1}> \mid \ f(t_1,\ldots,t_n)= t_{n+1} \}$ (if there is some f^n that occurs in X)
- $\mathfrak{I}^{\mathrm{H}}(P^n)=\{ \{<t_1,\ldots,t_n> \mid \ P^n t_1,\ldots,t_n \in X \}$

Let $\mathfrak{A}^{\mathrm{H}}=<\mathrm{D}_X,\mathfrak{I}^{\mathrm{H}}>$ range over Herbrand models. When X contains just one sentence P, i.e. when $X = \{P\}$, we write D_P instead of $\mathrm{D}_{\{P\}}$.

Remark. In an Herbrand model:
$\mathfrak{I}^{\mathrm{H}}(c)$ is the constant c, which occurs in X;
$\mathfrak{I}^{\mathrm{H}}(f^n(t_1,\ldots,t_n))$ is a grounded term made up of constants that occur in X, and if each of $t_1,\ldots,t_n$ is grounded, then $\mathfrak{I}^{\mathrm{H}}(f^n(t_1,\ldots,t_n))$ is $f^n(t_1,\ldots,t_n)$ itself;
$\mathfrak{I}^{\mathrm{H}}(P^n)$ is a relation on grounded terms made up of constants that occur in X and iff $<t_1,\ldots,t_n>$ is in $\mathfrak{I}^{\mathrm{H}}(P^n)$ iff the sentence $P^n t_1,\ldots,t_n$ is itself in X;
$\mathfrak{I}^{\mathrm{H}}(t=t')$ will be F, for any two grammatically distinct terms t and t'.

Metatheorem 3-4

Let $\mathfrak{A}^H$=<D_X,$\mathfrak{I}^H$> be a Herbrand model.

- The sentence (without free variables) $P^n{}_m t_1 \ldots t_n$ is T in $\mathfrak{I}^H$ iff $<\mathfrak{I}^H(t_1),\ldots,\mathfrak{I}^H(t_n)> \in \mathfrak{I}^H(P^n{}_m)$ iff $< t_1,\ldots,t_n> \in \mathfrak{I}^H(P^n{}_m)$
- If $f(t_1 \ldots t_n)$ is groiunded, $\mathfrak{I}^H(f(t_1 \ldots t_n)) = \mathfrak{I}^H(f)(\mathfrak{I}^H(t_1),\ldots,\mathfrak{I}^H(t_n)) = f(t_1 \ldots t_n)$
- More generally, if t is grounded, $\mathfrak{I}^H(t)= t$
- If $\mathfrak{A}^H$ is a model of D_X and some P in X contains a functor, then D_X is countably infinite.
- $\forall vP[v]$ is T in $\mathfrak{I}^H$ on domain $D_{P[v]}$ (identical to $D_{\forall vP[v]}$) iff for all $t \in D_{P[v]}$, $P[t]$ is T in $\mathfrak{A}^H$.
- The following are equivalent:
  1. $\mathfrak{A}^H \models \forall v_1 \ldots \forall v_n P[v_1,\ldots,v_n]$
  2. for all $t_i \in D_P$, $\mathfrak{A}^H \models P[t_1,\ldots,t_n]$
  3. $\mathfrak{A}^H \models \{P[t_1,\ldots,t_n]\}_{t_I \in D_P}$

Remark. As unpacked in its Herbrand truth-conditions, universal quantification in Herbrand models exhibits several simplifying properties not satisfied in models at large. Let $\mathfrak{A}^H$ be a Herbrand model for P on D_P. Then:

- The domain is at most countably infinite.
- No two constants or grounded terms stand for the same entity. (Hence identity statements with distinct grounded terms are false, and as a result programming languages that presuppose a Herbrand model (e.g. logic programming languages like *Prolog*) must eschew the ordinary arithmetic for the integers, e.g. it becomes false that 2+2=4. In practice these languages graft on a "calculator" that is not part of the main language and that is awkwardly integrated with it.))
- All the objects in the domain have a name that either occurs in P or is generated from the terms that occur in P.
- An open sentence is true iff every substitution instance is true for all grounded terms.

Metatheorem 3-5. Herbrand's Theorem.

P is satisfiable iff P is satisfiable in some Herbrand model on D_P. That is,

$\exists$B, B $\models P$ iff, $\exists \mathfrak{A}^H$ such that $\mathfrak{A}^H$=< D_P,$\mathfrak{I}^H$> and $\mathfrak{A}^H \models P$

Proof Sketch. The theorem is proven by constructing for any model an equivalent Herbrand model. A simple technique is to exploit the Skolem-Löwenheim Theorem. First by appeal to the theorem convert a model into an equivalent model of a countable infinite domain. This conversion either behaves like a Herbrand model in that no two terms refer to the same term, or it does not. If it does it is isomorphic to some Herbrand model, and hence the two have the same truths. If it contains some terms t and t' that stand for the same entity e, we inflate the model's domain by adding an entity e' that is indistinguishable for e in the sense that it is in the extension of every predicate that e is. If any functor f is defined for t and t' such that we make $f(\ldots t\ldots)=e$ we introduce an entity $e'=f(\ldots t'\ldots)$ that is in the extension of all predicates that e is. Since the language does not contain the identity predicate, the introduction of these entities will not affect the truth-values of any formula (the proof is by induction). Now this revised model has a denumerable domain and now has no terms standing for the same entity. It is therefore isomorphic to some Herbrand model, and would have the same truths as it.

Definition 3-19

P is ***truth-functionally satisfiable*** iff *P* is a truth-function and is satisfiable.

By definition, any formula without quantifiers or free-variables is a truth-function. The special interest in truth-functions is that since they lack variables and quantifiers they behave just like formulas from sentence logic, and it is easy to test they using truth-tables for satisfiability (some line of the table has T under the major connective), for absurdity (every line has F), and for tautologousness (every line has T). Hence, if *P* is truth-functionally satisfiable, this means that its satisfiability may be tested using truth-tables.

Metatheorem 3-6

Let X be a finite set of truth-functional formulas and $\wedge X$ the conjunction of all its elements:

X is satisfiable iff $\wedge X$ is truth-functionally satisfiable (by truth-tables)

Remark. Truth-functional satisfiability has the important property that it is decidable. The truth-table test provides a technique for defining a calculable decision procedure □(i.e. an algorithmic characteristic function) for the set of all truth-functionally satisfiable formula: □(*P*)=1 if 1 is listed under the major connective in some line of the truth-table of *P*, and □(*P*)=0 otherwise.

Corollary. Truth-Functional Satisfiability.

$\wedge X$ is satisfiable iff for some $\mathfrak{A}^{H}$, $\mathfrak{A}^{H} \vDash \wedge X$ (by Herbrand's Theorem)

We are now ready to prove the key fact needed later. Since it is a fact that, as it stands, lacks little intrinsic interest, it is called a "lemma." Its role in the overall proof however is conceptually critical, because it is in the proof of this lemma that all the important assumption necessary to the overall proof are used: Skolem's theorem. Herbrand's theorem, and the compactness theorem. Below we abbreviate the notation $a_1 \in C$ & … & $a_n \in C$, which asserts that all the elements $a_1,\ldots,a_n$ are in *C*, by the notation $a_1,\ldots,a_n \in C$.

Summary of Relevant Results:

Herbrand's Theorem

$\exists$B, B$\models P$ iff, $\exists\ \mathfrak{A}^{H}$ such that $\mathfrak{A}^{H}$ =< D_P,$\mathfrak{I}^{H}$> and $\mathfrak{A}^{H}\models P$

Skolem's Theorem

A formula is satisfiable iff its Skolemizations are.

Truth-Conditions for $\forall v_1 ... \forall v_n P[v_1,...,v_n]$ in a Herbrand model

$\mathfrak{A}^{H}\models\forall v_1...\forall v_n P[v_1,...,v_n]$ iff for all $t_1,...,t_n \in D_P$, $\mathfrak{A}^{H}\models P[t_1,...,t_n]$

Truth-Functional Satisfiability

$\wedge X$ is satisfiable iff for some $\mathfrak{A}^{H}$, $\mathfrak{A}^{H}\models\wedge X$

Metatheorem. Compactness

Entailment Formulation:

X$\models Q$ iff, there is some finite subset $\{P_1,...P_n\}$ of X such that $P_1,...P_n\models$Q.

Satisfiability Formulation:

$\mathfrak{A}\models X$ iff for all finite subset Y of X, $\mathfrak{A}\models Y$

Lemma. Let $\forall v_1...\forall v_n Q[v_1,...,v_n]$ be a complete Skolemization of P. The following equivalence then obtains:

$\exists\mathfrak{A}, \mathfrak{A}\models P \quad\Leftrightarrow\quad \forall X$, X is finite & $X\subseteq\models\{Q[t_1,...,t_n] \mid t_1,...,t_n \in D_Q\} \Rightarrow \wedge X$ is truth-functionally satisfiable

Proof:

$\exists\mathfrak{A}, \mathfrak{A}\models P \Leftrightarrow \exists\mathfrak{A}, \mathfrak{A}\models\forall v_1...\forall v_n Q[v_1,...,v_n]$ (by Skolem's Theorem)

$\Leftrightarrow \exists\mathfrak{A}^{H}$ on D_Q, $\mathfrak{A}^{H}\models\forall v_1...\forall v_n Q[v_1,...,v_n]$ (Herbrand's Theorem)

$\Leftrightarrow \exists\mathfrak{A}^{H}$ on D_Q, $\mathfrak{A}^{H}\models\{Q[t_1,...,t_n] \mid t_1,...,t_n \in D_Q\}$ (truth-conditions for $\forall$ In Herbrand models)

$\Leftrightarrow \exists\mathfrak{A}, \mathfrak{A}\models\{Q[t_1,...,t_n] \mid t_1,...,t_n \in D_Q\}$ (by Herbrand's Theorem)

$\Leftrightarrow \forall X$, X is finite & $X\subseteq\{Q[t_1,...,t_n] \mid t_1,...,t_n \in D_Q\} \Rightarrow \exists\mathfrak{A}, \mathfrak{A}\models X$ (Compactness)

$\Leftrightarrow \forall X$, X is finite & $X\subseteq\{Q[t_1,...,t_n] \mid t_1,...,t_n \in D_Q\} \Rightarrow \exists\mathfrak{A}, \mathfrak{A}\models\wedge X$ (truth-tables)

$\Leftrightarrow \forall X$, X is finite & $X\subseteq\{Q[t_1,...,t_n] \mid t_1,...,t_n \in D_Q\} \Rightarrow \exists\mathfrak{A}^{H}$, $\mathfrak{A}^{H}\models\wedge X$ (Herbrand's Theorem)

$\exists\mathfrak{A}, \mathfrak{A}\models P \Leftrightarrow \forall X$, X is finite & $X\subseteq\{Q[t_1,...,t_n] \mid t_1,...,t_n \in D_Q\} \Rightarrow \wedge X$ is truth-functionally satisfiable (X is a set of grounded literals)

We now introduce an intuitive concept in terms of which we can reformulate the lemma (by contraposition) into a metatheorem that is conceptually easier to understand.

Definition 3-20

Let P is ***truth-functionally refutable*** iff there is a bare Skolemization $Q[v_1,...,v_n]$ of P and $\exists X$, [X is finite & $X\subseteq\{Q[t_1,...,t_n] \mid t_1,...,t_n \in D_Q\}$ and $\wedge X$ is truth-functionally unsatisfiable.

Metatheore 3-7

P is truth-functionally refutable $\Leftrightarrow$ P is unsatisfiable (i.e. not($\exists\mathfrak{A}$, $\mathfrak{A}\models P$))

Proof. not($\forall X$, X is finite & $X\subseteq\{Q[t_1,...,t_n] \mid t_1,...,t_n\in D_Q\} \Rightarrow \wedge X$ is truth-functionally satisfiable) $\Rightarrow$ ($\exists\mathfrak{A}$, $\mathfrak{A}\models P$)

$\exists X$, [X is finite & $X\subseteq\{Q[t_1,...,t_n] \mid t_1,...,t_n\in D_Q\}$ & $\wedge$ X is truth-functionally unsatisfiable $\Rightarrow$ not($\exists\mathfrak{A}$, $\mathfrak{A}\models P$)

P is truth-functionally refutable $\Leftrightarrow$ not($\exists\mathfrak{A}$, $\mathfrak{A}\models P$)

Metatheorem 3-8

Truth-functionally refutability is ***recursively enumerable***: there is an effective procedure Π*applied all such that: for any formula P,

$\mathfrak{A}\models P$ iff $\Pi(P)=1$
not(P) iff $\Pi(P)=0$ or $\Pi(P)$ is incalculable and undefined.

Proof: We find some bare Skolemization Q of P and define $\Pi(P)$:

We note that the set of finite subset of $\{Q[t_1,...,t_n] \mid t_1,...,t_n\in D_Q\}$ is itself possibly countably infinite (it will be infinite in the case in which there are an infinite number of grounded formulas in D_Q). Let us first pair each finite subset C^n of $\{Q[t_1,...,t_n] \mid t_1,...,t_n\in D_Q\}$ of cardinality n with some conjunction of its elements $\wedge C^n$. Clearly $\{Q[t_1,...,t_n] \mid t_1,...,t_n\in D_Q\}$ is satisfiable iff $\wedge C^n$ is. Moreover each such $\wedge C^n$, for n=1,…., is a truth-function and is testable for satisfiability by truth-tables, i.e. the truth-table method provides an effective process (decision procedure) for deciding in a finite number of epistemically transparent steps ***Yes*** iff $\wedge C^n$ is satisfiable and ***No*** iff it is not. We now list these conjunction , for n=1,…., according to their numerical index: L = $\wedge C^n_1$,…,$\wedge C^n_m$. We now test each $\wedge C^n$ by truth-tables for satisfiability. If some $\wedge C^n$ is truth-functionally unsatisfiable then we set $\Pi(P)=1$. If no such $\wedge C^n$ is truth-functionally unsatisfiable, then we set $\Pi(P)=0$. (Note that these two cases are exhaustive.)

Clearly, by Skolem's theorem P is unsatisfiable iff its Skolemization Q is also unsatisfiable, which in turn, by Herbrand's Theorem, is unsatisfiable is unsatisfiable iff the null set of Q's the grounded instances, i.e. $\{Q[t_1,...,t_n] \mid t_1,...,t_n\in D_Q\}$, is also unsatisfiable. But if $\{Q[t_1,...,t_n] \mid t_1,...,t_n\in D_Q\}$ is unsatisfiable, then there is, for some finite number n, a conjunction $\wedge C^n$ equivalent to the subset A of $\{Q[t_1,...,t_n] \mid t_1,...,t_n\in D_Q\}$ such that $\wedge C^n$ is unsatisfiable. We now define the procedure Π. We proceed down the list L, testing by truth-tables each element $\wedge C^i$ of the list in order. At each position i, we test $\wedge C^i$ for unsatisfiability and if it is unsatisfiable, we set $\Pi(P)$=1. If $\wedge C^i$ is satisfiable we set $\Pi(P)$=0. We then proceed to step i+1. Clearly if P is unsatisfiable the process will eventually reach some $\wedge C^i$, that test to be unsatisfiable and hence $\Pi(P)$=1. On the other hand, if P is satisfiable, and $\{Q[t_1,...,t_n] \mid t_1,...,t_n \in \mathrm{D}_Q\}$ and hence L, is infinite, no such unsatisfiable conjunct will ever be encountered and the process Π will never terminate. QED.

Metatheorem 3-9. Undecidability of First-Order Logic.

The entailment relation $\models$ of $\mathbf{L_{FOL=}}$ is ***recursively enumerable*** in the sense that there is an effective procedure Π* applied all such that:

$X\models Q$ iff Π*(<X,Q>)=1.
not(($X\models$Q) iff Π*(<X,Q>)=0 or Π*(<X,Q>) is incalculable and undefined.

Proof: To test whether $X\models Q$, we define a partial function Π* from arguments of the form <X,Q> to the values {1,0}.

For an argument <X,Q> we proceed as follows. We first list all the finite subsets of X by cardinality groups as in the last proof in a possible infinite list: $C_1^1,\ldots,C_{m_1}^1;\ldots;C_1^n,\ldots,C_{m_n}^n;C_1^{n+1},\ldots,C_{m_{n+1}}^{n+1};\ldots$. Each C_j^i is the list is some finite set of sentences $\{P_1,...P_n\}$. By the sentence associated with C_j^i we mean $P_1\wedge...\wedge P_n\wedge{\sim}Q$. We now test $P_1\wedge...\wedge P_n\wedge{\sim}Q$ for truth-functional refutability by the decision procedure Π. If Π $(P_1\wedge...\wedge P_n\wedge{\sim}Q)$=1 (i.e. if $P_1\wedge...\wedge P_n\wedge{\sim}Q$ is refutable) then by the theorem, $P_1\wedge...\wedge P_n\wedge{\sim}Q$ is unsatisfiable and hence $\{P_1,...P_n\}\models Q$. If so $X\models Q$ and we set Π*(<X,Q>)=1 (i.e. it is decided that $X\models$Q). If not, we test the formula associated with the next set in the list. If the list is finite and all associated formulas test to be non-refutable we set Π*(<X,Q>)=0. If the list is infinite and no item in the list is reached so that its associated formula test to be refutable the testing procedure Π* does not terminate and □*(<X,Q>) is undefined. If <X,Q> is valid, however, we know by the compactness theorem (and indirectly by completeness) that there is some finite subset C_j^i of X such that $C_j^i\models Q$., and hence that for its associated $P_1\wedge...\wedge P_n\wedge{\sim}Q$, $\Pi(P_1\wedge...\wedge P_n\wedge{\sim}Q)$=1. Hence in cases in which $X\models Q$, it follows that Π*(<X,Q>)=1. Moreover in those cases in which Π*(<X,Q>)=0, we know that not($X\models Q$) because the only time that the list of subsets of X is finite is that in which X itself is finite, and is itself one of the C_j^i. In that case $X\cup\{{\sim}Q\}$ is satisfiable and hence not(($X\models$Q). Hence,

$X\models Q$ iff Π*(<X,Q>)=1, and
not(($X\models$Q) iff Π*(<X,Q>)=0 or Π*(<X,Q>) is undefined.

Moreover, when Π^* is defined it is effective. Hence, the validity relation $\models_{\textbf{FOL}}$ is recursively enumerable. **QED.**

III. EXERCISES

A. Skills

1. Find the greatest common divisor of 22 and 56.
2. Construct a Prolog program using the following data base:
   Small(*a*)
   Medium(*b*)
   Large(*c*)
   a. Add a law that states the necessary conditions for *Larger-than*(*x,y*) in terms of *Small* and *Medium*. Add a second law that does so in terms of *Medium* and *Large*. Add a third law that states the transitivity of *Larger-than***.**
   b. Construct an annotated proof writing it once in Prolog notation and once in FOL notation in which you process the query: *Larger-than*(*a,c*)?
   c. Construct an annotated proof writing it once in Prolog notation and once in FOL notation in which you process the query: *Larger-than*(*c,a*)?

3. A Prolog goal is defined very generally so that the sentence that is being tested is a literal, either positive or negative. Accordingly, it is permitted that a goal be negated, *e.g.* ←~*Gx.* Depending on the program, when run, it may then produce the answer ***yes*** or ***no***. Suppose the goal is ←~*Gx.*
   a. If ←~*Gx* is the goal, and running the program yields the answer ***yes***, what is the formula in FOL notation that will have been proven to be logically entailed by the program?
   b. If ←~*Gx* is the goal, and running the program yields the answer ***no,*** what is the formula in FOL notation that will have been proven to be logically entailed by the program? Would the formula proven to be entailed in this case be the same as that proven in a second case in which the goal is ←*Gx* and the program produces the answer ***yes****?* Explain.

B. Ideas

i. Effective Processes

a. The intuitive account of effective process above stresses its epistemic transparency and for that reason gives examples that consist of manipulations of perceptible syntactic entities. These are literally marks on a page, or "tokens" that exemplify is an "evident" manner expression "types." Some of the formal inductive characterizations of *effective process* do in fact appeal to be defined for syntactic entities of this sort: Turing machines apply to numerals, and Markov algorithms apply to linguistic rule sets. Others, however, namely Gödel’s definition of primitive recursive functions, Post’s algebras and Church’s lambda computable functions, are defined on numbers. But “numbers” are odd entities, usually thought of as abstract and non-concrete. Can an epistemically transparent operations apply to an abstract object? Answer by considering clear cases of operations that are and that are not “epistemically transparent.”

b. The inductively defined functions declared identical to effective processes by Church’s thesis may be computed by computational iterations of any finite length, even if that length is superhuman. In recent years, for example, numbers have been discovered to be primes by computers using calculations of such length that no human could reproduce them. Similarly there are now proofs in mathematics that can be checked for formal adequacy only by computer. Such finite but superhuman calculations meet the requirements of inductive definitions. Do they also satisfy the epistemic requirement for an effective process?

ii. Herbrand Models and Soklemizations

Give short answers to the following:

c. How does a Herbrand model differ from an ordinary model?

d. Though a formula is satisfiable in a model iff it is satisfiable in a Herbrand model (this is Herbrand's Theorem) this no longer remains true if the syntax contains the identity predicate. Why? (*Hint.* think of the sentence c=c□ when c and c□ are distinct constants.)

e. Why is the complete Skolemization of P not logically equivalent to P?

f. Why it is true that *P* is truth-functionally refutable iff it is not satisfiable.

C. Theory

Undecidability depends on details it will be easy to forget. Set out in your own words that you will be able to understand later how the following key elements:

1. The partial decision procedure for determining whether a formula is truth-functionally satisfiable, and why the process of testing subsets of formulas terminates in some cases but not others.
2. How the test may be extended to a test of whether an argument is valid, and why the test would only return ***yes*** if it is valid but may never return a ***no*** answer if it is not.
3. Why if *P* is satisfiable the conjunction of any finite subset of substitution instance of its Skolemization is satisfiable in a Herbrand model.

