Understanding Mathematical Logic and its Applications

Brenden Swafford

First Edition, 2012

ISBN 978-81-323-2696-0

Published by:
Orange Apple
4735/22 Prakashdeep Bldg,
Ansari Road, Darya Ganj,
Delhi - 110002
Email: info@wtbooks.com

Table of Contents

Chapter- 1

Introduction to Mathematical Logic

Mathematical logic (also known as **symbolic logic**) is a subfield of mathematics with close connections to computer science and philosophical logic. The field includes both the mathematical study of logic and the applications of formal logic to other areas of mathematics. The unifying themes in mathematical logic include the study of the expressive power of formal systems and the deductive power of formal proof systems.

Mathematical logic is often divided into the fields of set theory, model theory, recursion theory, and proof theory. These areas share basic results on logic, particularly first-order logic, and definability.

Since its inception, mathematical logic has contributed to, and has been motivated by, the study of foundations of mathematics. This study began in the late 19th century with the development of axiomatic frameworks for geometry, arithmetic, and analysis. In the early 20th century it was shaped by David Hilbert's program to prove the consistency of foundational theories. Results of Kurt Gödel, Gerhard Gentzen and others provided partial resolution to the program, and clarified the issues involved in proving consistency. Work in set theory showed that almost all ordinary mathematics can be formalized in terms of sets, although there are some theorems that cannot be proven in common axiom systems for set theory. Contemporary work in the foundations of mathematics often focuses on establishing which parts of mathematics can be formalized in particular formal systems, rather than trying to find theories in which all of mathematics can be developed.

History

Mathematical logic emerged in the mid-19th century