Chapter 4

Many-Valued and Intensional Logic

I. ABSTRACT STRUCTURES

A. Structure

We all have a good intuitive idea of a "structure." Examples include buildings, governmental institutions, ecologies, and polyhedral. In the branch of mathematics known as ***abstract*** or ***universal algebra*** the general properties of structures are studied, and these ideas help explain the structures we find in logic like those of grammars, semantical interpretations, and inferential systems.

The raw intuition behind the mathematical definition of a structure is an architect's blueprint. The blue print succeeds in describing a building by first listing its various materials and then by a diagram describing the relations that must obtain among these "building blocks" in the finished structure. In algebra a structure is defined in a similar way. First a list of set $A_1,...,A_k$ is given. These may be viewed as list of building blocks divided into various kinds or classes. Next are listed the relations $R_1,...,R_i$ and functions $f_1,...,f_m$ that hold among these materials. (Recall that functions are just a sub-variety of relations.) Lastly it is useful to list some specific individual building blocks $O_1,...,O_m$ that have special importance in the structure. It is customary to list all the elements of the structure in order, *i.e.* as an ordered tuple:

$$< A_1,...,A_k,R_1,...,R_i,f_1,...,f_m,O_1,...,O_m >.$$

Definition -1. Abstract Structure.

1. An ***abstract structure*** is any $<A_1,...,A_k,R_1,...,R_l,f_1,...,f_m,O_1,...,O_n>$ such that:
for each i=1…k, A_l is a set,
for each i=1…l, R_i is a relation on $C = \mathrm{U}\{A_1,...,A_n\}$,
for each i=1…m, f_i is a function on $C = \mathrm{U}\{A_1,...,A_n\}$, and
for each i=1…m, $O_i \in C = \mathrm{U}\{A_1,...,A_n\}$.

It is also common to investigate a family of structures with similar properties, and to assign the family a name, e.g. group, ring, lattice, or Boolean algebra. The properties defining such a family are usually formulated as defining conditions on the type of sets, relations, functions and designated elements that fall into the family. Sometimes these restrictions are referred to as the "axioms" of the structure-type. Strictly speaking they are not part of a genuine axiom system. Rather they are clauses appearing in the abstract definition of a particular set (family) of structures. Let us review some familiar examples.

B. Sentential Syntax

The usual definition of syntax for sentential logic may be recast so that it is clear that the rules of syntax "define" a certain kind of syntactic "structure." Let us begin by stating a version of the definition of the sort that usually appears in elementary logic texts, and which does not use algebraic ideas explicitly:

Definition 4-2. Well-Formed Formulas of Sentential Syntax.

The set of $\mathbf{F_{SL}}$ of (***Well-Formed***-) ***Formulas of*** **SL.**
Let $\mathbf{AF_{SL}}$ be the basis set {*A,B,C*} (of ***atomic formulas***) and let $\mathbf{R_{SL}}$ be the rule set $\{R_\sim,R_\wedge,R_\vee,R_\rightarrow,R_\leftrightarrow\}$ (of ***grammar rules***) defined below:

a. (Rule $R_\sim$) The result of applying $R_\sim$ to P is $\sim P$.
b. (Rule $R_\wedge$) The result of applying $R_\wedge$ to P and Q is $(P\wedge Q)$.
c. (Rule $R_\vee$) The result of applying $R_\vee$ to P and Q is $(P\vee Q.)$
d. (Rule $R_\rightarrow$) The result of applying $R_\rightarrow$ to P and Q is $(P\rightarrow Q)$.
e. (Rule $R_\leftrightarrow$) The result of applying $R_\leftrightarrow$ to P and Q is $(P\leftrightarrow Q)$.

Then, $\mathbf{F_{SL}}$ is the set inductively defined relative to $\mathbf{AF_{SL}}$ and $\mathbf{R_{SL}}$ as follows:

1. (Basis Clause) All formulas in $\mathbf{AF_{SL}}$ are in $\mathbf{F_{SL}}$.
2. (Inductive Clause) If P and Q are in $\mathbf{F_{SL}}$, then the results of applying the rules $R_\sim,R_\wedge,R_\vee,R_\rightarrow,R_\leftrightarrow$ from in $\mathbf{R_{SL}}$, namely $\sim P$, $(P\wedge Q)$, $(P\vee Q)$, $(P\rightarrow Q)$, $(P\leftrightarrow Q)$, are in $\mathbf{F_{SL}}$.
3. (Closure) Nothing is in $\mathbf{F_{SL}}$ except by Clauses 1 and 2.

Using the idea of an abstract structure, it is possible to reformulate the definition in a way that makes the structural aspects of the grammar fully explicit:

Definition 4-3. Sentential Syntax, Algebraic Formulation.

By a ***sentential logic syntax*** is meant any structure $\mathbf{Syn_{SL}}$=< $F_{SL},R_\sim,R_\wedge,R_\vee,R_\rightarrow,R_\leftrightarrow$> such that:

1. $R_\sim,R_\wedge,R_\vee,R_\rightarrow,R_\leftrightarrow$ are functions on symbol strings defined as follows:
   $R_\sim$ constructs $\sim x$ from any sting x;
   $R_\wedge$ constructs $(x\wedge y)$ from strings x and y;
   $R_\vee$ constructs $(x\vee y)$ from strings x and y;
   $R_\rightarrow$ constructs $(x\rightarrow y)$ from strings x and y;
   $R_\leftrightarrow$ constructs $(x\leftrightarrow y)$ from strings x and y.
2. There is a denumerable set of strings AF_{SL} (called the set of ***atomic formulas***), such that F_{SL} (the set of ***Well-Formed-Formulas of*** **SL**) is defined inductively as follows:
   a. (Basis Clause) All formulas in AF_{SL} are in F_{SL}.
   b. (Inductive Clause) If P and Q are in F_{SL}, then the results of apply the rules $R_\sim,R_\wedge,R_\vee,R_\rightarrow,R_\leftrightarrow$ from in R_{SL} , namely $\sim P$, $(P\wedge Q)$, $(P\vee Q)$, $(P\rightarrow Q)$, $(P\leftrightarrow Q)$, are in F_{SL}.
   c. (Closure) Nothing is in F_{SL} except by clauses 1 and 2.

Once a syntax is defined as a structure, then algebraic ideas may be applied to "explain" it, as "explanation" is understood in mathematics: specific properties of grammar can be seen to hold not as a result of peculiarities of grammar but as consequences of the fact that grammars happens to be special cases of yet more abstract types of structures. Many features of grammars do in fact hold because grammars happen to be subspecies of more abstract structure-types. A particularly interesting and simple case is that of partial orderings.

C. Partial Orderings

The familiar "less than" relation on numbers, symbolized by ≤, and the subset relation on sets, symbolized by ⊆, are instances of what is known as partial ordering. In algebra such orderings are viewed as structures. To define such a structure, however, we must first define some standard properties of relations. We then define several common varieties of ordered-structures.

Definition 4-4. Properties of Relations and Ordered Structures

A binary relation ≤ is said to be:

reflexive iff for any *x*, $x \leq x$;
transitive iff for any *x*,*y*, and *z*, if $x \leq y$ and $y \leq z$, then $x \leq z$;
symmetric iff for any *x* and *y*, if $x \leq y$ then $y \leq x$;
asymmetric iff for any *x* and *y*, if $x \leq y$ then not ($y \leq x$);
antisymmetric iff for any *x* and *y*, if $x \leq y$ and $y \leq x$, then $x = y$;
complete iff for any *x* and *y*, either $x \leq y$ or $y \leq x$;
x is a **≤*-least element*** of *B* iff $x \in B$ and for any $y \in B$, $x \leq y$.

Any structure $<A,\leq>$ such that *A* is a non-empty set and ≤ is a binary relation on *A* is called:

1. a ***pre-*** or ***quasi-ordering*** iff ≤ is reflexive and transitive;
2. a ***partially ordering*** iff ≤ is a pre-ordering and antisymmetric;
3. a ***total*** or ***linear*** ordering iff ≤ is partial and complete;
4. a ***well-ordering*** iff, ≤ is a partial ordering and
   for any subset *B* of *A, B* has a ≤-least element.

Definition 4-5

The ***subformula relation*** ≤ (read "is a part of") is defined on a sentential structure $\textbf{Syn}_{\textbf{SL}} = < F_{SL}, R_{\sim}, R_{\wedge}, R_{\vee}, R_{\rightarrow}, R_{\leftrightarrow} >$ defined inductively as follows:

For any atomic formula, $A \leq A$;
For any molecular formula $R_i(A_1,...,A_n)$, each A_k, for k=1,…,n,
is such that $A_k \leq R_i(A_1,...,A_n)$.

Metatheorem 4-1

The subformula relation ≤ of any $\textbf{Syn}_{\textbf{SL}}$ is a partial ordering.

Another example of ideas from logic that lend themselves to algebraic formulation are truth-tables. Viewed algebraically, truth-tables form a structure on the values {T,F} and each truth-table defines a specific function defining structure on this minimal set.

Definition 4-6. The Classical Bivalent Structure of Truth-Values

By the ***classical algebra of truth-values*** is meant the structure $<\{T,F\},\wedge,\vee,\rightarrow,\leftrightarrow,\sim>$ such that

$\wedge=\{<T,T,T>,<T,F,F>,<F,T,F>,<F,F,F>\}$
$\vee=\{<T,T,T>,<T,F,T>,<F,T,T>,<F,F,F>\}$
$\rightarrow=\{<T,T,T>,<T,F,F>,<F,T,T>,<F,F,T>\}$
$\leftrightarrow=\{<T,T,T>,<T,F,F>,<F,T,F>,<F,F,T>\}$
$\sim=\{<T,F>,<F,T>\}$

Often T is identified with 1 and 0 with F, and {0,1} with 2.

D. Standard Abstract Structures

The structure of truth-values is actually a special case of a more general (i.e. abstract) set of structures known as Boolean algebras, which includes the standard algebra of sets. There are a number of equivalent ways to define a Boolean algebra, some of which we shall encounter later, but for purposes of illustration here let us use a simple definition that employs the idea of partial ordering.

Definition 4-7. Properties of Binary Operations (aka Functions).

Let • be a binary operation on a set B, and let us write $\bullet(x,y)$ as $x\bullet y$. Then,

B is ***closed under •*** iff for all x,y of B, $x\bullet y\in B$,
• is ***associative*** iff for all x,y of B, $x\bullet y=y\bullet x$,
• is c***ommutative*** iff for all x,y of B, $x\bullet(y\bullet z)=(x\bullet y)\bullet z$,
• is ***idempotent*** iff for all x,y of B, $x\bullet x=x$,

Definition 4-8. Varieties of Structures.

A structure $\langle B,\wedge\rangle/\langle B,\vee\rangle$ is a ***meet/join semi-lattice*** iff $\wedge/\vee$ is a binary operation under which B is closed and $\wedge/\vee$ is associative, commutative, and idempotent.

If $\langle B,\wedge\rangle$ is a meet semi-lattice, then the ordering relation $\leq$ on B is defined as
$x\leq y$ iff $x\wedge y=x$.
If $\langle B,\wedge\rangle$ is a join semi-lattice, then the ordering relation $\leq$ on B is defined as
$x\leq y$ iff $x\vee y=y$.

The structure $\langle B,\wedge,\vee\rangle$ is a **lattice** iff $\langle B,\wedge\rangle$ and $\langle B,\vee\rangle$ are receptively meet and join semi-lattices, and the ordering relation $\leq$ on B is defined as: $x\leq y$ iff $x\wedge y=x$ iff $x\vee y=y$.

If $\langle B,\wedge,\vee\rangle$ is a lattice, then 0 is ***the least element*** of B iff
$0\in B$
for any x in B, $0\leq x$,
$0\wedge x=0$ and
$0\vee x=x$.
If $\langle B,\wedge,\vee\rangle$ is a lattice, then 0 is ***the greatest element*** of B iff
$1\in B$
for any x in B, $x\leq 1$,
$1\wedge x=x$ and
$1\vee x=1$.

If $\langle B,\leq\rangle$ is a partially ordered structure and x and y are in B, then
the greatest lower bound (briefly, ***glb***) of $\{x,y\}$ (if it exists) is the $z\in B$ such that
$z\leq x$ and $z\leq y$
for any w in B if $w\leq x$ and $w\leq y$, then $w\leq z$.
If $\langle B,\leq\rangle$ is a partially ordered structure and x and y are in B, then
the least upper bound (briefly, ***lub***) of $\{x,y\}$ (if it exists) is the $z\in B$ such that
$x\leq z$ and $y\leq z$
for any w in B if $x\leq w$ and $y\leq w$, then $z\leq w$.

A lattice $\langle B,\wedge,\vee\rangle$ is **distributive** iff
$x\vee(y\wedge z)=(x\vee y)\wedge(x\vee z)$, and
$x\wedge(y\vee z)=(x\wedge y)\vee(x\wedge z)$.

If $\langle B,\wedge,\vee,0,1\rangle$ is a structure such that $\langle B,\wedge,\vee\rangle$ is a lattice and 0 and 1 are respectively its least and greatest elements, then $-$ is a (***unique***) ***complementation operation*** on the structure iff
$-$ is a one-place operation on B
for any $x\in B$, $-x\in B$
$x\wedge -x=1$
$x\vee -x=0$
$-x=x$
$-1=0$
$-0=1$
$-(x\wedge y)=-x\vee -y$
$\sim(x\vee y)=\sim x\wedge\sim y$
$x\leq y$ iff $-x\wedge y=0$ iff $-y\leq -x$ iff $-x\vee y=1$

A structure $\langle B,\wedge,\vee,-,0,1\rangle$ is a ***Boolean algebra*** iff
$\langle B,\wedge,\vee\rangle$ is a lattice
$\langle B,\wedge,\vee\rangle$ is distributive
0 and 1 are respectively the least and greatest elements of $\langle B,\wedge,\vee\rangle$
$-$ is a complementation operation on $\langle B,\wedge,\vee,0,1\rangle$

Metatheorem 4-2

If $<B,\wedge,\vee>$ is a lattice, then $<B, \leq>$ is a partial ordering.

Metatheorem 4-3

If $\leq$ is a partial ordering on a set B and if for any x and y in B, the glb$\{x,y\}$ and the lub$\{x,y\}$ exist and are in B, and if $\wedge$ and $\vee$ are binary operations on B defined as follows

$$x\wedge y = \text{glb}\{x,y\}, \text{ and } x\vee y = \text{lub}\{x,y\},$$

then the structure $<B,\wedge,\vee>$ is a lattice with ordering relation $\leq$.

Metatheorem 4-4

The classical structure of truth-values is a Boolean algebra.

E. Sameness of Structure

One of the most important ideas in algebra is sameness of structure. Two teacups from the same set and two pennies have the same structure. So too do two twins. In these cases the structures match very closely. But family members and even members of the same species have some features of structure in common. More abstractly, the reason maps work is that there is a similarity of structure between geographical features in the world and the symbols on the map that represent them. Blue-prints work for this reason too. Mathematically this sameness is explained by saying that there is a mapping from the entities of one structure into the entities of a second in such a way that the mapping "preserves structure." Informally, if we have two structures and entity x_1 in the first that "corresponds" to an entity x_2 in the second, we may call x_2 ***the representative*** of x_1. Often one structure may be more complex than the other, yet both exhibit some structural features in common. One way this happens occurs when elements of the more complex are "identified" or viewed as a unit in the second. This happens, for example, in our representative democracy in which all the citizens in an election district are represented by a single individual in Congress. Thus for a "similarity of structure" to obtain we require as a minimum that each entity of one structure corresponds to one and only one entity in the second. In mathematical terms, there is an ***into-function*** that assigns a ***value*** in the second structure to each ***argument*** in the first. If h is the mapping function, then $h(x_1)=x_2$. Here $h(x_1)$ is the representative of x_1. Such a mapping is called a ***homomorphism*** (from the Greek *homos* = *the same* and *morphos*=*structure*.)

Definition 4-9

Two structures $S=<A_1,...,A_k,R_1,...,R_l,f_1,...,f_m,O_1,...,O_n>$ and $S'=<A'_1,...,A'_k, R'_1,...,R'_l,f'_1,...,f'_m,O'_1,...,O'_n>$ are said to be of the ***same character*** or ***type*** iff

for each i=1,…,l, there is some n such that R_i and R'_i are both n-place relations, and

for each i=1,…,n, there is some n such that f_i and f'_i are both n-place functions.

Very often a discussion is clearly limited to structures of the same type. When this restriction is clear, it is tedious to keep mentioning it, and it is usually assumed without saying so explicitly.

Definition 4-10. Homomorphism.

If S=<A_1,...,A_k,R_1,...,R_i,f_1,...,f_m,O_1,...,O_n> and S′=<A'_1,...,A'_k,R'_1,...,R'_i, f'_1,...,f'_m,O'_1,...,O'_n> are structures of the same character, *h* is called a ***homomorphism from*** S to S′ iff *h* is a function from U{A_1,...,A_n} into U{A'_1,...,A'_n} such that

1. for each i=1,…,k, if $x \in A_i$, then $h(x_i) \in A'_i$;
2. for each i=1,…,l, $<x_1,...,x_n> \in R_i$ iff $<h(x_1),...,h(x_n)> \in$ R′$_i$;
3. for each i=1,…,m, $h(f_i(x_1,...,x_n)) = f'_i(h(x_1),...,h(x_n))$;
4. for each i=1,…,m, $h(O_i)=O'_i$.

F. Sentential Semantics

One of the simplest and most elegant applications of algebraic ideas to logic is its use in formulating standard truth-functional semantics. We have already seen how to formulate syntactic structure and the structure of truth-values as algebras. It is now possible to formulate the idea of a "valuation," i.e. the traditional notion of an assignment of truth-values to formulas, as a homomorphism between the two structures. Many of the familiar semantic properties of classical valuations then follow directly as properties of morphisms.

Let us begin by restating the standard definition of a valuation in non-algebraic terms.

Definition 4-11. The Semantics for Sentential Logic

A (***classical***) ***valuation*** for the set $\mathbf{F_{SL}}$ of ***formulas*** of an **SL language** generated by $\mathbf{AF_{SL}}$ is any assignment V of a truth-values T or F to the formulas in $\mathbf{F_{SL}}$ that meets the following conditions:

V assigns to every atomic sentence in $\mathbf{AF_{SL}}$ either T or F;

V assigns to negations, conjunctions, disjunctions, conditionals and biconditionals the truth-value calculated by the truth-tables from the truth-values that $\Im$ assigns to its parts.

The formula *P* is a ***tautology*** (abbreviated $\vDash_{\mathbf{SL}}P$) iff for all V, V assigns T to *P*.

The argument from P_1,...P_n,... to Q is ***valid*** (abbreviated, P_1,...P_n,...$\vDash_{\mathbf{SL}}Q$) iff for any V, if V assigns T to all of P_1,...P_n,..., then V assigns T to *Q*.

The algebraic formulation is short and sweet.

Definition 4-12. Sentential Semantics, Algebraic Formulation.

If **Syn**$_{\mathbf{SL}}$=< F$_{SL}$,R$_{\sim}$,R$_{\wedge}$,R$_{\vee}$,R$_{\rightarrow}$,R$_{\leftrightarrow}$> is a sentential syntax and **2**=<{T,F},∧,∨,→,↔,–> is the classical algebra of truth-values, then V is a ***classical valuation*** for **Syn**$_{\mathbf{SL}}$ iff V is a homomorphism from a **Syn**$_{\mathbf{SL}}$ to **2**.

We still have to define *tautology* and *validity*. The two are given equivalent definitions, which are stated by reference to the strucure. Before stating these new definitions, however, let us perform an abstraction.

The algebraic formulation of classical semantics is so elegant that it invites immediate generalization or "abstraction" from the fact that the semantics has merely two truth-values. Indeed it is just such an abstraction that was made by Lukasiewicz and other Polish logicians in the 1920's and which has provided the standard framework for the development of valuational semantics ever since. In its abstract version valuational semantics is a special sort of algebraic structure called a logical matrix. This is very like the structure for the two-valued classical truth-values just employed, but in addition it singles out as a designated set a subset of truth-values, called the ***designated values***, that are those used to defining tautology and validity.

Definition 4-13. Sentential Semantics Formulated in Terms of Logical Matrices

A ***logical matrix*** is any structure =M=<U,D,∧,∨,→,↔,–> such that

U is non-empty (usually a subset of the real numbers)
D (the set of ***designated values***) is a non-empty subset of U
∧,∨,→,↔ are binary relations on U
– is a unary operation on U.

The set of valuations Val$_{M}$ (relative to **Syn**$_{\mathbf{SL}}$) is the set of all homomorphisms V from **Syn**$_{\mathbf{SL}}$ to M.

A ***sentential matrix language*** SL is any < **Syn**$_{\mathbf{SL}}$**,**Val$_{M}$>.

The argument from P_1,...P_n,... to Q is ***valid*** in SL (abbreviated, P_1,...P_n,...$\models_{\mathbf{SL}}Q$) iff for any V, if V(P_1)∈D,..., V(P_n)∈D, then V(Q)∈D.

The formula P is a ***tautology*** in SL (abbreviated $\models_{\mathbf{SL}}P$) iff for all V, V(P)∈D.

Much of our discussion of the intensions will be formulated in terms of logical matrices.

G. Sameness of Kind

Sameness is one of the "great ideas." Aristotle was the first to clearly distinguish ***numerical identity*** (he coined the term) from other sorts of sameness. Algebra has a nice set of concepts that make all the relevant distinctions. It also provides a battery of useful collateral ideas. Let us first distinguish numerical identity. This is the idea treated in "first-order logic with identity." It is given = as its own logical symbol in the syntax, and special *ad hoc* clauses in the definition of a semantic interpretation specifying that the symbol stands for the identity relation on the domain. This identity relation is understood to be a theoretical primitive (part of the stock of primitives that metatheory incorporates from set theory). It is the idea that is then summarized in the two semantic metatheorems whose syntactic versions are used to axiomatize truths of numerical identity:

$$\models_{\mathbf{FOL=}} x=x$$
$$\{x=y, P\} \models_{\mathbf{FOL=}} P[\mathrm{y//x}]$$

Sameness of kind has to do with classification into sets of individuals of the same "sort." One traditional way to discuss the idea is in terms of the sameness relation where this relation is understood to fold among more than one thing. Algebra specifies the properties that must hold of such a relation:

Definition 4-14. Equivalence Relation, Equivalence Class.

A binary relation $\equiv$ on a set A is said to be an ***equivalence relation*** on A iff $\equiv$ is reflexive, transitive and symmetric. The ***equivalence class*** of x under $\equiv$, briefly $[x]_{\equiv}$, is defined as $\{x| x\equiv x\}$.

Clearly numerical identity counts as an equivalence relation, but so do many other relations. Sameness of kind is also discussed in terms of sets. One way to do so is to put things into sets, as it were, manually, by means of set abstracts: we find and open sentences $P(x)$ that is true of all the "same" things. The " $P(x)$ " describes what they all have in common. We may go through everything there is and find such defining characteristics for "kinds" or "sorts" so that we can classify everything into non-overlapping, mutually exclusive sets $\{x| P_1(x)\},\ldots,\{x|P_n(x)\}$. Algebra provides a name for such a classification into "kinds:"

Definition 4-15. Partition.

A family F=$\{B_1,\ldots,B_n\}$ of sets is said to be a ***partition*** of a set A iff, A=U$\{B_1,\ldots,B_n\}$ and no two B_i and B_j overlap (i.e. for each i and j, $B_i \cap B_j = \varnothing$).

There is moreover a way to generate a partition from a sameness relations and vice versa.

Metatheorem 4-5

If a family F={B_1,...,B_n} of sets is a partition of a set A, then the binary relation ≡ on A is defined as follows: $x \equiv y$ iff for some i, $x \in A_i$ and $y \in A_i$ is an equivalence relation.

Metatheorem 4-6

The family of all equivalence classes $[x]_{\equiv}$ for all x in a given set A is a partition of A.

The set of all entities from the first structure that have the same representative are in a sense "the same:" they form an equivalence class. For example, the set of citizens represented by the same congressman is a equivalence class. One of direct consequences of these ideas is the fact that equivalence classes do not overlap and that they exhaust all the entities of the first structure.

Metatheorem 4-7

Let h be a homomorphism from S=<A_1,...,A_k,R_1,...,R_i,f_1,...,f_m,O_1,...,O_n> to S′=<A'_1,...,A'_k,R'_1,...,R'_i,f'_1,...,f'_m,O'_1,...,O'_n>, and let the binary relation $\equiv_h$ on C=U{A_1,...,A_n} be defined as follows:

$$x \equiv_h y \text{ iff } h(x)=h(y).$$

It follows that:

1. $\equiv_h$ is an equivalence relation on C.

Furthermore, if $[x]_h$, called ***the equivalence class*** of x under h is defined as {y| $y \equiv_h x$}, then it foolows that:

2. the family F of all equivalence classes, i.e. {$[x]_h$ | $x \in C$ }, is a partition of C.

H. Identity of Structure

If a structural representation is so tight that it exhausts the elements of the second structure in the sense that all of its elements are representatives of some entity in the first, then the representation function is said to be ***onto***. There are, for example, no voting members of Congress that do not represent some state. In Germany, however, where some members of Parliament are allotted to parties due to national voting percentages there are members that do not represent a specific district. We have seen, for example, that truth-value assignments (valuations) are onto homomorphisms from formulas onto the set {T,F} structures by the "truth-functions" specified in the truth-tables for the connectives.

In some instances the representation is so fine grained that no two entities of the first structure have the same representative. Such a mapping would be too cumbersome for Congress, but it is essential for social security numbers. Such mappings are said to be 1 to 1. Any mapping that is 1 to 1 and onto totally replicates the structure and entities of the first structure. It is called an ***isomorphism*** (from *isos*=*equal*).

Definition 4-16. Isomorphism**.**

If h is a homomorphism from S $=<A_1,...,A_k,R_1,...,R_i,f_1,...,f_m,O_1,...,O_n>$ to S $' =<A'_1,...,A'_k,R'_1,...,R'_i,f'_1,...,f'_m,O'_1,...,O'_n>$, then h is said to be an ***isomorphism*** from S to S $'$ if h is a 1-1 and onto mapping.

It follows from the definitions that given a homomorphism from a first structure to a second we can define a third structure made up of the equivalence classes of the first and this new structure can be made to have exactly the same structure as (be isomorphic to) the second. This new structure is called the quotient algebra.

Definition 4-17. Quotient Algebra.

If h is a homomorphism from S$=<A_1,...,A_k,R_1,...,R_i,f_1,...,f_m,O_1,...,O_n>$ to S$'=<A'_1,...,A'_k,R'_1,...,R'_i,f'_1,...,f'_m,O'_1,...,O'_n>$, then ***the quotient algebra*** for $<A_1,...,A_k,R_1,...,R_i,f_1,...,f_m,O_1,...,O_n>$ under h is S$''=<A''_1,...,A''_k,R''_1,...,R''_i,f''_1,...,f''_m,O''_1,...,O''_n>$ defined as follows:

given $x{\equiv_h}y$ iff $h(x)=h(y)$ and $[x]_h$ to be $\{y|\ y{\equiv_h}x\}$,
$A''_i = \{[x]_h \mid x\in A_i\}$
$<[x_1]_h,\ldots, [x_n]_h>\in R''_i$ iff $<x_1,...,x_n>\in R_i$
$f_i([x_1]_h,\ldots, [x_n]_h)= [f_i(x_1,...,x_n)]$
$O''=[O_i]_h$

Metatheorem 4-8

If h is a homomorphism from S$=<A_1,...,A_k,R_1,...,R_i,f_1,...,f_m,O_1,...,O_n>$ to
S $'=<A'_1,...,A'_k,R'_1,...,R_i,f'_1,...,f'_m,O'_1,...,O'_n>$,
then S is homomorphic to its quotient algebra S$''$ under h , and S$'$ is isomorphic to S$''$.

I. Congruence and Substitution

We are familiar in logic with various sorts of substitutability. One of the most familiar kind is the substitutability of material equivalents *salve veritate.* This phenomenon is a special case of a much more general one that results from the homomorphic nature of valuations.

Definition 4-18

The formula P is a ***tautology*** (abbreviated $\models_{\textbf{SL}}P$) iff for all V, V assigns T to P.

Definition 4-19

If S$=<A_1,...,A_k,R_1,...,R_i,f_1,...,f_m,O_1,...,O_n>$ is a structure with a binary relation $\equiv$ on C=U$\{A_1,...,A_n\}$, $\equiv$ is said to have ***the substitution property*** and to be a ***congruence relation*** iff

if $x_1{\equiv}y_1,..., x_n{\equiv}y_n$, then $<x_1,...,x_n>\in R_i$ iff $<y_1,...,y_n>\in R_i$, and if $x_1 \equiv y_1, \ldots, x_n \equiv y_n$, then $f_i(x_1,\ldots,x_n) \equiv f_i(y_1,\ldots,y_n)$.

Metatheorem 4-9

If h is a homomorphism from S=$<A_1,\ldots,A_k,R_1,\ldots,R_i,f_1,\ldots,f_m,O_1,\ldots,O_n>$ to S′=$<A'_1,\ldots,A'_k,R'_1,\ldots,R'_i,f'_1,\ldots,f'_m,O'_1,\ldots,O'_n>$, then the equivalence relation $\equiv_h$ is a congruence relation with the substitution property.

Corollary. (**Substitutability of Material Equivalents**.) If V is a classical valuation (i.e. a homomorphism from a sentential syntax $\mathbf{Syn_{SL}}$=$<F_{SL},R_{\sim},R_{\wedge},R_{\vee},R_{\rightarrow},R_{\leftrightarrow}>$ to the classical truth-value structure **2**=$<\{T,F\},\sim,\wedge,\vee,\rightarrow,\leftrightarrow>$ then the equivalence relation $\equiv_V$ is a congruence relation and has the substitution property.

J. Applications in Logic

Much of what we shall encounter in these sections are generalization from these basic results. The techniques will be to treat syntaxes from sentential logic to modal and epistemic logic to first-order logic as algebras defined on "strings" of symbols. Semantics is then conducted by defining structures, and then defining what are more familiarly known as valuations and interpretations as various sorts of morphisms over these structures. Various substitutability results then follow.

Definition 4-20. Subalgebra

If S=$<A_1,\ldots,A_k,R_1,\ldots,R_i,f_1,\ldots,f_m,O_1,\ldots,O_n>$ and S′=$<A'_1,\ldots,A'_k,R'_1,\ldots,R'_i,f'_1,\ldots,f'_m,O'_1,\ldots,O'_n>$ are structures of like character, then S is a ***subalgebra*** of S′ iff C=$\bigcup\{A_1,\ldots,A_k\} \subseteq \bigcup\{A'_1,\ldots,A'_k\}$ and each $R_1,\ldots,R_i,f_1,\ldots,f_m$ is the restriction repectively of $R'_1,\ldots,R'_i,f'_1,\ldots,f'_m$ to C. (When he structure is short and A'_i is empty, it is customary to delete it from the list detailing the structure if it is clear from the context to which set it corresponds.)

II. Matrix Semantics

A. Language and Entailment in the Abstract Structures

Let us begin our abstract study of logic by defining the core notions of syntax, semantics, and proof theory in there broadest algebraic senses. We shall assume at a minimum that the language in question contains sentences and that these are the syntactic units that make up arguments to be appraised for their validity.

i. Syntax

It is sufficient to define a syntax as a structure on "expressions" organized by rules of grammatical construction. In logic, "expressions" are normally understood to be finite strings built up by "concatenation" from a finite set of signs by means of the grammar rules understood as 1 to 1 ("uniquely decomposable") operations on finite strings. As customary, let Σ stand for the set of signs used to construct the syntax.

Definition 4-21. Syntax.

By a ***syntax*** Syn is meant a structure $<A_1,...,A_k,f_1,...,f_m>$ such that for some finite set Σ of signs, each f_i is a 1-1 function defined in terms of concatenation (the operation $^\frown$ on signs and strings) that maps some subset of Σ^* 1 to 1 into Σ^*, where Σ^* is the set of all finite strings of signs in Σ.

We assume that there is some A_i intended to represent sentences, and we use Sen as the preferred name of that A_i.

We let P and Q range over Sen, and X,Y and Z over subsets of Sen.

Example. Sentential Logic.

By a ***SL syntax*** is meant a structure $<\text{Sen},f_{\sim},f_{\wedge},f_{\vee},f_{\rightarrow}>$ such that there is some set ASen such that

1. ASen is an at most denumerable set (of "atomic sentences") constructed from some finite base of signs.
2. the operations are defined as follows:

$f_{\sim}(x)= \sim^\frown x$

$f_{\wedge}(x,y)= (^\frown x^\frown \wedge^\frown y^\frown)$

$f_{\vee}(x,y)= (^\frown x^\frown \vee^\frown y^\frown)$

$f_{\rightarrow}(x,y)= (^\frown x^\frown \rightarrow^\frown y^\frown)$

3. Sen is the least set (the set inductively defined) such that ASen $\subseteq$ Sen and Sen is closed under $f_{\sim},f_{\wedge},f_{\vee},f_{\rightarrow}$.

Substitution may also be defined for abstract syntaxes of this sort.

Definition 4-22. Substitution.

1. is a ***(uniform) substitution operation*** for Syn iff is a homomorphism from Syn into itself.
2. The notion is extended to sets as follows: $\sigma(X)=\{ \sigma(P)|P\in X\}$.
3. We let **Sub**Syn be the set of all substitution operations for Syn.

Definition 4-23. Subalgebra.

If $S=<A_1,...,A_k,R_1,...,R_i,f_1,...,f_m,O_1,...,O_n>$ and $S'=<A'_1,...,A'_k,R'_1,...,R'_i,f'_1,...,f'_m,O'_1,...,O'_n>$ are structures of like character, then S is a ***subalgebra*** of S′ iff $C=\bigcup\{A_1,...,A_k\}\subseteq \bigcup\{ A'_1,...,A'_k\}$ and each $R_1,...,R_i,f_1,...,f_m$ is the restriction repectively of $R'_1,...,R'_i,f'_1,...,f'_m$ to C.

Definition 4-24. Sentential Subalgebra.

The sentential subalgebra Syn|Sen of a syntax Syn is its subalgebra in which all categoeries of expressions other than Sen are empty.

Definition 4-25. Sentential Substitution.

1. σ is a ***(uniform) sentential substitution operation*** for Syn iff σ is a homomorphism from Syn|Sen into itself.
2. The notion is extended to sets as follows: $\sigma(X)=\{ \sigma(P)|P\in X\}$.
3. Let **Sub**Sen be the set of all sentential substitution operations for Syn.

Later we shall use $\vdash$ to represent the "deducibility" relation. We then use substitution to define "formal" inference patterns, and do so traditionally by means of tree diagrams that presuppose the working of substitution in a somewhat hidden way. Since we are stating relevant general concepts defined in terms of substitution, we shall explain this form of definition here.

.

Definition 4-26. "Formal" Relations (Inference Patterns) by Tree Diagrams.

We call $<X,P>$ a "deduction" and usually write it as $X\vdash P$. Let $\sigma(<X,P>)$ be $<\sigma(X), \sigma(P)>$.

By the tree R: $\dfrac{X_1 \vdash P_1>, \quad \ldots. \quad X_k \vdash P_k}{Y \vdash Q}$

we refer to (define) the following relation R on deductions

$R = \{<\sigma(<X_1,P_1>), \ldots., \sigma(<X_k,P_k>), \sigma(<Y,Q>)> \mid \sigma$ is a sentential substitution for Syn and $\sigma(<X_1,P_1>), \ldots., \sigma(<X_k,P_k>), \sigma(<Y,Q>)$ are deductions in Syn}.[55]

ii. Semantics

Characteristic of algebraic semantics is the interpretation of syntax by means of morphisms over structures of a character similar to that of syntax. It is also standard to interpret validity as some sort of "truth-preserving" relation holding between a set of premises and a conclusion. In general it is not necessary to specify the exact meaning of "truth", nor employ a single "truth-value" as the unique value preserved under valid inference.

Definition 4-27. Semantic Ideas.

By a ***semantic structure*** for Syn=$<A_1,\ldots,A_k,f_1,\ldots,f_m>$ for A_i=Sen is meant any structure Sem =$<B_1,\ldots,B_{k+1},f_1,\ldots,f_m>$ such that

1. $\cup\{B_1,\ldots,B_k\}\neq\varnothing$
2. $B_{k+1}\neq\varnothing$. (B_{k+1} is called the set of ***designated values****;* it is usually referred to as *D*. It is used below to define logical entailment.)
3. $<B_1,\ldots,B_k,f_1,\ldots,f_m>$ is of the same character as $<A_1,\ldots,A_k,f_1,\ldots,f_m>$.

If Syn=$<A_1,\ldots,A_k,f_1,\ldots,f_m>$ is a syntax and Sem=$<B_1,\ldots,B_{k+1},f_1,\ldots,f_m>$ is a semantic structure for Syn, then the set **I-**Sem of all ***semantic interpretations*** of Syn relative to Sem is the set of all homomorphisms from $<A_1,\ldots,A_k,f_1,\ldots,f_m>$ into $<B_1,\ldots,B_k,f_1,\ldots,f_m>$.

By a ***language*** L is meant any pair <Syn,F> such that F is family of semantic structures for Syn.

[55] Later in natural deduction theory we shall state more complex rules in which specific form and term stubstitution is specified in the tree diagram. For example, the rules $\forall$+ defined by the tree

$\dfrac{X\vdash P}{X\vdash\forall x P^c_x}$ refers to $\{<\blacklozenge(<X,P>), <\blacklozenge(X)_, \forall x\ \blacklozenge(P)^c_x>) \mid \blacklozenge$ is a sentential substitution for Syn and $\blacklozenge(<X,P>), <\blacklozenge(X)_, \forall x\ \blacklozenge(P)^c_x>$ are deductions in Syn}.

(In the unusual case in which <Syn,F> is a language in which F is a singleton set {S}, we identity F with S.)

Definition 4-28. Logical Ideas.

For any h in **I**-Sem, h is said to ***satisfy*** P iff $h(P)\in D$, and to ***satisfy*** X iff for all P in X, $h(P)\in D$.
X is said to ***semantically entail*** P in **I**-Sem (briefly, $X\vDash_{\text{Sem}}P$) iff, for any in h in **I**-Sem, h satisfies X only if it satisfies P.
X is ***satisfiable*** in **I**-Sem iff, for some h in **I**-Sem, h satisfies X,
X is ***unassailable*** in **I**-Sem iff, for any h in **I**-Sem, there is some $P\in X$, such that $h(P)\in D$.
X is said to ***semantically entail*** P in L=<Syn,F> (briefly, $X\vDash_{\text{L}}P$) iff for any semantic structure Sem of Syn in F, $X\vDash_{\text{Sem}}P$.
If $X\vDash_{\text{L}}P$, the argument from X to P is said to be ***valid***, and $\vDash_{\text{L}}$ is called ***entailment***.
$\vDash_{\text{L}}$ is ***compact*** or ***finitary*** iff $X\vDash_{\text{L}}P$ iff for some finite subset Y of X, $Y\vDash_{\text{L}}P$
We let $\vDash$ stand for either $\vDash_{\text{Sem}}$ or $\vDash_{\text{L}}$, and abbreviate $\varnothing\vDash P$ as $\vDash P$ and refer to P in this case as ***valid***. We abbreviate $\{P_1,\ldots,P_n\}\vDash Q$ as $P_1,\ldots,P_n\vDash Q$, and $X\cup Y\vDash P$ as $X,Y\vDash P$

iii. Proof Theory

Intuitively deduction is a matter of determining by reference to precise syntactic rules the sentences that are deducible from other sentences. The rules are not invented arbitrarily but rather are designed to provide a syntactic characterization of the more fundamental semantic relation of logical entailment. If we are able to "deduce" a sentence P from a set of premises X, we say that the deducibility relation holds between X and P. We begin by characterizing some of the relational properties of this very special relation.

Definition 4-29. An Abstract Characterization of Deducibility. Let $\vdash$ be a relation that holds between sets of sentences and sentences in**.**

$\vdash$ is ***reflexive*** iff $P\vdash P$
$\vdash$ is ***transitive*** iff $X\vdash P$ and $Y\cup\{P\}\vdash Q$, then $X\cup Y\vdash Q$
$\vdash$ is ***monotonic*** iff $X\vdash P$ and $X\subseteq Y$, then $Y\vdash P$
$\vdash$ is a ***consequence relation*** iff $\vdash$ is reflexive, transitive and monotonic.
$\vdash$ is ***finitely axiomatizable*** iff $X\vdash P$ only if for some Y, $Y\subseteq X$ and $Y\vdash P$.
$\vdash$ is ***closed under substitution*** iff
($X\vdash P$ only if, for any substitution operation $\sigma\in$ **Sub**Sen, $\sigma(X)\vdash\sigma(P)$).
$\vdash$ is a ***deducibility relation*** iff it is a finitely axiomatizable consequence relation closed under substitution.

Metatheorem 4 -10
For any Syn, $\models$ is a consequence relation.

Metatheorem 4-11

If $\vdash$ is a ***deducibility relation***, then ($X\vdash P$ iff, for any substitution operation $\sigma\in$ **Sub**Seṇ, $\sigma(X)\vdash\sigma(P)$).

Definition 4-30. Mutual Deducibilit

Definition 4-31.

$P \dashv\vdash Q$ iff, $P\vdash Q$ and $Q\vdash P$.

Metatheorem 4-12.

$\dashv\vdash$ is an equivalence relation on Sen.

iv. Provability

A notion narrower than deducibility is provability. A sentence is provable intuitively if its deduction does not depend on the truth of anything unproven. That is, *P* is provable from *X* iff if *X* is provable, so is *P.* But this account is not quite general enough. For *P* to be provable from *X* it is required that the proof be a matter of form. That is, if anything of the same form as *X* is provable, then anything of the same form as *P* should also be provable. We capture the idea of "sameness of form" by appeal to substitution.

Definition 4-32. The Provability Relation.

P is ***provable from*** *X* relative to a deducibility relation $\vdash$ (briefly, $X\Vdash P$), iff for all substitution operation $\sigma\in$ **Sub**Sen, (for all $Q\in X$, $\vdash\sigma(Q)$) only if $\vdash\sigma(P)$.

Metatheorem 4-13

$X\Vdash P$ only if $X\vdash P$, but not conversely.

Rules of Proof and Provability

The tree $\dfrac{\varnothing\vdash P}{\varnothing\vdash Q}$ aka $\dfrac{\vdash P}{\vdash Q}$ is the normal form used to stipulate a rule of proof.

Examples

1. Necessitation in Modal Logic. $\dfrac{\vdash P}{\vdash \Box P}$

2. Theoremhood in Classical Sentential Logic: $\dfrac{\vdash P}{\vdash \mathbf{Th}P}$

3. Theoremhood in PM: $\dfrac{\vdash P}{\vdash \mathscr{Th}(\underline{\mathbf{n}}_P)}$

Remark. Rule 2 (and Rule 1 if □=**Th**) is classical sound:

$$\varnothing \models_C P \quad \text{iff} \quad \varnothing \models_C (P \leftrightarrow (Q \vee {\sim} Q))$$

Rule 3 is not classically sound because neither $\mathscr{Th}$ nor $\underline{\mathbf{n}}_P$ have logically fixed referents. But if Val_{PM} is that subset of Val_C that satisfies the axioms of PM and $\models_{PM}$ is the restriction of $\models_C$ to Val_{PM}, then

$$\varnothing \models_{PM} P \quad \text{iff} \quad \varnothing \models_{PM} \mathscr{Th}(\underline{\mathbf{n}}_P)$$

Metatheorem 4-14

$$\frac{\varnothing \vdash P}{\varnothing \vdash Q} \quad \text{iff} \quad P \Vdash Q$$

v. Inductive Systems

The deducibility relations we shall be studying in these sections are primarily those of classical and intuitionistic logic. They both exhibit a good deal more structure than is captured in the abstract notion of a deducibility relation. They fall into the class of deducibility relations, familiar to students of elementary logic, that are characterizable in terms of axiom and natural deduction systems. In order to characterize this kind of deducibility relation, we begin by defining the concepts that abstract their special structural features. The ideas come from the theory of inductive sets.

Definition 4-33. Inductive System, Derivation and Proof.

An ***inductive system*** is any structure $<B,C,\{R_1,...,R_n\}>$ such that:

1. B (the set of *basic elements* of the system) and C (the set ***constructed*** *by* the system) are at most denumerable sets;
2. each R_i (a ***construction rule*** of the system) is a finite relation on $B \cup C$;
3. C is the least set X such that $B \subseteq X$ and, for any R_i, if R_i is an m+1-place relation, $<e_1,...,e_{m+1}> \in R$ and $<e_1,...,e_m> \in C$, then $e_{m+1} \in C$.

Relative to an at most denumerable set B, and a set of finitary relations $\{R_1,...,R_n\}$ defined for tuples in B, a ***derivation (tree)*** relative to B and $\{R_1,...,R_n\}$ is defined as any finite labeled tree Π such that:

1. every leaf node of Π is labeled by an element in B,
2. for any node n of Π with immediate predecessor nodes $m_1,...,m_k$,
   a. each m_i (for $i \leq k$) is labeled by some element e_i,
   b. n is labeled by some rule R_i such that
      $<e_1,...,e_k,e> \in R_i$.

If the leaf nodes of a deduction tree Π are labeled respectively $e_1,...,e_k$, its root node is labeled by e, and if $\{R_1,...,R_n\}=\{R \mid R$ is a finitary relation on Sen that labels some node of $\Pi\}$, we say Π is a **derivation (tree) of** e from $< e_1,...,e_k>$ relative to $\{R_1,...,R_n\}$. If in addition all the leaf nodes of Π are in B, then Π is called a ***proof (tree) of*** e from $<e_1,...,e_k>$ relative to B and $\{R_1,...,R_n\}$.

Metatheorem 4-15

e$\in C$ for an inductive system $<B,C,\{R_1,...,R_n\}>$ iff there is some proof tree of e relative to B and some subset of $\{R_1,...,R_n\}$.

vi. Axiom Systems

From an abstract perspective an axiom system is identified with an inductive system

Definition 4-34. Axiom System.

An ***axiom system*** for Syn=$<A_1,...,A_k,f_1,...,f_m>$ is any inductive system such that $<Ax,\vdash,\{R_1,...,R_n\}>$ such that Ax and $\vdash$ are subsets of Syn.

An axiom system $<Ax,\vdash,\{R_1,...,R_n>$ is ***finite*** and $\vdash$ is said to be ***finitely axiomatizable***, iff Ax is finite.

One weakness of analyzing $\vdash$ as a set is that in order to capture the more general idea of a deducibility, it must then be extended in some manner to a relation. In languages which have a semantic entailment relation that is compact and a conditional → that yields a "deduction theorem" (i.e. $\{P_1,...,P_n\}\vdash Q$ iff $\vdash(P_1\wedge...\wedge P_n)\rightarrow Q$), then the extension is possible. Conceptually the analysis is not very convincing because of its lack of generality: it depends on specific features of the syntax (on the right sort of connectives $\wedge$ and $\rightarrow$) and on compactness, a property not exhibited by some interesting logical systems.

Definition 4-35. Deducibilty in an Axiom System

The set $\vdash$ is extended to a relation as follows: $\{P_1,...,P_n\}\vdash Q$ iff $\vdash(P_1\wedge...\wedge P_n)\rightarrow Q,$

$X \vdash Q$ iff, there is some finite subset $\{P_1,...,P_n\}$ of X such that $P_1,...,P_n\vdash Q$**.**

Whether this $\vdash$ relation is a deducibility relation will depend on the properties of $\wedge$ and $\rightarrow$.

Example. The System C for Classical Sentence Logic.

C is defined as the inductive system $<\text{Ax}_\text{C},\vdash_\text{C},\{MP\}>$ such that :

1. Ax_C is any instance of the following three schemata:
   i. $P\rightarrow(Q\rightarrow P)$
   ii. $(P\rightarrow(Q\rightarrow R))\rightarrow((P\rightarrow Q)\rightarrow(P\rightarrow R))$
   iii. $(\sim P\rightarrow\sim Q)\rightarrow(Q\rightarrow P)$
2. *MP* (*modus ponens*) is $\{<P,Q,R>|\ Q=P\rightarrow R\}$

Metatheorem 4-16

The relation $\vdash_\text{C}$ is a deducibility relation.

vii. Natural Deduction Systems

Natural deduction systems too are inductive systems, but in this case the elements included in the inductive sets are "deductions," i.e. pairs <*X*,*P*> consisting of a set of premises and a conclusion that follows from them. The basic elements of the construction, therefore, must be a special selected set of deductions and the rules of construction must be rules that take deductions as arguments and yield deductions as values.

Definition 4-36. Natural Deduction Systems.

*B*y a ***deduction*** in Syn is meant any pair <*X*,*P*> such that $P\in$Sen and *X* is a finite subset of Syn. Here *X* is called the ***premise set*** of the deduction and *P* the ***conclusion****.*

An ***inference rule*** for Syn as any finitary relation on deductions in Syn. In addition a special set *BD* of deductions is distinguished, called the set of ***basic deductions***.

By a ***natural deduction system*** for Syn is meant any inductive system <*BD*,$\vdash$,*RL*> such that

1. *BD* is a distinguished set of deductions for Syn, and
2. *RL* is a set of derivation rules for Syn.

The inductively defined relation $\vdash$ is called the set of ***provable deductions*** for Syn relative to *BD* and *RL*.

We write $X\vdash P$ for $<X,P>\in\vdash$, and adopt the customary abbreviations:

$X,P\vdash Q$ means $X\cup\{P\}\vdash Q$
$P_1,\ldots,P_n\vdash Q$ means $\{P_1,\ldots,P_n\}\vdash Q$
$\vdash P$ means $\varnothing$**.** <*X*,*P*> is a provable deduction in <*BD*,$\vdash$,*RL*> iff there is some proof tree of Syn relative to *BD* and some subset of *RL* such that its root node is labeled by <*X*,*P*>.

Example. A Natural Deduction Systems C for the Classical Sentential Logic

C=$<\mathrm{BD}_C, \vdash_C, R_{\bot+}, R_{\bot-}, R_{\sim+}, R_{\sim-}, R_{\wedge+}, R_{\wedge-}, R_{\vee+}, R_{\vee-}, R_{\rightarrow+}, R_{\rightarrow-}, R_{Th}>$ is the inductive system such that

1. Let $<X,P>$ be a ***deduction*** iff $X \subseteq$Sen and $P \in$Sen. We adopt these abbreviations:

$X \vdash_C P$	for	$<X,P>$ is in $\vdash_C$;
$X,Y \vdash_C P$	for	$X \cup Y \vdash_C P$;
$X,P \vdash_C Q$	for	$X \cup \{P\} \vdash_C Q$;
$P_1,...,P_n \vdash_C Q$	for	$\{P_1,...,P_n\} \vdash_C Q$;
$\vdash_C P$	for	$\varnothing \vdash_C P$.
$\bot$	for	$P_1 \wedge \sim P_1$ (Here P_1 is the 1st atomic sentence.)

2. BD_C is the set of all deductions $<X,P>$ such that $P \in X$.
3. The rules in $\{R_{\bot+}, R_{\bot-}, R_{\sim+}, R_{\sim-}, R_{\wedge+}, R_{\wedge-}, R_{\vee+}, R_{\vee-}, R_{\rightarrow+}, R_{\rightarrow-}, R_{Th}\}$ are defined as follows:

	Introduction (+) Rules	***Elimination (-) Rules***
$\bot$	$\dfrac{X \vdash_C P \quad Y \vdash_C \sim P}{X,Y \vdash_C \bot}$	$\dfrac{X \vdash_C \bot}{X\text{-}\{\sim P\} \vdash_C P}$ (for $P \neq \sim Q$)
$\sim$	$\dfrac{X \vdash_C \bot}{X - \{P\} \vdash_C \sim P}$	$\dfrac{X \vdash \sim\sim P}{X \vdash_C P}$
$\wedge$	$\dfrac{X \vdash_C P \quad Y \vdash Q}{X \vdash_C P \wedge Q}$	$\dfrac{X \vdash_C P \wedge Q}{X \vdash_C P}$ $\dfrac{X \vdash_C P \wedge Q}{X \vdash_C Q}$
$\vee$	$\dfrac{X \vdash_C P}{X \vdash_C P \vee Q}$ $\dfrac{X \vdash_C P}{X \vdash_C P \vee Q}$	$\dfrac{X \vdash_C P \vee Q \quad Y \vdash_C R \quad Z \vdash_C R}{X, Y - \{P\}, Z - \{Q\} \vdash_C R}$
$\rightarrow$	$\dfrac{X \vdash_C P}{X - \{Q\} \vdash_C Q \rightarrow P}$	$\dfrac{X \vdash_C P \quad X \vdash_C P \rightarrow Q}{X \vdash_C Q}$

Thinning $\dfrac{X \vdash_C P}{X,Y \vdash_C P}$

We extend the notion of deduction to possibly infinite sets of premises X by saying $X \vdash_C Q$ relative to $\vdash_C$ iff, there is some finite subset $\{P_1,...,P_n\}$ of X such that $P_1,...,P_n \vdash_C Q$**.**

Metatheorem 4-17

The relation $\vdash_C$ is a deducibility relation.

In cases in which the notion of a uniform substitution σ is defined for Syn, it is customary to define a derivation rule R for Syn by a tree diagram.

Recall that by the tree R: $\dfrac{X_1 \vdash P_1>, \quad .... \quad X_k \vdash P_k}{Y \vdash Q}$

we refer to the relation

$R \quad = \quad \{<\sigma(<X_1,P_1>), ..., \sigma(<X_k,P_k>), \sigma(<Y,Q>)> \mid \sigma$ is a sentential substitution for Syn and $\sigma(<X_1,P_1>)$, ..., $\sigma\ (<X_k,P_k>)$, $\sigma(<Y,Q>)$ are all deductions in Syn}.

Since elegance and brevity are theoretical ideals of proof theory, finding the minimal set of rules necessary is often a goal. Some basic notions in terms of which systems are simplified and compared can now be defined.

Definition 4-37

A relation R is said to be ***definable*** relative to rules $R_1,...,R_n$ and is called a ***derived rule*** in $<BD,\vdash,RL>$, where $\{R_1,...,R_n\}\subseteq RL$, iff there is a derivation tree Π of d_{n+1} from $d_1,...,d_n$ relative to BD and $\{R_1,...,R_n\}$, and $R=\{<\sigma(d_1),..., \sigma(d_{n+1})>|\ \sigma$ is a substitution for Sen }.

A natural deduction system $<BD,\vdash,RL>$ is said to be ***reducible to*** a natural deduction system $<BD',\vdash',RL'>$ iff, $BD\subseteq BD'$ and every $R\in RL$ is a derivable rule in $<BD',\vdash',RL'>$.

Two systems are ***strictly equivalent*** iff they are mutually reducible. Let two systems $<BD,\vdash,RL>$ and $<BD',\vdash',RL'>$ be called ***constructively equivalent*** iff $\vdash = \vdash'$.

B. Logical Matrices for Sentential Logic.

One of the oldest and most productive branches of logic is the investigation of the semantic properties of sentential logic by means of structures known as logical matrices. Logical matrices are algebras of "truth-values" and the interpretations they spawn are homomorphisms between syntax and these structures.

Definition 4-38. Logical Matrices

A ***logical matrix*** for any SL syntax $\text{Syn}=<\text{Sen},f_{\sim},f_{\wedge},f_{\vee},f_{\rightarrow}>$ is any semantic structure $M=<U,D, g_{\sim},g_{\wedge},g_{\vee},g_{\rightarrow}>$ for Syn such that U and D are non-empty, and $D\subseteq U$.

Frequently U is some set of ordered numbers starting with 0, e.g. $\{0,1\}$, $\{0,{}^1/_2,1\}$, $\{0,1,\ldots,n\}$ starting with 0, in which case M is said to be ***m-valued*** where m is the cardinality of U.

A semantic interpretation relative to a logical matrix M is called a ***valuation*** of M, and the set of all semantic interpretations **I**-M of Syn relative to M is traditionally called the ***set of valuations*** of M, which we abbreviate Val_M. We let V range over Val_M. Clearly, $X\models_M P$ is well defined.

A ***matrix language*** is any language <Syn,F> such that F is a family of logical matrices.

It is customary to refer to both the series of syntactic operations $f_{\sim},f_{\wedge},f_{\vee},f_{\rightarrow}$ and the series of semantic operations $g_{\sim},g_{\wedge},g_{\vee},g_{\rightarrow}$ by $\sim,\wedge,\vee,\rightarrow$. In some contexts where it would be unclear which is meant, we shall distinguish one series from the other by the use of prime marks.

One of the most useful investigations in matrix semantics is the "representation" of one matrix in another. Such representations are used to simplify the semantics by replacing a broad set of valuations (and its characterization of entailment) with a narrower one, generated by a simpler matrix which is also characteristic of the entailment relation in question. We shall see several important examples of such representations in the course of these sections.

The relevant concept of representation is captured by the idea of homomorphism. Designated values play no role in the definition of valuations. As a result there is one sense of representation in which they are ignored, and a stricter sense in which they are not.

Definition 4-39. Matrix Morphisms

h is a (***matrix***) ***homomorphism*** (in the weak sense) from a logical matrix $M=\langle U,D, \sim,\wedge,\vee,\rightarrow\rangle$ to another matrix $M'=\langle U',D', \sim',\wedge',\vee',\rightarrow'\rangle$ (of the same character) iff *h* is a homomorphism from to $M=\langle U,\sim,\wedge,\vee,\rightarrow\rangle$ into $\langle U', \sim',\wedge',\vee',\rightarrow'\rangle$.

h is a ***strict*** (***matrix***) ***homomorphism*** from $M=\langle U,D, \sim,\wedge,\vee,\rightarrow\rangle$ to $M'=\langle U',D', \sim',\wedge',\vee',\rightarrow'\rangle$ (of the same character) iff *h* is a homomorphism and *h* ***preserves designation and non-designation*** in the sense that for any *x* in U,

$x\in D$, then $h(x)\in D'$, and
if $x\notin D$, then $h(x)\notin D'$,

We shall call these morphisms ***onto*** and ***1 to 1*** if *h* is an onto or 1 to 1 function respectively.

Note two additonal formulations that are equivalent to the condition that *h* preserves designation and non-designation:

1. for any *x* in U, $x\in D$ iff $h(x)\in D'$.
2. *h* maps D into D′, and U–D into U′–D′.

Notice also that if we interpret a syntax by a matrix M and there is a second matrix M′ to which M is homomorphic under *h*, then we can interpret the syntax by M'. For any sentence *P*, we assign it a value v(*P*) in M, and then using *h* we assign it to *h*(v(P)). We call composition the process of defining a third function by taking an argument's value under one function, turning it into the argument of a second function, and then calculating its value.

Definition 4 -40

If *f* and *g* are (one-place) functions, their ***composition*** $f \circ g$ is defined: $f \circ g(x) = g(f(x))$.

Metatheorem 4-18

If M=<U,D,~,∧,∨,→> is a logical matrix for Syn=<Sen,~,∧,∨,→> and *h* is a matrix homomorphism from M to M′, then $\{v \circ h \mid v \in \mathrm{Val}_{M}\} \subseteq \mathrm{Val}_{M'}$.

Proof. Consider an arbitrary $v \circ h$ such that $v \in \mathrm{Val}_{\tilde{M}}$We show it meets the conditions for membership in $\mathrm{Val}_{M'}$. If *P* is atomic, then *h* is defined for v(*P*) and the range of *h* is a subset of U′. Thus, $h(v(P)) \in U'$. For the molecular case consider an arbitrary complex sentence $O_i(P_1,...,P_n)$ such that O_i is the grammatical operation generating the sentence, and the operations in M and M′ corresponding to O_i are respectively g_i and g'_i. Then by the relevant definitions, $v \circ h(O_i(P_1,...,P_n))$ = $h(v(O_i(P_1,...,P_n)))$ =$hg_i(v(P_1),..., v(P_n)))$ = $g'_i(h(v(P_1),\ldots, h(v(P_n))$ = $g'_i(v \circ h(P_1),\ldots, v \circ h(P_n))$. Hence, $v \circ h \in \mathrm{Val}_{M'}$. **QED**

Metatheorem 4-19

If *h* is a strict matrix homomorphism from M=<U,D, ~,∧,∨,→> to M′=<U′,D′, ~′,∧′,∨′,→′>,
then $X \models_{M'} P$ only if $X \models_{M} P$.

(**Analysis.** Assume:

1. *h* is a strict matrix homomorphism from M=<U,D, ~,∧,∨,→> to M′=<U′,D′, ~′,∧′,∨′,→′>
2. $X \models_{M'} P$ i.e. for any $v' \in \mathrm{Val}_{M'}$, if for all *Q* in *X*, $v'(Q) \in D'$ then $v'(P) \in D'$
3. that v is arbitrary, that $v \in \mathrm{Val}_{M}$ and that for any *Q* in *X*, $v(Q) \in D$

Show: $v(P) \in D$.
The trick is to apply 1 to 3 and derive that $v \circ h(Q) \in D'$ that is,$h(v(Q)) \in D'$, for all $Q \in X$. Then apply Theorem 10, and deduce that $v \circ h \in \mathrm{Val}_{M'}$, and hence by 2, $v \circ h$ satisfies *P* in the relevant sense. Show then that *v* satisfies (in the relevant sense) *P.*

Metatheorem 4-20

If *h* is a strict matrix homomorphism from M=<U,D, ~,∧,∨,→> onto M′=<U′,D′, ~′,∧′,∨′,→′>, then $\{v \circ h \mid v \in \mathrm{Val}_{M}\} = \mathrm{Val}_{M'}$.

Proof. By theorem 10 all we need show is $\mathrm{Val}_{M} \subseteq \{v \circ h \mid v \in \mathrm{Val}_{M}\}$. Assume $v' \in \mathrm{Val}_{M'}$. We show that that $v' \in \{v \circ h \mid v \in \mathrm{Val}_{M}\}$. We construct a some *v*, such that $v' = v \circ h$ and $v \in \mathrm{Val}_{M}$. Let *P* be an atomic sentence. Since *h* is onto we know that whatever v′(*P*) is, let's call it *x*, there is some $y \in U$ such that *h*(*y*)=*x.* We define *v*(*P*) to be that *y.* We do so for each atomic sentence, and then project these values to molecular sentences by the operations in M. That is, we define v to be that $v \in \mathrm{Val}_{M}$ such that for any atomic sentence *P*, *h*(v(*P*)) = v′(*P*). We now show that $v' = v \circ h$, i.e. that for any sentence *Q*, v′(*Q*) =*H*=*h*(v(*Q*)). Proof is by induction. The atomic case is ture by the definition of *v*. For the molecular cases we assume the identity holds for the immediate parts of the sentence and show it is true for the whole. Consider the case of conjunction $R \wedge S$. Assume (as the induction hypo.) that v′(*R*)=*h*(v(*R*)) and v′(*S*)=*h*(v(*S*)). Now, $v'(R \wedge S)$ = [by membership of v′ in $\mathrm{Val}_{M'}$])$h(v(R)) \wedge h(v(S))$ = [since *h* is a homomorphism from M to M′])Ih$(v(R) \wedge v(S))$ = [since v is a homomorphism from Syn to M]$h(v(R \wedge S))$. The cases of the other connectives are similar.
QED

Metatheorem 4-21

If h is a strict matrix homomorphism from $M=<U,D, \sim,\wedge,\vee,\rightarrow>$ onto $M'=<U',D', \sim',\wedge',\vee',\rightarrow'>$, then $X\models_{M'}P$ iff $X\models_{M}P$.

i. Examples of Traditional Matrix Logics

Lukasiewicz and his colleagues were largely motivated by philosophical issues in developing matrix semantics, particularly their doubts about classical bivalence. Logical issues too are central. Both the matrix and the resulting entailment relation must be acceptable. Acceptability here is rather complex matter.

Acceptability is partly conceptual. The definitions offered by the theory must be "conceptually adequate." Roughly this is a requirement that the definitions conform with prior usage, both in ordinary language and in the earlier literature of logic and philosophy. For example, if matrix elements are intended to be "truth-values," then metatheorems concerning them should translate into plausible claims about truth. While the law of bivalence (every sentence is either true or false) may be doubted, the law of non-contradiction (no sentence and its negation can both be true) is less so. It is issues of this sort that are of concern to philosophers of language when they evaluate many-valued semantics.

Logical issues, however, are equally important. By their nature they tend to be the focus of logicians rather than philosophers. Logicians hone their intuitions about which inferences are valid. Doing so is a matter partly of common sense, partly of thinking about the meanings of the "logical terms" at play, and partly of tradition, logical tradition itself being one of the major determinants of the meaning of logical terms. Because classical two-valued logic has been the standard theory throughout this tradition, logical issue largely centers on how much, if at all, a matrix entailment relation departs from classical logic, and whether these departures are desirable. It has been proven to be very difficult to give a simple matrix semantics that is both conceptually plausible and yields an intuitively acceptable entailment relation.

A third criterion that is of less concern to the non-mathematical is elegance. Matrix semantics are very elegant indeed, and the goal of revising classical semantics using matrices has been a serious research enterprise, involving some of the best logicians, for almost eighty years. One of the large chapters of this story concerns the matrix characterization of intuitionistic logic, one of the century's major revisions of classical logic. We will take up intuitionistic semantics in detail later. At this point it will be instructive to illustrate the methods by citing some of the simpler and more famous many-valued theories.

Definition 4-41. Truth-Tables for Standard Matrices

The Classical
Bivalent
Matrix **C**

	~	∧	T	F	∨	T	F	→	T	F
T	F		T	F		T	T		T	F
F	T		F	F		T	F		T	T

Klenne's Weak
(Bochvar's Internal)
Matrix **KW**

	~	∧	T	F	N	∨	T	F	N	→	T	F	N
T	F		T	F	N		T	T	N		T	F	N
F	T		F	F	N		T	F	N		T	T	N
N	N		N	N	N		N	N	N		N	N	N

Klenne's Strong
Matrix **KS**

	~	∧	T	F	N	∨	T	F	N	→	T	F	N
T	F		T	F	N		T	T	T		T	F	N
F	T		F	F	F		T	F	N		T	T	T
N	N		T	F	N		T	T	N		T	N	N

Lukasiewicz'
3-valued
Matrix **L3**

	~	∧	T	F	N	∨	T	F	N	→	T	F	N
T	F		T	F	N		T	T	T		T	F	N
F	T		F	F	F		T	F	N		T	T	T
N	N		T	F	N		T	T	N		T	N	T

Jaskowski's
C^2-valued
Matrix

	~	∧	11	10	01	00	∨	11	10	01	00	→	11	10	01	00
11	00		11	10	01	00		11	11	11	11		11	10	01	00
10	00		10	10	00	00		11	10	11	10		11	11	01	01
01	00		01	00	01	00		11	11	01	01		11	10	11	10
00	00		10	10	00	00		11	10	11	10		11	11	11	11

None of these matrices with the exception of C^2 is classical. Whether the classical inferences they reject are in fact invalid is a further issue which we do not have time to go into here. Suffice it to say that none of these has proven very convincing.

Definition 4-42

We shall use M^D to refer to a matrix M with desiganted values D. It is also traditional to identify T with 1, 0 with F, N with ${}^1/_{2,}$ 2 with the set {0,1}, and in general n with $\{m \mid 0 \leq m < n\}$. As the universe for the matrix in question we take the set of all values appearing in the truth-table.

By **Ln**=<U,D, $\sim,\wedge,\vee,\rightarrow$> we mean the generalization of L3 in which D={1} and the operations conform to these rules:

$\sim x = 1\text{-}x$
$x \wedge y = \min\{x,y\}$
$x \vee y = \max\{x,y\}$
$x \rightarrow y = \min\{1, (1\text{-}x)+y\}$

For finite matrices $\mathbf{L}_n$ its domain U=$\{\frac{x}{n} \mid$ x is a natural number and $0 \leq x \leq n$ }. $\mathbf{L}_\omega$ is a limiting case in which U = ω = { x | x is a rational number and $0 \leq x \leq 1$ }. (In this limiting case what is important is the fact that U is countably infinite, *i.e.* that it have the cardinality of the set ω, the set of natural numbers. Hence U it may be identified with the rationals, which is equipollent to ω.) One can also set U equal to the continuum, *i.e.* the closed interval [0,1].

By $\mathbf{M}^\mathbf{n}$=<U^n,D^n, $\sim^n$,$\wedge^n$ $\vee^n$,$\rightarrow^n$>, we mean the generalization of M=<U,D, $\sim,\wedge,\vee,\rightarrow$> in which the operations f^n_i corresponding to conform to the following rules:

U^n and D^n are respectively the n-th Cartesian products of U and D, and
$f^n_i(<x_{1,1},...,x_{1,n}>,\ldots,<x_{m,1},...,x_{m,n}>)=<f_i(x_{1,1},...,x_{1,n}),\ldots, f_i(x_{m,1},...,x_{m,n})>$

Definition 4-43

Let C be the classical matrix <{T,F}{T},$\sim,\wedge,\vee,\rightarrow$>. Then, a matrix M=<U,D,$\sim',\wedge',\vee',\rightarrow'$> is ***normal*** iff, {T,F}$\subseteq$U, {T}$\subseteq$D, and for any x and y in {T,F}, $\sim x = \sim' x$, $\wedge(x,y)=\wedge'(x,y)$, $\vee(x,y)=\vee'(x,y)$, $\rightarrow(x,y)=\rightarrow'(x,y)$.

Metatheorem 4-22

$KW^{\{T\}}$, $KW^{\{T,N\}}$, $KS^{\{T\}}$, $KS^{\{T,N\}}$ are normal.

$\models_{KW\{T\}}$, $\models_{KW\{T,N\}}$, $\models_{KS\{T\}}$, $\models_{KS\{T,N\}}$, are proper subsets of $\Vdash_{C\{T\}}$.

$\{P \mid \models_{KW\{T\}} P\}$ and $\{P \mid \models_{KW\{T\}} P\}$ are empty.

$\models_{M^n D^n} = \models_{M}D$, and hence $\models_{C^n\{T\}^n} = \models_{C\{T\}}$,

ii. Lindenbaum Algebras

The first abstract results which we shall actually prove in which we use matrix semantics to characterize a proof theoretic idea will consist of ways to characterize the relatively weak provability relation $\Vdash$. They consist of constructing the relevant matrix from the syntax itself. Since these matrices are relate purely sytaxtic entities (sentences) they fall short of what the philosphers have traditionally thought of as a "world" or a "semantic interpretation." They are nevertheless excellent illustrations of algebraic ideas we have been introducing, so sucessful in fact that they may give philosphers pause.

We shall begin with an utterly trivial matrix, interpreting the syntax literally by the syntax itself. That is, we shall assign sentences to other sentences in a way that preserves syntactic structure. The sentence assigned to a whole will be that of like construction generated from those assigned to its parts. A representative, therefore will be a negation, conjunction, disjunction, etc. if the sentence it represents is, but the representative may have more structure because the atomic sentences of the original may be assigned to molecular sentences with internal structure. As designated elements let us use the set of provable sentences, i.e. the theorems of $\vdash$.

Definition 4-44

$\mathbf{Th}_{\vdash}=\{P \mid \Vdash P\}$

$[P]_{\vdash}=\{Q \mid Q \dashv\vdash P\}$ (Recall that $\dashv\vdash$ is an equivalence relation.)

M-Syn be $=<\mathrm{Sen},\mathbf{Th}_{\vdash},\sim,\wedge,\vee,\rightarrow>$

Metatheorem 4-23

For any $\Vdash$ for Syn$=<\mathrm{Sen},\sim,\wedge,\vee,\rightarrow>$, there exists a denumerable matrix M such that M

$X \Vdash P$ iff $X \vDash_{M} P$

Proof. For the matrix in question let us take Syn itself with all the theorems of $\vdash$ as designated elements. Let M-Syn be $=<\mathrm{Sen},\mathbf{Th}_{\vdash},\sim,\wedge,\vee,\rightarrow>$ where $\mathbf{Th}_{\vdash}=\{P \mid \Vdash P\}$. Observe that valuations over this matrix are just substitution relations: $\mathrm{Val}_{\text{M-Syn}} = \mathbf{Sub}\mathrm{Sen}$

Now, $X \vDash_{\text{M-Syn}} P$ iff $\forall\sigma\in\mathrm{Val}_{\text{M-Syn}}, [\forall Q\in X, \sigma(Q)\in\mathbf{Th}_{\vdash}] \Rightarrow \sigma(P)\in\mathbf{Th}_{\vdash}$

iff $\forall\sigma\in\mathrm{Val}_{\text{M-Syn}}, [\forall Q\in X, \vdash\sigma(Q))] \Rightarrow \vdash\sigma(P)$

iff $\forall\sigma\in \mathbf{Sub}\mathrm{Sen}, [\forall Q\in X, \vdash\sigma(Q))] \Rightarrow \vdash\sigma(P)$

iff $X \Vdash P$. **QED**

A more elegant syntactic matrix, called a Lindenbaum algebra, is that formed by the equivalence classes generated by $\vdash$. In such a structure the set of its logical equivalents "represent" a sentence. Such a class does indeed "stand proxy" for something like a "meaning" or "propositions," at least if we grant that "sameness of meaning" is at some level of abstraction the same as logical equivalence. If such a matrix is well-defined, it is in fact characteristic of the provability relation. In general, however, not all $\vdash$ relations generate such a structure. Though $\dashv\vdash$ is trivial an equivalence relation, to generate the structure in question it must also be a congruence relation (have the substitution property.

Definition 4-45

If $\dashv\vdash$ is a congruence relation on M-Syn=<Sen,$\mathbf{Th}_{\vdash}$,~,∧,∨,→>, then the quotient algebra determined by $\dashv\vdash$, namely

$$M_{\vdash}=<\{ [P]_{\vdash} \mid P\in\text{Sen}\},\{\mathbf{Th}_{\vdash}\},\sim,\wedge,\vee,\rightarrow>,$$

is called the ***Lindenbaum algebra*** for M-Syn.

Notice that corresponding to the set $\mathbf{Th}_{\vdash}$ of designated values in M-Syn is the set of 's designated values

$$\{\mathbf{Th}_{\vdash}\} = \{ [P]_{\vdash} \mid P\in\mathbf{Th}_{\vdash} \},$$

that this set contains one entity only that can serve as a designated value in $M_{\vdash}$ value, and that this single entity, namely $\mathbf{Th}_{\vdash}$, is itself a set, the set of $\vdash$ theorems.

of in Furthermore, the operation $[\ ]_{\vdash}$ preserves designation and non-designation: $P\in\mathbf{Th}_{\vdash}$ iff $[P]_{\vdash}\in \{\mathbf{Th}_{\vdash}\}$. The following theorems follow directly from (and illustrate how to apply) the general results we have already proven about congruence relations and strict homomorphisms between matrices.

Metatheorem 4-24

If $\dashv\vdash$ is a congruence relation on M-Syn, then the mapping $[\ ]_{\vdash}$ is a strict homomorphism from M-Syn to $M_{\vdash}$..

Metatheorem 4-25

If $M_{\vdash}$ exists, then $Val_{M_{\vdash}} = \{ \sigma \circ [\ ]_{\vdash} \mid \sigma\in \mathbf{Sub}\text{Sen}\}$

Metatheorem 4-26. (Lindenbaum)[56]

If $M_{\vdash}$ exists, then $X \Vdash P$ iff $X \models_{M_{\vdash}} P$

Metatheorem 4-27

If ≤ is the syntactic part-whole relation, then in general $[\ ]_{\vdash}$ is not an ≤-order preserving homomorphism: for some $\vdash$, P, and Q, it is not the case that ($[P]_{\vdash} \leq [Q]_{\vdash}$ iff $P\leq Q$).

[56] This theorem is not the more famous Lindenbaum Lemma which says that every consistent set may be extended to a maximally consistent set.

(Thought Question.) Let us call a puported inference relation ├ ***conceptually plausible*** if its definition consists of some principle (X├P iff …X…P…)) that is true of the inference relation ├$_C$ of classical logic. (Here "…X…P…" would state the definiting conditions the would have to hold for X and P in order for the relation ├ to hold.) Such a definition would be an "abstraction" from that of classical logic. Think up a definition for a conceptually plausible inference relation that is not transitive. Think up one that is not montonic. Think up one that fails for substitutions. What implication does failure of subsitutivity have for finite axiomatizability? What would these failures do to the ordinary notion of proof

III. BOOLEAN ALGEBRAS AND CLASSICAL LOGIC

A. Boolean Algebras

In this section we shall be concerned with what is probably the most important structures used in used in semantics. These are the Boolean algebras used in the interpretation of classical logic. As operations on sets they were studied by Boole, and as truth-functions by Pierce and Wittgenstein. They are basic to standard set theory and elementary logic, and as a class of algebras have many interesting properties that have inspired fruitful generalizations.

Definition 4-46. Boolean Algebra.

A structure $<B,\wedge,\vee,-,0,1>$ is a ***Boolean algebra*** iff it is a structure satisfying the following conditions. Let x, y and z be arbitrary members of B.

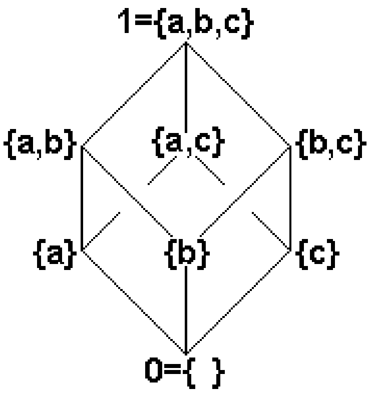
1. $<B,\wedge,\vee>$ is a lattice, i.e.
   L1. $x\wedge y=y\wedge x$; $x\vee y=y\vee x$
   L2. $(x\wedge y)\wedge z=x\wedge(y\wedge z)$; $(x\vee y)\vee z=x\vee(y\vee z)$;
   L3. $x\wedge x=x=x\vee x$;
   L4. $x\vee(x\wedge y)=x=x\wedge(x\vee y)$.
2. $<B,\leq>$ is a partially ordered structure, i.e. by definition $x\leq y \Leftrightarrow x\wedge y=x \Leftrightarrow x\vee y=y$ and
   P1. $x\leq x$;
   P2 $x\leq y$ & $y\leq z$.$\Rightarrow$ $x\leq z$;
   P3. $x\leq y$ & $y\leq x$.$\Rightarrow$ $x=y$.
3. $<B,\wedge,\vee>$ is distributive, i.e.
   D1. $x\vee(y\wedge z)=(x\vee y)\wedge(x\vee z)$;
   D2. $x\wedge(y\vee z)=(x\wedge y)\vee(x\wedge z)$.
4. 0 and 1 are respectively the least and greatest element of B in $<B,\wedge,\vee,0,1>$, i.e.
   G1. $0\leq x\leq 1$;
   G2. $1\wedge x=x$;
   G3. $1\vee x=1$;
   G4. $0\wedge x=0$;
   G5. $0\vee x=x$.
5. $-$ is a unique complementation operation on one-place operation on $<B,\wedge,\vee,-,0,1>$, i.e.
   B is closed under $-$ and
   C1. $x\wedge -x=0$
   C2. $x\vee -x=1$
   C3. $-x=x$, $-1=0$, $-0=1$;
   C4. $x\leq y \Leftrightarrow x\wedge -y=0 \Leftrightarrow -y\leq -x \Leftrightarrow -x\vee y=1$
   C5. $-(x\wedge y)=-x\vee -y$, $-(x\vee y)=-x\wedge -y$.

Metatheorem 4-28

$<B,\wedge,\vee,-,0,1>$ is a Boolean algebra iff $\wedge$ and $\vee$ are binary and $-$ a unary operation on B under which B is closed, $1,0\in B$ and

L1. $x\wedge y=y\wedge x$; $x\vee y=y\vee x$;
D1. $x\vee(y\wedge z)=(x\vee y)\wedge(x\vee z)$;
D2. $x\wedge(y\vee z)=(x\wedge y)\vee(x\wedge z)$;
C1. $x\wedge -x=1$
C2. $x\vee -x=0$
G2. $1\wedge x=x$;
G5. $0\vee x=x$;

Example. A three element Boolean Algebra



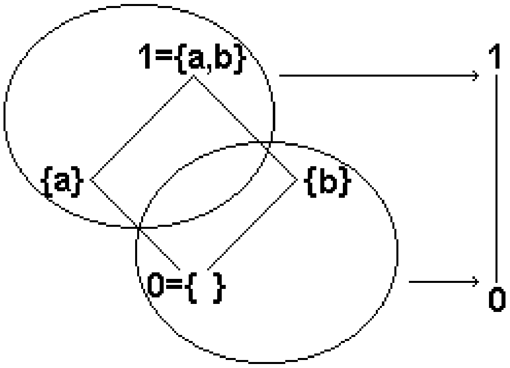
A Boolean Algebra of the Power set of {a,b,c}

We shall let **B**=$\langle B,\wedge,\vee,-,0,1\rangle$ range over Boolean algebras, distinguish one algebra from another by prime marks on its various components.

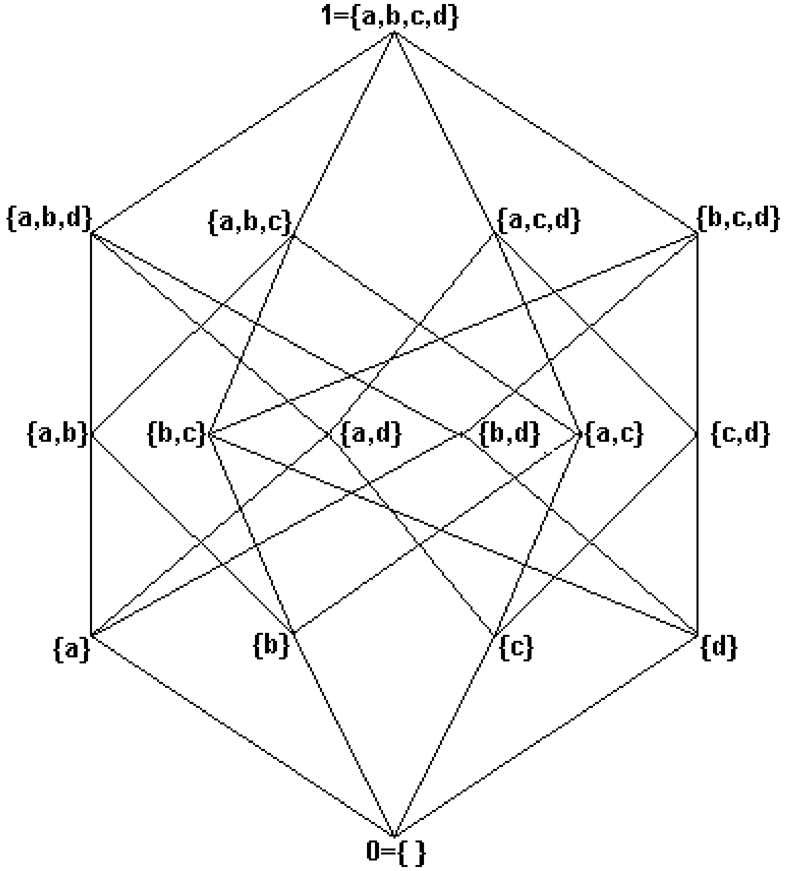
Metatheorem 4-29

Although any congruence relation for a Boolean Algebra $\langle B,\wedge,\vee,-,0,1\rangle$ has (by definition) the substitution property for $\wedge,\vee,-$ it does not in general have the substitution property for $\leq$. That is, there are some Boolean Algebras with congruence relation $\equiv$such that for some a,b,c in B, a$\equiv$b, c$\equiv$d, and a$\leq$c, yet not(b$\leq$c).

Consider the function *h* diagrammed below mapping one Boolean algebra to another and hence determining a congruence relation $\equiv_h$. That is *h* is defined: Here *h*(1)=1 *h*(a)=1, *h*(b)=0, *h*(0)=0.) Here $h(x\wedge y)=h(x)\wedge h(y)$ and $x\equiv_h y$ & $z\equiv_h w$.$\Rightarrow$ $x\wedge z\equiv_h y\wedge w$, and likewise for $\vee$. But, 0$\equiv_h$b & 1$\equiv_h$a & 0$\leq$1, yet not(b$\leq$0). **QED**



Example. A four element Boolean Algebra



A Boolean Algebra of the Power set of {a,b,c,d}

B. Filters, Ideals and the Binary Representation Theorem

A important subset of the universe of a Boolean algebra is the set of elements above *x*, or dually the elements below *x.* The former is called a *filter*, the latter an *ideal.* A maximal filter of *x* and dual maximal ideal of̃ *x* have the very nice property that they partition the algebra into just two equivalence classes that also determine a congruence relation. In other words, they proved a two element Boolean algebra with the "same structure" as the original. This binary structure "represents" the original and allows all Boolean algebras to be simplified into the structure on {0,1}. In the next section we shall apply this representation to the matrix interpretations of classical logic, where we shall find that the family of Boolean algebras is characteristic of classical deducibility, but by means of the representation theorem these may all be simplified to the familiar classical matrix on {T,F}.

Definition 4-47. Filters and Ideals.[57]

Let $\mathbf{B}=\langle B,\wedge,\vee,-,0,1\rangle$ be a Boolean algebra and $A\subseteq B$.

A is a ***filter*** on **B** iff

1. $\forall x,y\in B,\ x\in A\Rightarrow x\vee y\in A$, and
2. $\forall x,y\in B,\ x,y\in A\Rightarrow x\wedge y\in A$

(equivalently, iff $\forall x,y\in B,\ x,y\in A\Leftrightarrow x\wedge y\in A$).

A is an ***ideal*** on **B** iff

1. $\forall x,y\in B,\ x\in A\Rightarrow x\wedge y\in A$, and
2. $\forall x,y\in B,\ x,y\in A\Rightarrow x\vee y\in A$

(equivalently, iff $\forall x,y\in B,\ x,y\in A\Leftrightarrow x\vee y\in A$).

For any $x\in B$, by $[\boldsymbol{x}]\uparrow$ we mean $\{y|x\leq y\}$ and by $[\boldsymbol{x}]\downarrow$ we mean $\{y|y\leq x\}$

Metatheorem 4-30

For any Boolean algebra $\mathbf{B}=\langle B,\wedge,\vee,-,0,1\rangle$a and any $x\in B$,

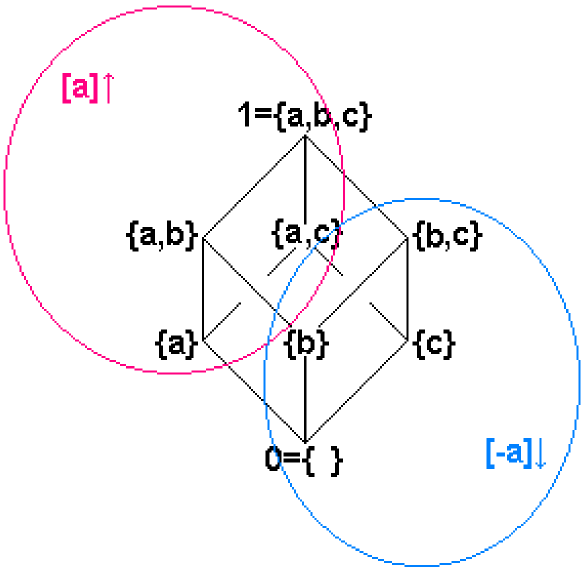
$[x]\uparrow$ is a filter on **B**, and
$[x]\downarrow$ is an ideal on **B**.

Definition 4-48

For any Boolean algebra $\mathbf{B}=\langle B,\wedge,\vee,-,0,1\rangle$a and any $x\in B$,

$[x]\uparrow$ is ***the prime*** (or ***principle***) ***filter on* B *relative to*** *x* and
$[x]\downarrow$ is ***the prime*** (or ***principle***) ***ideal on* B *relative to*** *x*.

Example. The prime filter of a and the prime ideal of its complement $\tilde{a}$={b,c}.



[57] A note on symbolism. We abbreviate the conjunction $x\in A$ & $y\in A$ as $x,y\in A$.

Definition 4-49. For any Boolean algebra **B**=$<B,\wedge,\vee,-,0,1>$, every filter/ideal of **B** is prime iff B is finite.

Definition 4-50

A filter/ideal of a Boolean algebra **B** is ***maximal*** iff

1. for some filter/ideal H, $B\subset H$, and
2. for any filter/ideal G, if there is a filter/ideal H such that $G\subset H$, then, if $B\subseteq G$, B=G (i.e. if G is a proper filter/ideal then B is not properly contained in it.)

Metatheorem 4-31

For any Boolean algebra **B**=$<B,\wedge,\vee,-,0,1>$,

1. F is a maximal filter/ideal of **B** iff, $\forall x\in F$, not($x\in F \Leftrightarrow -x\in F$).
2. F is a maximal filter/ideal of **B** iff, $\tilde{B}F$ is a maximal ideal/filter of **B**
3. F is a maximal ideal of **B** iff, the function *h* from B into its power set **P**(B) defined as follows: $\forall x\in B$,

   $h(x)=F$ if $x\in F$, and
   $h(x)=B-F$ if $x\notin F$

   is a homomorphism from **B** onto the Boolean

   $<\{F,B-F\},\cap,\cup,-,F,B-F>$

Definition 4-51

Let $<X,\leq>$ be a partially ordered structure.

A ***chain*** in $<X,\leq>$ is any non-empty subset *Y* if *X* such that if $x,y\in Y$ then $x\leq y$ or $y\leq x$.

An ***upper bound*** of a chain *Y* is $<X\leq>$ is a member *x* of *X* such that for all $y\in Y$, $y\leq x$.

An element *x* of is a ***maximal element*** of $<X,\leq>$ iff, for $x,y\in X$, $x\leq y \Rightarrow x=y$

Axiom. (Zorn's Lemma, equivalent to the **Axiom of Choice)**
If every chain of a partially ordered structure $<X,\leq>$ has an upper bound, then $<X,\leq>$ has a maximal element.

Metatheorem 4-32

For any Boolean algebra **B**=$<B,\wedge,\vee,-,0,1>$, any $x\in B$ and any ideal H of **B** that does not contain *x,* there exists a maximal ideal M of **B** such that $H\subseteq M$ and $x\notin M$.

Metatheorem 4-33

For any Boolean algebra **B**=$<B,\wedge,\vee,-,0,1>$, and any *x* and *y* of B, if not($y\leq x$), then there exists a maximal ideal M of **B** such that $x\in M$ and $y\notin M$.

Metatheorem 4-34

Every Boolean algebra is homomorphic to some two element Boolean algebra.

C. Boolean Interpretations of Classical Logic

Like any structure, a Boolean algebra if it has the same caharacter as a syntax may be used to fashion a logical matrix for the interpretation of the syntax. To do so we must specify in additiona a set of desiganted elements. Boolean algebras have the very nice property that the ordering relation within maximal filters replicate classical entailment. That is, if we sepcify a mximal filter as the set set of designated values, it will happen that when ever the premises of a classical valid argument are assigned values in the filter, the value assigned to the conclusion will also be in ther filter.

This replication, which is stated precisely in Theorem 9, is the semantic foundation that underlies the fact that Boolean algebras (with maximal filters as designated) are characteristic of classical deducibility. This "characterization" is spelled out in a soundness and completeness theorem. One appraoch is to adapt the Henkin compleness proof for sentential logic, which is familiar from elementary logic. The proof divides into one for soundness and one for completeness. The completeness proof remains unchanged, because M_C is itself a Boolean algebra, and hence the proof that any maximally consistent set is satifiable in an M_C-valuation automatically establishes that it is satisfiable in a Boolean valuation (with {1} as the maximal filter of designated elements.)

The proof of the soundness theorem needs to be adapted to Boolean algebras but the structure of the proof remains the same and the steps are just as straightforward as they are it is in the case of M_C. Soundness, recall, is estabilished by an induction that shows every provable deduction is valid. First every basic deduction is shown to be valid, Then, assuming (the induction hypothesis) that the arguments for a derivation rule are valid, it is shown that the value for the rule is valid. By the Boolean replication of $\models$ by ≤, these facts about validity translate into facts about ≤ in the Boolean structure, and going from ≤-facts about the inputs of a derivation rule to the relevant ≤-facts about the output becomes an exercise in applying the properties of the Boolean operations.

We are now ready for the definitions and theorems.

Definition 4-52. Boolean Matrices and Sentential Languages**.**

In this section we shall let Syn=<Sen,$f_{\sim}$,$f_{\wedge}$,$f_{\vee}$,$f_{\rightarrow}$> range over *SL* syntaxes .

By a ***Boolean matrix*** we mean any M=<B,F,∧,∨,–,0,1> such that

1. <B,∧,∨,–,0,1> is a Boolean albegra, and
2. F is a maximal filter on <B,∧,∨,–,0,1>.

The set of all Boolean matrices is **BM.**

By a ***Boolean*** (***sentential***) ***language*** is meant <Syn,**BM**> for any sentential syntax Syn.

By the ***classical matrix*** M_C we mean the Boolean matrix <{0,1},{1},∧,∨,–,0,1> in which the operations are defined as follows:

	~	∧	T	F	∨	T	F	→	T	F
T	F		T	F		T	T		T	F
F	T		F	F		T	F		T	T

We shall continue to use $\vdash_C$ to refer to the natrual deduction deducibility relation relation for classical logic defined in Section 2.

If *Y* is some finite subset {y_1,...,y_n} of B, then we shall use $\bigwedge\{f(x)\}_{x\in Y}$ to mean $f(y_1)\wedge\ldots\wedge f(y_n)$

Metatheorem 4-35

In any Boolean matrix M, $X\models_{M}P$ iff $\wedge\{V(Q)\}_{Q\in X} \leq V(P)$

Metatheorem 4-36

If L be a Boolean sentential langauge, then $X\vdash_{C}P$ iff $X\models_{\mathbf{L}}P$.

(Proof is as sketched above)

The ordinary completeness proof which states that classical deducibility is characterized by entailment over the two-valued matrix M_C is then a corollary of the this Boolean characterization theorem plus the bivalent representation theorem.

Metatheorem 4-37

If L be a Boolean sentential langauge, then $X\vdash_{C}P$ iff $X\models_{M}CP$.

Of course, there is a clear sense in which Theorem 11 is stronger than Theorem 10, and the stremlined Henkin completeness proof that estabilishes completeness for interpretations over just M_C is a more direct route to it. The weaker interpretaion, however, is interesting for two reasons. The first is conceptual, to which we now turn.

IV. Frege's Intensional Semantics

There has long been a tradition in logic and philosophy that logic and the "propositions" it expresses are not entities that exist in the common and garden material world but rather have a special status as intensional entities. Aristotle spoke about genera and species which are not the same as sets, and are indeed antitonic to them. Conceptually the genus G and differential D are "contained" in the species because they are used to define it: if we use the symbol + to indicate the process of "conceptual addition" operation, then we might symbolize the relationship as S=D+G, and hence G≤S. But the extensions of genera and species fall in a reverse ordering: Ext(S)=Ext(D)∩Ext(G), and hence S⊆G.[58]

This Aristotelian tradition lasted through the Middle Ages. The rationalists too spoke of logical truths as describing conceptual inclusion. But it is Frege's use in the nineteenth century of intensional entities as part of an informal semantics of indirect statements (*S believes that P*) that is the inspiration for the study of intensions in modern logic.

[58] A patially ordered structure <*X*,≤> is ***antitonic*** relative to ❑ to a partially ordered structure <*Y*,≤′> (and *h* is an ***antitone mapping*** from the first to the second) iff for any *x,y*∈*X*, *x*≤*y* iff *h*(*y*) ≤*h*(*x*).

Belief statements have the distinctive logical property that substitution of material equivalents and identities within the belief clause is invalid.
It is invalid to infer (3) from (1) and (2). (The example is Russell's)

(1) *George III believes Scott is Scott.*
(2) *Scott is the author of Waverley.*
(3) *George III believes Scott is the author of Waverley.*

Likewise {*S believes P*, $P \leftrightarrow Q$} does not entail *S believes P*. What is the explanation of this failure? The answer can be put in algebraic terms.

We know that substitutivity of material equivalents is a manifestation of homomorphic structure. In more traditional semantic terms this property is called the **compositionality of extension** or **reference**.

Principle 1. The Compositionality of Reference. The referent of a whole expression is determined in a rule-like way from the referents of its parts.

Algebraically, the principle asserts that there is a function g on "referents" that corresponds to a grammatical operation f, and that the reference relation Ext is a homomorphism mapping expressions to referents: $\text{Ext}(f(e_1,...,e_n)) = g(\text{Ext}(x_1),\ldots,\text{Ext}(x_n))$.[59] Fore example, if f is a grammatical operation that generates sentences then the referent of the sentences would be a truth value, i.e. $\text{Ext}(f(e_1,...,e_n)) \in \{T,F\}$. The compositionality of reference then would require that g would be a function mapping the semantic values of $\text{Ext}(e_1),...,\text{Ext}(e_n)$ to that very truth-value.

Belief statements, however, appear to be a counter-example to the principle. Let $f(a,P)=Bel_aP$. (We read Bel_aP as "a believes that P".) Let $\text{Ext}(a)$ be an object in the domain and $\text{Ext}(P)$ be a truth-value. Let g be the semantic "rule" corresponding to f. Then it is easy to show, it seems, that g is not a function.

On the one hand:

$$\begin{aligned} T &= \text{Ext}(Bel_{Georege\ III}Scott\ is\ Scott) \\ &= g(\text{Ext}(George\ III),\text{Ext}(Scott\ is\ Scot)) \\ &= g(\text{Ext}(George\ III),T) \end{aligned}$$

But on the other hand:

$$\begin{aligned} F &= \text{Ext}(Bel_{Georege\ III}Scott\ is\ the\ author\ of\ Waverley) \\ &= g(\text{Ext}(George\ III),\text{Ext}(Scott\ is\ the\ author\ of\ Waverley)) \\ &= g(\text{Ext}(George\ III),T) \end{aligned}$$

The semantic "rule" g for belief sentences is a relation not a function.

[59] The notation Ext for the reference function, and Int (below) for assigments of intensions is due to Richard Montague. Rudolf Carnap is responsiblle for fixing the terms of art *extension* and *intension* to these entities.

Frege's actually tries to avoid this conclusion and proposes an ingenious analysis that preserves functional compositionality of reference. He suggests that words, simple and complex, have intensions (he calls them *senses* but we might also call them "meanings") as well as extensions (which he calls referents). Senses too obey a rule of compositionality.

Principle 2. The Compositionality of Intensions. The intension of the whole expression is determined in a rule like manner from that of its parts.

Algebraically, Frege is postulating a structure of intensions homomorphic to grammatical structure. The ordinarily language terms we employ to name this homomorphism is the verb "express." We say a term or sentence "expresses" an idea or thought. Frege suggests that such locutions are informal ways of indicating the important semantic that holds between expressions and their intensions.

The algebras of intensions and extension, moreover, are related. Senses, Frege says, determine referents.

Principle 3. Sense Determines Reference. The extension of an expression is determined in a rule-like way from its intension.

Algebraically, the principle says that there is a homomorphism from intensional structure to extensional structure. There is no traditional name for this mapping, though some philosophers of language have called it the "reality function." Here we shall merely call it *R*.

Frege now applies these principles to explain the logical workings of belief. His idea briefly is that words which occur within the scope of the *belief*-predicate (or verb or operator or whatever it should be called) do not in those occurrences have there usual referents. Rather they refer to the intension that they usually "express." Terms in Frege's view, then, are systematically ambiguous. Most of the time, outside the scope of verbs like *believes*, they stand for their normal referent. Once they are understood this way, they do not violate the three basic principles of compositional semantics.

The details of Frege's theory, however, we shall postpone to the next section -- they are actually rather controversial. In the remaining part of this section, we shall focus on developing these ideas for the more ordinary example of sentential logic.

One of the lovely properties of Boolean interpretations of sentential logic is that they may be used to provide a detailed mathematical theory of the operations of extensions and intensions. Extensions are organized in the usual structure of truth-values, i.e. the classical matrix M_C. Intensional structure, however, is one that organizes the intensions of sentences. In modern logic these are usually called propositions. (Frege called them thoughts as well. In traditional logic a proposition is a sentence.) But what are propositions?

One interpretation is in terms of "possible worlds." A sentence's truth limits what is possible to those situations in which it is true. A more detailed, informative, meaningful sentence -- these terms are roughly synonymous -- restricts possibilities more than an less detailed, less informative, or less meaningful sentence. The "set of worlds in which a sentence *P* is true" is roughly the "information content" of *P*, and one such set is characteristic of each "proposition." Indeed, modern semantics employs just such world-sets as proxies for informal "senses." The world-sets may be put into structures and these structures made to exhibit all the structural features intuitively attributed to propositions. Moreover, to an algebrist if two sorts of entities exhibit exactly the same structure they are essentially identical. Naïve "sense" are reduced to or replaced by mathematically constructed proxies.

The intensional structures of world-sets appropriate for sentential logic is the Boolean algebra that has a universe consisting of a family of sets of worlds called *propositions*) and which relates these sets by the Boolean operations on sets.

Let us now state the formal version of Frege's sentential semantics. We will postulate a set of possible worlds, traditionally called K. These propositions will be world-sets, and are intuitively the "information content" of some sentence. Since propositions are world-sets, they are subsets of K. The universe of the intensional structure, therefore, is inhabited by subsets of K, and "the universe of intensional structure" is identical to the set of all of K's subsets. This set is called *the power set* of K. Since propositions are sets, the operations in the structure organaizing them may be seen as set theoretic intersections, union, and complimentation. In addition there is a special operation $\Rightarrow$ used to interpret the conditional, and defined in terms of complementation and union. An intensional interpretations that assign a proposition to each sentence is then simply a valuation (homorphism) from syntax to intensional structure. With the proposition that *P*, symbolized Int(*P*), in hand, it is possible to find *P*'s extension (truth-value) in a world: *P* is true in *k* iff $k\in$Int(*P*). The reality function, called R_k, that assigns to each *P* in *k* a truth-value is then easily defined: (Int(*P*))=T if $k\in$Int(*P*) and R_k(Int(*P*))=F if $k\notin$Int(*P*).

Frege's Semantics of Intensional and Extensional Structures for Classical Sentential Logic.

Definition 4-53

An ***intensional structure for sentential logic*** relative to a set K (called the set of possible worlds) is $<\mathbf{P}(K),\cap,\cup,\Rightarrow,-,\varnothing,K>$ such that **P**(K) (the power set of K) is the set of all subsets of K, and $\cap,\cup,-$ are the standard set theoretic operations on **P**(K) (here $x\Rightarrow y =_{\mathrm{def}} -x\vee y$) and $\varnothing$ is the empty set.

Definition 4-54

If M is an intensional structure, then Val_M relative to a sentential syntax Syn is called the set of ***intensional interpretations*** of Syn. We let **Int** range over this set.

Metatheorem 4-38. (Principle 2. Intensions are Compositional).

Any intentional interpretation Int of a sentential syntax Syn relative to an intensional structure I is a homomorphism from the syntax to the intensional structure.

Definition 4-55

We shall call ***the reality function*** relative to an intentional interpretation Int (over an intensional structure I) and a possible world *k* of I that function R_k from KxSen to {T,F} such that

$R(k,\mathrm{Int}(P))=T$ if $k\in\mathrm{Int}(P)$
$R(k,\mathrm{Int}(P))=F$ if $k\notin\mathrm{Int}(P)$

Instead of $R(k,\mathrm{Int}(P)$ we shall write $R_k(\mathrm{Int}(P))$, which is somewhat easier on the eyes. If we use $[\mathrm{Int}(P)]^{\nabla}$ to indicate the characteristic function of Int(*P*), i.e. the function that maps *k* to T if $k\in\mathrm{Int}(P)$ and *k* to F if $k\notin\mathrm{Int}(P)$, then R_k may be alternatively defined as:

$R_k(\mathrm{Int}(P))=T$ if $(\mathrm{Int}(P))(k)=T$
$R_k(\mathrm{Int}(P))=F$ if $(\mathrm{Int}(P))(k)=F$

(As we shall see, for some purposes it is even more convenient to think of this characteristic function as the proposition Int(*P*) intself.)

Metatheorem 4-39. (Principle 3. Intension Determines Extension.)

$\mathrm{Val}_M C = \{\mathrm{Int}\circ R_k \mid$ Int is an intentional interpretation on an intensional structure I of Syn, *k* is a possible world in K of I, and R_k is the reality function relative to Int.}

Proof Analysis. The proof requires establishing that any function v on Sen defined as

$v(P)) = R_k(\mathrm{Int}(P))$

qualifies for membership in the set of classical valuations $\mathrm{Val}_M C$ over M_C. This is done by showing that it assigns the right truth-values to atomic sentences and then assigns truth-values to molecular sentences in a manner that conforms to the classical truth-tables.

Second, it must be shown conversely that if $v \in \mathrm{Val}_M C$ then there is some Int and *k* of K such that $v= \mathrm{Int}\circ R_k$ *i.e.* for any *P*, $v(P)) = R_k(\mathrm{Int}(P))$. Select that Int that assigns to each atomic *P* a set containing the world some world *k* iff $v(P))=T$. It will then follow (by induction) that for all *Q* (atomic and complex), that

Metatheorem 4 -40. (Principle 1. Extensions are Compositional.)

Any v in $\mathrm{Val}_M C = \{\mathrm{Int}\circ R_k \mid$ Int is an intentional interpretation on an intensional interpretation I of Syn, *k* is a possible world in K of I, and R_k is the reality function relative to Int.} is a homoprphism from the syntactic structure Syn to M_C.

We shall finish by noting in addition that in this semantics one could interpret logical inference intensionally. Classical logic is the entailment relation determined by the class of Boolean matrices determined by intensional structures in which a maximal filter is selected as distinguished elements.

Likewise, according to Theorem 9 above, the ordering relation $\subseteq$ on propositions replicates entailment. That is, entailment is a kind of conceptual inclusion.

Definition 4-56

An ***intensional matrix for sentential logic*** (in the class **IMSL**) is any Boolean matrix $<\mathbf{P}(K),F,\cap.\cup.\Rightarrow, -,\varnothing,K>$ relative to the Boolean structure $<\mathbf{P}(K),\cap,\cup,\Rightarrow,-,\varnothing,K>$.

Metatheorem 4-41

If L=< Syn, **IMSL**> is a Boolean sentential language, then $X\vdash_C P$ iff $X\models_{\mathbf{IMSL}}P$.

Metatheorem 4-42

For any intensional matrix M and any Int in Val_M, $X\models_M P$ iff $\wedge\{Int(Q)\}_{Q\in X}\subseteq Int(P)$.

Corollary. For any intensional matrix M and any Int in Val_M,

$$\wedge\{Int(Q)\}_{Q\in X}\subseteq Int(P) \text{ iff } X\models_C P \text{ iff } X\vdash_C P$$

The set of possible world structures can be narrowed to a special one equally characteristic of classical entailment. This is the structure in which the worlds are themselves classical valuations over the bivalent matrix M_C. Indeed, classical valuations are "worlds" in the sense that they record a story: the set of sentences true in that world. For sentential logic, in other words, classical valuations themselves may serve adequately as the only notion needed for a Fregean intensional semantics .

Definition 4-57

The classical valuational structue for a sentential syntax Syn is

$$I_C=<P(Val_C), \cap,\cup,\Rightarrow,-,\varnothing, Val_C>$$

Let **B M I_C** be the set of all Boolean matricies relative to I_C.

Metatheorem 4-43

For any M∈B M I_C, any Int∈Val_M, and any v∈Int(P), $v(P)=R_v(P)$

Metatheorem 4-44

Let L=<Syn, B M I_C> for a sentential syntax Syn. For all M∈ B M I_C and any Int∈Val_M,

$$\wedge\{Int(Q)\}_{Q\in X}\subseteq Int(P) \text{ iff } X\models_C P$$

There exists in addition a special family of I_C interpretaions. This family alone is charcteristic of classical entailment and is so independently of the choice of designated values. That is, these interpretations differ only in their choice of designated values, but each alone is equally characteristic of classical entailment and is so in a manner that does not depend on its choice of desigated values.

The extra step then of introducing the matricies with its designated values in addition to I_C is for these interpretations is unnecessay. That is, we might simply identitfy them or more precisely defined a notion of intensional interpretaion directly from I_C that omits mention of designated values all together. Classsical entailment then proves to be simple set inclusion over valuations. This special interpretation, which we shall call Int_C, is that in which $\mathrm{Int}_\mathrm{C}(P)$ is the truth-set of P in classical bivalent semantics, i.e. $\mathrm{Val}_{\mathrm{MC}}(P)$.

Metatheorem 4-45

Let Int be an intentional interpretaion of I_C relative to some M in B M I_C such that

$$\mathrm{Int}(P) = \{\ \mathrm{v}{\in}\mathrm{Val}_\mathrm{M}\mathrm{C}(P)\ |\ \mathrm{v}(P){=}\mathrm{T}\ \}$$

Then,

$$\wedge\{\mathrm{Int}(Q)\}_{Q\in X} \subseteq \mathrm{Int}(P) \text{ iff } X \models_\mathrm{C} P$$

Definition 4-58

Let ***the preferred classical interpretation*** of a sentential syntax Syn be that homomorphism from Syn to I_C, which we shall call $\mathbf{Int_C}$, such that $\mathrm{Int}_\mathrm{C}(P) = \{\mathrm{v}{\in}\mathrm{Val}_\mathrm{M}\mathrm{C}(P)\ |\ \mathrm{v}(P){=}\mathrm{T}\ \}$.

Metatheorem 4-46

$\wedge\{\mathrm{Int}_\mathrm{C}(Q)\}_{Q\in X} \subseteq \mathrm{Int}_\mathrm{C}\ (P)$ iff $X \models_\mathrm{C} P$

V. Intensional Logic

A. The Idea of Intension in the History of Philosophy and Logic

In the last section we met briefly for the first time the main subject of this course, the concept of meaning as studied in modern logic. We saw that it was an idea introduced by Frege to explain the logic of sentences constructed from propositional attitude verbs like *believe*. Understanding what sort of problem Frege identified and what sort of theory he offers as a solution goes a long way towards explaining the dominant way in which logicians have conceived of the concept of meaning.

Meanings for Frege and the tradition that follows him are explanatory entities introduced as part of a "science" designed to explain some "data." The date in question are facts about particular logical inferences. The explanation is a general theory of inference. The theory incorporates various "laws" describing the behavior of meanings and these laws together with other parts of the theory entail the observed data.

The general shape of the over-all theory of inference is the familiar one of modern formal semantics. Though this sort of theory was only vaguely suggested in Frege's own writings it has since become standard. Validity is conceived of as some sort of truth-preserving relation among sentences in a formal syntax. The goal of the theory is to define this relation. The standard approach is to first define the notion of truth and to define truth as the correspondence of sentences with "the world." In the process the theory must met certain standards of adequacy.

Prominent among these standards is that the theory be mathematically rigorous. In practice this means that all its ideas must be well defined, and its assertions proved. The background theory usually assumed to get the process off the ground is set theory. In principle this should be some version of axiomatic set theory from which the paradoxes have be expunged, but in practice theorists use the naïve version (with an unrestricted axiom of comprehension) on the understanding that its results are "modulo axiomatization." That is, its results should properly be read as they would be written in an axiomatic version. (Proposition referring to paradoxical sets should be read as referring to "classes" or reformulated in favor of the open sentences that would naively define the set.)

A second criterion, which is responsible for much of the interest philosophers have in the theory, is that the definitions it offers of key concepts be conceptually plausible. Central among these are the concepts of "world," correspondence," and "truth." Frege's theory in addition employs "meanings" above and beyond standard entities in the world. The philosopher's ears immediate prick up. These are ideas they have been puzzling about for millennia. Any global "scientific" theory in which they play a role they find interesting indeed. A central concern in the theory is then that these key definitions conform, in the rough and ready way scientific terms always do, to previous usage, both in ordinary language and in earlier intellectual traditions.

Meanings are particularly intriguing. Philosophers have "postulated" queer entities above and beyond the common sense denizens of "the world" in order to explain the unknown. Since (at least) Plato they have done so to explain linguistic truth. Often the same entity has served multiple purposes. Plato's Forms are used in explanations in ontology, epistemology, ethics and semantics. So are Aristotle's genera and species and the universals of the Middle Ages. The ideas of the rationalists and empiricists are likewise put to various uses including semantics.

To see more clearly the link of Frege's idea of meaning to these earlier theories is necessary to be clear about the exact problems both Frege and the earlier theories were trying to solve, about the properties of the entities used to solve them.

Frege's problems and entities take a new direction. One tradition departs in a major way from Frege in that is conceives of logic as something mental and non-linguistic. The rationalists and Kant think of inference as conceptual inclusion or instances of mental laws. Even this tradition however is related to Frege. Frege is investigating the semantics of belief-sentences. These are sentences that describe "mental states." The so-called intentionalist tradition in logic is interested in part in these same mental states. Aristotle and the medievals described the mind (*animus*) as containing thoughts (*ratio*) that contained a mental content (*intentio*). The content determines the qualities ascribed to the thought's object but in such a way that the though itself does not have these qualities. The ideas of the rationalists and empiricists are similar. Brentano refers to the special features of such entities as *the intentional* and calls it the mark of the mental. Frege's meanings are the content of beliefs but in a linguistic fashion. They are the entities needed to be hypothesized (as referents of indirect statements in Frege's original theory) in order to explain the truth-conditions of belief-statements. Indeed he sometimes calls the senses of sentences *thoughts*. In is not a mistake then to see Frege's account as a recasting in linguistic terms of the intentionalist tradition. It is largely for this reason that Carnap coins the term *intension* (with an *s*) to refer to Frege's sentence meanings. The change in spelling indicates that the idea is recast into a new context, that of semantic theory.

It must be stressed that in taking this "linguistic turn" Frege imports a host of considerations not present in the intentionalist tradition.

One such concern is an explanation of the public inter-personal nature of linguistic communication. Frege's meanings, he says, are public in the sense that everybody understands the same meaning for the same sentence. Hence they are not part of an individual's mind. Not every intentionalist thinks intentions are private. In the Platonic tradition, for example, our mutual understanding consist of us both having a metal apprehension of one and the same "public" Idea, which is held to exist outside our minds in some public place like Platonic Heaven or God's soul. But many intentionalists are not concerted with language and conceive of intentions as parts of an individuals mind.

A second characteristic of Frege's linguistic approach is its deep link to explaining logical inference conceived of as a relation among sentences. He falls into an older tradition that includes Plato and Aristotle, the stoics, and (most) medieval logicians. The general approach has several key features:

1. views the syntactic form as determining a arguments validity,
2. its validity is conceived of as a truth-preserving relation among sentences, and
3. truth is defined as correspondence with the world.

Within the tradition individual theories differ depending on what semantic phenomenon they are tying to explain. The general strategy is to see if the problem may be solved by postulating some semantic entities with special properties tied to reference. Some of these entities are quite like Frege's and some not. Universals, for example (as in the semantics of Plato's that appeals to forms or of Aristotle's that employs secondary substances and qualities), are used to explain what predicates refer to, not as in Frege to explain inferences about belief-statements. (However, as mentioned above, Plato's ideas and Aristotle's impressions of forms on the soul are used as the objects of knowledge and belief states.)

Closer to Frege are the *lecta* (literately *the-what-is-read* or *the-what-is-meant*) which the stoics postulated as the "meanings" of sentences. These lecta have parts, and the lecton of the whole sentence appears to be a function of that of its parts. These complexes with a structure that mirrors that of syntax are also the objects of knowledge and belief. A similar view was arrived at independently in the Middle Ages by Ockham who posited a level of "mental language" between spoken language and the objects words stand for. Much like Frege, conceptualists like Ockham and Buridan explain the reference of words (the mediaeval cocept was called *supposition* and it was viewed as relative to a contexts of speech) as working through intermediate steps: words are paired by convention with "concepts" (terms of mental language) and concepts in turn naturally determine a referent in the world, Such mental language is also the object of knowledge.

Though neither the stoics nor Ockham develop a complete theory of inference, their accounts do share an important feature that is lacking in the intentions of the rationalists and empiricists: they posit the three levels of parallel homomorphic structure.

From this introduction it is possible understand the motivation for the algebraic approach taken in these sections. The algebra at once exhibits the mathematical rigor required of a formal theory and does so in a conceptually perspicuous way: it displays with clarity why validity is a truth-preserving relation and how truth as correspondence to the world falls out of a more general theory of language functioning as part of a three level combination of homomorphic structures.

B. Modal Operators and Cross-World Structure.

Frege observed that we cannot know from the extension of a sentence what the extensions of a belief sentence will be in which the sentence functions as an indirect statement. This general failure of "extensionality" (marked by the invalidity of the substitutivity of co-extensional parts) is a feature of other verbs that take indirect statements as complements *want, desire, hope, intend*) and of various sentential adverbs (*necessarily, possibly,*)

In this section a number of examples of intensional languages will be developed using the ideas of Carnap and Richard Montague. Montague succeed in capturing the algebraic properties of intensions by set theoretic proxies that are essential functions from possible worlds to the extensions of expressions in those worlds. The resulting theory is extremely elegant and quite abstract. Its abstractness moreover allows it to characterize the structural properties of "meanings" without making any claim about what sort of entities they might be proxies for in linguistic reality. Montague semantics, for example, has been embraced by those who think intensions are literally mental entities (parts of the brain), those who think they are abstract like mathematical entities, and those who think they are essential social phenomena.

Abstract Characterization of Fregean Intensional Semantics

Le us adopt the following set theoretic notation. If A and B are sets, then by A^B is meant the set of all functions from B into A. By 2 we mean the set of classical truth-values {T,F}={0,1}.
Let adopt the following syntactic conventions. Let Syn = $<A_1,...,A_m,f_1,...,f_n>$ be a syntax such that for some A_i =Sen. Let us use E_{Syn}, called the set of (***well-formed***) ***expressions*** of Syn, to stand for $\bigcup\{A_1,...,A_m\}$, and let e range over E_{Syn}. We shall let Syn range over syntaxes $<A_1,...,A_m,f_1,...,f_n>$.

Definition 4-59

By a ***Fregean intensional structure*** is meant a matrix structure I of Syn. That is I=$<B_1,...,B_m,h_1,...,h_n>$ such that $<B_1,...,B_m,h_1,...,h_n>$ off like character to Syn such that each h_i is a function.

Here B_{m+1} is the intended set of designated values used to define validity and each B_i is the set of possible "intensions" for expressions of category A_i. We let I=$<B_1,...,B_m,h_1,...,h_n>$ range over such structures.

Definition 4-60

If I is an intensional structure relative syntax Syn, then its set of matrix valuations Val_I is called the set of ***Fregean intensional interpretations*** of Syn. We let Int range over this set.

Metatheorem 4-47. (Compositionality of Intension: Intensions of Parts Determine Intension of the Whole).

Any Int of a syntax Syn relative to I is a matrix homomorphism from the syntax to the intensional structure, and $\equiv_{\text{Int}}$ is an equivalence relation with the substitution property.

(In bivalent languages an interpretation is a homomorphism in the unqualified sense from Syn to I.)

Definition 4-61

Let K be a non-empty set, called a set of ***possible worlds***, and let k range over K. Then, by a ***reality function relative to* I, Int, *and* K** we mean any function on domain E_{Syn}xK. We let R range over the reality functions relative to I and Int, and abbreviate $R(e,k)$ by $R_k(e)$.

Definition 4-62

By ***the extensional interpretation relative to* I, Int, K *and* R** is meant that function Ext on domain E_{Syn}xK defined as follows:

$$\text{Ext}_k(e)= R_k(\text{Int}(e)).$$

We let Ext stand for the extensional interpretation relative to I, Int, K, *and* R, and let R_k and Ext_k stand respectively for R and Ext relativized to k, i.e.

. R_k is that function f on E_{Syn} such that $f(e)= R_k(e)$.

Ext_k is that function g on E_{Syn} such that $g(e)=\text{Ext}_k(e)$.

Definition 4-63

By a ***Fregean (intensional) language*** is meant any matrix language <Syn,F> such that for each Fregean intensional structure (matrix) I of F, there is some non-empty set K (called ***the set of possible worlds of*** I) and some reality function R (called ***the reality function of*** I) such that any intensional interpretation (i.e. matrix valuation) Int in Val_I is defined relative to I and K, and R is defined relative to I, Int, and K.

In the remained of this section we shall let <Syn,F> range over Fregean intensional languages. We shall also use the set theoretic notation A^B as a name for the set of all functions from B into A.

Metatheorem 4-48. (Sense Determines Reference).

For any <Syn,F>, any $I \in F$ with possible world set K and reality function R, $\text{Ext}_k = \text{Int} \circ R_k$.

Definition 4-64.

By ***the Fregean extensional structure relative to*** <Syn,F> such that ***and to*** $I \in F$ ***with possible world set*** K ***and reality function*** R^* is meant $E=<C_1,...,C_{m+1},R_1,...,R_n>$, such that $C_i=\{ \text{Ext}_k(e)| k \in K \text{ and } e \in A_i\}$, and for $j=1,\ldots,n$, $R_j=\{<\text{Ext}_k(e_1),\ldots, \text{Ext}_k(e_n), \text{Ext}_k(e_{n+1})> \mid f_j(e_1,\ldots,e_n)=e_{n+1}\}$. Let $E=<C_1,...,C_{m+1},R_1,...,R_n>$ range over the extensional structures relative to <Syn,F>, I, K, and R^*.

Definition 4-65

1. If $e=f_i(e_1,\ldots,e_n)$ and there is some extensional structure of <Syn,F> such that is not a function then any occurrence of an expression within e is said to occur in an ***intensional context*** and e is said to be ***non-extensional*** and ***opaque***.
2. $E=<C_1,...,C_{m+1},R_1,...,R_n>$ (relative to <Syn,F>, I, K, and R^* is said to be ***extensional*** iff each R_j is a function for $j=1,\ldots,n$.
3. I (relative to <Syn,F> K, and R is ***extensional*** iff each E relative to <Syn,F>, I, K, and R^* is extensional.
4. The language <Syn,F> is ***extensional*** iff for any K and R^*, and for any $I \in F$ defined relative to K and R^*, I is extensional.

Metatheorem 4-49

The following are equivalent:

1. $E=<C_1,...,C_{m+1},R_1,...,R_n>$ relative to <Syn,F>, I, K, and R^* is extensional.
2. $E=<C_1,...,C_{m+1},R_1,...,R_n>$ is a logical matrix and Val_E for Syn is $\{ \text{Ext}_k \mid k \in K \}$.
3. For all $k \in K$, Ext_k is a matrix homomorphism from Syn to E.

Metatheorem 4-50. (Compositionality of Extension: Extension of Parts Determines Extension of the Whole).

In an extensional Fregean language, every extensional interpretation Ext_k, for any $k \in K$, is a matrix homomorphism from Syn to E, and $\equiv_{\text{Ext}_k}$ is an equivalence relation with the substitution property. (When $\equiv_{\text{Ext}_k}$ is restricted to Sen it is called material equivalence.)

Montague's Set Theoretic Characterization of Fregean Intensions

Definition 4-66

By a ***Montague structure relative*** **K** is meant a Fregean intensional structure interpretation I of Syn relative to a world structure such that $I=<B^K{}_1,...,B^K{}_{m+1},h_1,...,h_n>$.

Here $B_{m+1}{}^K$ is the intended set of designated values used to define validity, and each $B_i{}^K$ is the set of possible "intensions" for expressions of category A_i. We let I_M range over such structures. Intuitively, K is a set of possible worlds, and we shall call its power set P(K), the set of ***propositions***, and let π,ρ,θ range over P(K). We shall call the set $\{0,1\}^K$ of characteristic functions of propositions the set of ***sentential intensions***. If $f \in \{0,1\}^K$ is a characteristic function, let us use π_f to name the proposition of which it is the characteristic function. Conversely, if π is a proposition let π^c be its characteristic function.We let π^c, ρ^c, θ^c range over $\{0,1\}^K$.

Classical Logic and Classical Sentential Operators

Syntax: functions $f_{\sim}$,$f_{\wedge}$,$f_{\vee}$,$f_{\rightarrow}$ on signs previously defined.

Intensional Semantics. Let π and ρ range over P(K). Relative to a non-empty set K (of ***possible worlds***) $g_{\sim}$ is the 1-place function on $\{0,1\}^{K}$, and $g_{\wedge}$,$g_{\vee}$,$g_{\rightarrow}$ are the 2-place functions on $\{0,1\}^{K}x\{0,1\}^{K}$ such that

$$g_{\sim}(\pi^{c})=(-\pi^{c})$$
$$g_{\wedge}(\pi^{c}, \rho^{c}) = (\pi \cap \rho)^{c}$$
$$g_{\vee}(\pi^{c}, \rho^{c}) = (\pi \cup \rho)^{c}$$
$$g_{\rightarrow}(\pi^{c}, \rho^{c}) = (\pi \Rightarrow \rho)^{c}$$

Definition 4-67

L=<SL,F> is said to be ***classical for sentential Montague logic*** iff **SL**=<Sen,$f_{\sim}$,$f_{\wedge}$,$f_{\vee}$,$f_{\rightarrow}$> is a sentential syntax and F is the set of all logical matrices M such that for some non-empty set K, M=<2^{K},{ 2^{K} },$g_{\sim}$,$g_{\wedge}$,$g_{\vee}$,$g_{\rightarrow}$>.

Metatheorem 4-51. SL is extensional, and $X\models_{\mathbf{SL}}P$ iff $X\models_{\mathbf{C}}P$

(Alethic) Modal Operators

Syntax. $f_{\Box}$ and $f_{\Diamond}$ are defined as the 1-place operations on signs such that $f_{\Box}(P)=\Box P$ and $f_{\Diamond}(P)=\Diamond P$.

Intensional Sematics. W=<K,≤> such that K is non-empty and ≤ is a binary relation on K(***the alternativeness relation***) is said to be

- an **M *world structure*** iff ≤ on K is reflexive;
- a **B *world structure*** iff ≤ on K is reflexive and symmetric;
- a **S4 *world structure*** iff ≤ on K is reflexive and transitive;
- a **S5 *world structure*** iff ≤ on K is reflexive, symmetric, and transitive.

Relative to a worlds structure W=<K,≤>, $g_{\Box}$ and $g_{\Diamond}$ are defined as 1-place operations on $\{0,1\}^{K}$ into $\{0,1\}^{K}$ such that

$g_{\Box}(\pi^{\mathbf{c}})(k)$=T if for all k' such that $k\leq k'$, $\pi^{\mathbf{c}}(k')$=T, and
$g_{\Box}(\pi^{\mathbf{c}})(k)$=F if for some k' such that $k\leq k'$, $\pi^{\mathbf{c}}(k')\neq$T.
$g_{\Diamond}(\pi^{\mathbf{c}})(k)$=T if for some k' such that $k\leq k'$, $\pi^{\mathbf{c}}(k')$=T, and
$g_{\Diamond}(\pi^{\mathbf{c}})(k)$=F if for all k' such that $k\leq k'$, $\pi^{\mathbf{c}}(k')\neq$T.

Definition 4 -68
L=<Syn,F> is said to be a **M, B, S4,** or **S5** ***sentential modal (Montague) language*** respectively iff S**yn**=<Sen,$f_{\sim}$, $f_{\Box}$, $f_{\Diamond}$,$f_{\wedge}$,$f_{\vee}$,$f_{\rightarrow}$> and F is the set of all logical matrices M such that for some M, B, S4, or S5 world structure W=<K,≤> respectively, M=<2^{K},{K^{c}},$g_{\sim}$,$g_{\Box}$, $g_{\Diamond}$,$g_{\wedge}$,$g_{\vee}$,$g_{\rightarrow}$>. (We shall use the letters M, B, S4, and S5 to range over such languages.

Metatheorem 4-52

If L∈{M,B,S4,S5}, then L is non-extensional.

Metatheorem 4-53

$\Box P \dashv\vDash_{E} \sim\Diamond\sim P$, and $\Diamond P \dashv\vDash_{E} \sim\Box\sim P$.

Definition 4-69

If □ and ◊ produce valid arguments as stipulated in the consequences of the last theorem, they are called ***duals.*** More generally, of *E* and *E′* are sentential operators, then when ($EP \dashv\vDash_{E} \sim E'\sim P$, and $E'\mathrm{P} \dashv\vDash_{E} \sim E\sim P$), *E* and *E′* are called ***duals***.

Metatheorem 4-54

1. If L∈{M,B,S4,S5}, then $P \Vdash_{L} \Box P$ and $\Box P \vDash_{L} P$.
2. If L∈{M,B,S4}, then $\Box(P\rightarrow Q) \vDash_{L} \Box P\rightarrow\Box Q$.
3. If L∈{B,S5}, then $\Box P \vDash_{L} \Box\Diamond P$.
4. If L∈{S4,S5}, then $\Box P \vDash_{L} \Box\Box P$.

Definition 4-70

If □ and ◊ produce valid arguments as stipulated in the consequences of the previous theorem in the pattern appropriate to the languages M, B, S4,or S5, then □ and ◊ are called **M, B, S4,**or **S5** ***operators*** respectively. Likewise if a sentential operator *E* is replace by □ and *E′* by ◊ with the result that the replacements are M, B, S4,or S5 operators respectively then *E* is called an respectively an M, B, S4,or S5 ***necessity operator*** and *E′* a ***possibility operator*** for respectively M, B, S4,or S5.

Epistemic Descriptive Operators
Syntax. $f_{\mathbf{K}}$ and $f_{\mathbf{B}}$, are defined as the 1-place operations on signs such that $f_{K}(P)$=**K***P, and* $f_{\mathbf{B}}(P)$=**B***P*.
Intensional Sematics. W=<K,$\leq_{K}$,$\leq_{\mathbf{B}}$> is called an ***epistemic world structure*** iff K is a non-empty (of ***epistemically possible worlds***) and $\leq_{\mathbf{K}}$ and $\leq_{\mathbf{B}}$ are transitive binary relations on K (***the empistemic*** and ***doxastic alternativeness*** relations respectively) such that $\leq_{K} \subseteq \leq_{\mathbf{B}}$**.** In addition $\leq_{\mathbf{K}}$ is reflexive, and symmetric. Relative to an epistemic world structure W=<K,≤>, $g_{\mathbf{K}}$ and $g_{\mathbf{B}}$ are defined as 1-place operations on $\{0,1\}^{K}$ into $\{0,1\}^{K}$ such that

$g_{\mathbf{K}}(\pi^{\mathbf{c}})(k)$=T if for all k', $k\leq_{\mathbf{K}}k'$, $\pi^{\mathbf{c}}(k')$=T; $g_{\mathbf{K}}(\pi^{\mathbf{c}})(k)$=F if for some k', $k\leq_{\mathbf{K}}k'$, $\pi^{\mathbf{c}}(k')\neq$T.
$g_{\mathbf{B}}(\pi^{\mathbf{c}})(k)$=T if for all k', $k\leq_{\mathbf{B}}k'$, $\pi^{\mathbf{c}}(k')$=T; $g_{\mathbf{B}}(\pi^{\mathbf{c}})(k)$=F if for some k', $k\leq_{\mathbf{B}}k'$, $\pi^{\mathbf{c}}(k')\neq$T.

Definition 4-71.

L=<Syn,F> is said to be a ***sentential epistemic (Montague) language*** iff S**yn**=<Sen,$f_{\sim}$, $f_{\mathbf{K}}$, $f_{\mathbf{B}}$,$f_{\wedge}$,$f_{\vee}$,$f_{\rightarrow}$> and F is the set of all logical matrices M such that for some epistemic world structure W=<K,$\leq_{K}$,$\leq_{\mathbf{B}}$>, M=<2^{K},{ 2^{K} },$g_{\sim}$,$g_{\mathbf{K}}$, $g_{\mathbf{B}}$,$g_{\wedge}$,$g_{\vee}$,$g_{\rightarrow}$>.

Metatheorem 4-55
If L is a sentential epistemic language, then

1. L is non-extensional,
2. **K** is an S5 modal operator (hence $\mathbf{K}P \models_{\mathbf{L}} P$, and $\mathbf{K}P \models_{\mathbf{L}} \mathbf{KK}P$),
3. $\mathbf{K}P \models_{\mathbf{L}} \mathbf{B}P$
4. $\mathbf{B}P \models_{\mathbf{L}} \mathbf{BB}P$

Tense Operators
Idea. We let time branch towards the future and introduce two operators **H** and **F** for future tenses (**H***P* is read "it has to be that *P*" and **F***P* is "it will be that *P*") and two operators **P** and **W** for past tenses (**P***P* is read "it has to have been that *P*" and **F***P* is "it was the case that *P*").
Syntax. $f_{\mathbf{H}}$, $f_{\mathbf{F}}$, $f_{\mathbf{P}}$, and $f_{\mathbf{W}}$ are defined as the 1-place operations on signs such that , $f_{\mathbf{H}}(P)=\mathbf{H}P$, $f_{\mathbf{F}}(\mathbf{F}P)=P$, $f_{\mathbf{P}}(P)=\mathbf{P}P$, and $f_{W}(P)=\mathbf{W}P$.
Intensional Sematics. W=<K,≤> is a ***temporal world structure*** iff K is a non-empty set (of ***times***) and ≤ is reflexive and transitive binary relation (of ***temporal order***) on K. Then, relative to a temporal world structure W=<K,≤>, $g_{\mathbf{H}}$, $g_{\mathbf{F}}$, $g_{\mathbf{P}}$, and $g_{\mathbf{W}}$ are defined as 1-place operations on $\{0,1\}^K$ into $\{0,1\}^K$ such that

$g_{\mathbf{H}}(\pi^{\mathbf{c}})(k)$=T if for all k', $k \le k'$, $\pi^{\mathbf{c}}(k')$=T; $g_{\mathbf{H}}(\pi^{\mathbf{c}})(k)$=F if for some k', $k \le k'$, $\pi^{\mathbf{c}}(k') \neq$T.
$g_{\mathbf{F}}(\pi^{\mathbf{c}})(k)$=T if for some k', $k \le k'$, $\pi^{\mathbf{c}}(k')$=T, and $g_{\mathbf{F}}(\pi^{\mathbf{c}})(k)$=F if for all k', $k \le k'$, $\pi^{\mathbf{c}}(k') \neq$T.
$g_{\mathbf{P}}(\pi^{\mathbf{c}})(k)$=T if for all k', $k' \le k$, $\pi^{\mathbf{c}}(k')$=T; $g_{\mathbf{P}}(\pi^{\mathbf{c}})(k)$=F if for some k', $k' \le k$, $\pi^{\mathbf{c}}(k') \neq$T.
$g_{\mathbf{W}}(\pi^{\mathbf{c}})(k)$=T if for some k', $k' \le k$, $\pi^{\mathbf{c}}(k')$=T, and $g_{\mathbf{W}}(\pi^{\mathbf{c}})(k)$=F if for all k', $k' \le k$, $\pi^{\mathbf{c}}(k') \neq$T.

Definition 4-72

L=<Syn,F> is said to be a ***sentential tense (Montague) language*** iff S**yn**=<Sen,$f_{\sim}$,$f_{\mathbf{H}}$,$f_{\mathbf{F}}$,$f_{\mathbf{P}}$,$f_{\mathbf{W}}$,$f_{\wedge}$,$f_{\vee}$,$f_{\rightarrow}$> and F is the set of all logical matrices M such that for some epistemic world structure W=<K,$\le_{\mathrm{K}}$,$\le_{\mathbf{B}}$>, M=<2^K,{ 2^K },$g_{\sim}$,$g_{\mathbf{H}}$,$g_{\mathbf{F}}$,$g_{\mathbf{P}}$,$g_{\mathbf{W}}$,$g_{\wedge}$,$g_{\vee}$,$g_{\rightarrow}$>.

Metatheorem 4-56

If L is a sentential tense language, then

1. L is non-extensional,
2. **H** and **F** are duals and respectively S4 necessity and possibility operators,
3. **P** and **W** are duals and respectively S4 necessity and possibility operators,
4. $\mathbf{W}P \models_{\mathbf{L}} \mathbf{HW}P$ (the "necessity" of the past), but not($\mathbf{F}P \models_{\mathbf{L}} \mathbf{HF}P$)

Deontic Operators
Idea. Utilitarian moral choice is a consequentialist comparison among immediate temporal alternatives "situations." Let situations be places in a temporal tree structure opening towards the future and associate with each a utility value. A utilitarian says we ought to bring about that *P* (in symbols, **O***P*) f no matter what immediate situation ~*P* is true in there is a better one in which *P* is true. Similarly, it is morally permissible to bring about that *P* if it is not the case that for any situation in which *P* is true there is a better one in which ~*P* is true.
Syntax. $f_{\mathbf{O}}$ and $f_{\mathbf{P}}$, are defined as the 1-place operations on signs such that $f_{\mathrm{O}}(P)=\mathbf{O}P$, *and* $f_{\mathbf{Pr}}(P)=\mathbf{Pr}P$.
Intensional Sematics. W=<K,≤,U > is called a ***deontic choice structure*** iff K is a non-empty set (of ***choices***) and ≤ is binary relation (or ***temporal order***) on K such that ≤ determines an ascending tree structure with immediate successor relation << (hence ≤ is reflexive, transitive, and symmetric) such that each node has an immedicate successor, and U is a real valued function on K (called a ***utility function***). Relative to a deontic choice structure W=<K,≤,U >, $g_{\mathbf{O}}$ and $g_{\mathbf{Pr}}$ are defined as 1-place operations on $\{0,1\}^K$ into $\{0,1\}^K$ such that

$g_{\mathbf{O}}(\pi^{\mathbf{c}})(k)$=T, if for all k', $k \ll k'$ if $g_{\sim}(\pi^{\mathbf{c}})(k')$=T, there is some k'' such that $k \ll k''$, $\pi^{\mathbf{c}}(k'')$= T, and $\mathrm{U}(k') \le \mathrm{U}(k'')$, and $g_{\mathbf{O}}(\pi^{\mathbf{c}})(k)$=F otherwise.

$g_{\mathbf{Pr}}(\pi^{\mathbf{c}})(k)$=T, if for some k', k<<′, $\pi^{\mathbf{c}}(k')$=T, there is no k'' such that k<<k'', $g_{\sim}(\pi^{\mathbf{c}})(k'')$=T, and U($k'$) ≤ U($k''$) , and $g_{\mathbf{P}}(\pi^{\mathbf{c}})(k)$=F otherwise.

Definition 4-73

L=<Syn,F> is said to be a ***deontic sentential (Montague) language*** iff S**yn**=<Sen,$f_{\sim}$, $f_{\mathbf{O}}$, $f_{\mathbf{Pr}}$,$f_{\wedge}$,$f_{\vee}$,$f_{\rightarrow}$> and F is the set of all logical matrices M such that for some deontic world structure W=<K,≤,U >, M=<2^{K},{ 2^{K} },$g_{\sim}$,$g_{\mathbf{O}}$, $g_{\mathbf{Pr}}$,$g_{\wedge}$,$g_{\vee}$,$g_{\rightarrow}$>.

Metatheorem 4-57

If L is a deontic sentential language, then

1. L is non-extensional,
2. **O** and **Pr** are duals.

The Languages Combined

Definition 4-74

<K,≤,$\leq_{\mathbf{T}}$,$\leq_{K}$,$\leq_{\mathbf{B}}$,U > is a ***global sentential world structure*** iff K is a non-empty, <K,≤> is an S5 world structure; <K,$\leq_{K}$,$\leq_{\mathbf{B}}$> is an epistemic worlds structure such that $\leq_{K} \subseteq \leq$; <K,$\leq_{\mathbf{T}}$> is a temporal world structure such that $\leq_{\mathbf{T}} \subseteq \leq$; <K,$\leq_{\mathbf{T}}$,U > is a deontic world structure. Let $g_{\Box}$, $g_{\Diamond}$ be defined relative to <K,≤>; let $g_{\mathbf{K}}$, $g_{\mathbf{B}}$ be defined relative to <K,$\leq_{K}$,$\leq_{\mathbf{B}}$>; let $g_{\mathbf{H}}$,$g_{\mathbf{F}}$,$g_{\mathbf{P}}$,$g_{\mathbf{W}}$ be defined relative to <K,$\leq_{\mathbf{T}}$>; and let $g_{\mathbf{O}}$, $g_{\mathbf{Pr}}$ be defined relative to <K,$\leq_{\mathbf{T}}$,U >.

Definition 4-75

L=<Syn,F> is said to be a ***global sentential (Montague) language*** iff S**yn**=<Sen,$f_{\sim}$,$f_{\Box}$, $f_{\Diamond}$,$f_{\mathbf{O}}$,$f_{\mathbf{K}}$ $f_{\mathbf{B}}$,$f_{\mathbf{H}}$,$f_{\mathbf{F}}$,$f_{\mathbf{P}}$,$f_{\mathbf{W}}$,$f_{\mathbf{Pr}}$,$f_{\wedge}$,$f_{\vee}$,$f_{\rightarrow}$> and F is the set of all logical matrices M such that for some global sentential world structure <K,≤,$\leq_{\mathbf{T}}$,$\leq_{K}$,$\leq_{\mathbf{B}}$,U >,
and M=<2^{K},{ 2^{K} },$g_{\sim}$,$g_{\Box}$,$g_{\Diamond}$,$g_{\mathbf{O}}$,$g_{\mathbf{K}}$,$g_{\mathbf{B}}$,$g_{\mathbf{H}}$,$g_{\mathbf{F}}$,$g_{\mathbf{P}}$,$g_{\mathbf{W}}$,$g_{\mathbf{O}}$,$g_{\mathbf{Pr}}$,$g_{\wedge}$,$g_{\vee}$,$g_{\rightarrow}$>.

Metatheorem 4-58

If L is a global sentential language, then

1. L is non-extensional,
2. □$P \vDash_{\mathbf{L}}$ **K**$P \vDash_{\mathbf{L}} P \vDash_{\mathbf{L}} \Diamond P$, but not $\vDash_{\mathbf{L}}$**B**P,
3. □$P \vDash_{\mathbf{L}}$**O**$P \vDash_{\mathbf{L}} \Diamond$**F**$P \vDash_{\mathbf{L}} \Diamond P$ ("ought" implies "can")
4. not($_{\mathbf{L}}$**Pr**$P \vDash_{\mathbf{L}} \Diamond$**F**$P$)

VI. EXERCISES

A. Skills and Ideas

The material in this chapter has graduated to a more mathematical presentation. Here the relevant "skills" consist of being able to prove as metatheorems the claims made. Understanding the "ideas" consists likewise of grasping the definitions sufficiently to construct the proofs.

1. *Morphisms*. Exercise 1.

   a. Prove Metatheorem 4-7.
   b. Prove Metatheorem 4-8.
   c. Prove Metatheorem 4-9 (optional)

2. *Logical Matrices and Many-Valued Logic*. Exercise 2.

   a. Prove Metatheorem 4-19 .

   b. Prove $\models_{KW\{T\}}$ is a proper subsets of $\models_{C\{T\}}$. Show first that it is a subset by defining the right sort of homomorphism between the structures and then appealing to a previous theorem. Then show that it is a proper subset by finding some inference valid in the second that is not valid in the first. (This is part of Metatheorem 4-22.)

   c. Show $\{P| \models_{KW\{T\}}P\} = \varnothing$. (This is part of Metatheorem 4-22)

   d. Explain why theorem Metatheorem 4-24 , Metatheorem 4-25 , and Metatheorem 4-26. all follow directly from earlier results about matrices and valuations.

Fregean Intensional Semantics. Exercise 3.

   a. Prove Metatheorem 4-30 .
   b. Prove Metatheorem 4-35 .
   c. Prove Metatheorem 4 -40. .
   d. Prove Metatheorem 4-42 .

Intensional logic. Exercise 4.

   a. Prove Metatheorem 4-47, Metatheorem 4-48, and Metatheorem 4-49 using the definitions and facts previously proven about homomorphism and matrices interpretations.
   b. Prove Metatheorem 4-54, part 4.
   c. Prove Metatheorem 4-58 , part 3.

B. Theory

i. Evaluating Many-Valued and Modal Theories

Exercise 5. A major methodological problem in logic is articulating standards for critically comparing alternative logical theories. A major category of such theories is alternative many-valued logics. These have essentially the same syntax, treating the same "logical terms", but offer alternative semantics. These semantics involve new "truth-values" and new truth-tables, and these in turn affect logical entailment. Modal logics likewise form a family with slightly different semantics and resulting sets of valid arguments. It is not always obvious how alternatives like these should be evaluated because it is not obvious what properties it is possible to show hold of a formal language. The metatheorems in the text attempt to aid in this process. In some cases they prove that a logic has a specific feature. In others they make generalizations relating logics with specific properties. But they share the goal of enabling the critical comparison of alternatives. Explain in your own words some examples of how they do so. Do the metatheorems proven allow you to address any of the criteria C for evaluating alternative logics that you proposed in your exercises for Chapter 2? Do they ignore any?

ii. Extensionality and Intensionality.

The difference between extensional and intensional languages is often defined in the traditional philosophical literature in terms of the failure of the substitutivity of identities or of material equivalents to preserve truth. Algebraically, this fact may be expressed as the failure of the substitutivity property for "sameness of extension" relation among expressions. That is, "sameness of extension" fails to be a congruence relation relative to mappings (valuations or interpretations) from syntactic structures to semantics structures. This insight motivates much of the algebraic formulation found in Montague's intensional logic. In a short essay that you will be able to refer back to and understand later, try using the algebraic framework in your own words to explain why the failures of substitutivity *salve veritate* noted by Frege and others (for propositonal attitude constructions and modal operators) follow from the fact that the extensions of the parts of relevant expressions fail to determine in a "rule-like" way (*i.e.* by means of a genuine function) the extension of the whole.

References

Ashworth, E. J. "Traditional Logic." In *The Cambridge History of Renaissance Philosophy*, edited by Charles et al. Schmitt. Cambridge: Cambridge University Press, 1988.

Barwise, Jon and John Etchemendy. *Languge, Proof and Logic*. New York: Seven Bridges, 1999.

Bloc, Leonard and Pioto Borowik. *Many-Valued Logic*. Berlin: Springer-Verlag, 1992.

Carnap, Rudolf. *Meaning and Necessity*. Chicago: University of Chicago Press, 1947.

Chomsky, Noam. *Cartesian Linguistics*. New York: Harper and Row, 1966.

Church, Alonzo. *Introduction to Logic*. Vol. I. Princeton: Princeton University Press, 1956.

Copi, Irving M. *The Theroy of Logical Types*. London: Routledge and Kegan Paul, 1971.

———. *Symbolic Logic*. 5th ed. New York: Macmillian Publishing Co., 1979.

Davis, Philip J. and Reuben Hersh. *The Mathematical Experience*. Boston: Houghton Mifflin, 1981.

Davis, Martin and Elaine J. Weyuker. *Computability, Complexity and Languages*.. Orlando: Academic Press, 1983.

Davis, Martin. *Computability and Unsolvability*. New York: Raven, 1983.

De Long, Howard. *A Profile of Mathematical Logic*. Reading, MA: Addison-Weskey, 1970.

Doxiadis, Apostolos and Christos H. Papadimitriou, *Logicomix* (N. Y.: Bloombury, 2009).

Frege, Gottlob. *Grundgesetze der Arithmetik*. Vol. II. Jena: Verlag Hermann Pohle, 1903.

———. "Basic Laws of Arithmetic.", edited by Montgomery Furth. Berkeley: University of California Press, 1964 [1983].

———. "Begriffsshrift." In *From Frege to Gödel*, edited by Jean van Heijenoort. Cambridge, MA: Harvard University Press, 1967 [1879].

Gardner, Martin. *Logic Machines, Diagrams and Boolean Algebra*. New York: Dover, 1968 [1958].

Gentzen, Gerhard. "Untersuchungen über das logische Schliessen." *Mathematische Zeitschrift* 39 (1934, 1935): 179-210, 405-31.

Gödel, Kurt. "[Some Metamathematical Results on Completeness and Consistency]." In *From Frege to Gödel*, edited by Jean van Heijenoort. Cambridge, MA: Harvard, 1970 [1930].

———. "[On Formally Undecidable Propositions of Principia Mathematica and Relatied Systems I]." In *From Frege to Gödel*, edited by Jean van Heijenoort. Cambridge, MA: Harvard University Press, 1970 [1931].

———. "[On Completeness and Consistency]." In *From Frege to Gödel*, edited by Jean van Heijenoort. Cambridge, MA: Harvard University Press, 1970 [1931].

Herbrand, Jaques. “Recherches sur la Théorie de la Démonstration.” Available from.

———. “Sur la Théorie de la Démonstration.” *Comptes Rendus des Séances de la Sociéte des Sciences et des Lettres de Varsovie, Classe III* 24 (1931): 12-56.

Hilbert, D. and W. Ackermann. *Principles of Mathematical Logic*. New Yorl: Chelsea, 1950 [1928].

Hintikka, Jaakko, *Knowledge ans Belief* Ithaca, N.Y.: Cornell University Press. 1962.

Hodges, Wilfrid. “Elementary Predicate Logic.” In *Handbook of Philosophical Logic*, edited by D. and F. Guethner Gabby. Dordrecht: Reidel, 1983.

Hofstader, Douglas R. *Gödel, Esher, Bach*.,: Vintage Books, 1989.

Jardine, Lisa. “Humanistic Logic.” In *Chambridge History of Renaissance Logic*, edited by Charles et al. Schmitt. Cambridge: Cambridge University Press, 1988.

Kaplan, David. “Quantifying In.” In *Words and Objections*, edited by Donald and Jaakko Hintikka Davidson. Dordrecht: Reidel, 1969.

Levey, Samule. “Leibniz on Mathematics,” *Philosophical Review* 107 (1998), pp. 49-96. “Leibniz on Mathematics,” *Philosophical Review* 107 (1998), pp. 49-96.

Lewis, C. I. *A Survey of Symbolic Logic*. New York: Dover, 1961 [1918].

Lukasiewicz, Jan and A. Tarski. “Untersuchugen über den Aussagenlalkül.” *Comptes Rendus des Séances de la Sociéte des Sciences et des Lettres de Varsovie* 23 (1930).

Martin, John N. *Elements of Formal Semantics*.. Orlando: Academic Press, 1987.

Montague, Richard. “Intensional Logic.” In *Formal Philosophy*, edited by Richamond Thomason. New Haven: Yale University Press, 1974.

———. “Pragmatics and Intensional Logic.” In *Formal Philosophy*, edited by Richamond Thomason. New Haven: Yale University Press, 1974.

Parkinson, G. H. R. *Leibniz, Logical Papers*. Cambridge: Clarendon Press, 1966.

Prawitz, Dag. *Natural Deduction*. Stockholm: Almqvist and Wiskell, 1965.

Proclus. *Proclus: A Commentary on the First Book of Euclid's Elements*. Translated by Glenn R. Morrow. Princeton, NJ: Princeton University Press, 1970.

Quine, W. V. O. *Methods of Logic*. New York: Holt, Reinhart and Winston, 1950.

———. *Mathematical Logic*. Revised Edition ed. New York: Harper and Row, 1951.

———. *Set Theory and Its Logic.* Cambridge: Harvard University Press, 1963.

Rescher, Nicholas. *Many-Valued Logic*. New York: McGraw Hill, 1969.

Richard Soroabji. *Time, Creation and the Continuum* (London: Duckworth, 1983).

Simons, Leo. “Logic Without Tautologies.” *Notre Dame Journal of Formal Logic* 15 (1974): 411-31.

———. “More Logic Without Tautologies.” *Notre Dame Journal of Formal Logic* 19 (1978): 543-57.

Smullyan, Raymond M. *What is the Name of This Book?* New York: Penguin, 1990.

———. *Gödel's Incompleteness Theorem*. New York: Oxford University Press, 1992.

Tarski, Alfred. “Truth and Proof.” *Philosophy and Phenomenological Research* 4 (1944): 341-75.

———. “Contributions to the Theory of Models.” *Indagationes Mathematicae* 16 (1954): 572-88.

———. “The Concept of Truth in Formalized Langauges.” In *Logic Semantics, Metamathematics*, edited by Alfred Tarski. Oxford: Clarendon Press, 1956 [1931].

———. “Foundations of the Calculus of Systems.” In *Logic, Semantics, Metamathematics*, edited by Alfred Tarski. Oxford: Clarendon Press, 1956 [1935].

———. “Truth and Proof.” *Scientific American* 194 (1969): 63-77.

Tennant, Neil. *Natural Logic*. Edinburgh: Edingburgh University Press, 1978.

Whitehead, Alfred and Bertrand Russell. *Principia Mathematica*. 3 vols. Cambridge: Cambridge University Press, 1910-11.

Wilder, Raymond L. *Introduction to the Foundations of Mathematics*. 2nd ed. New York: Wiley, 1967.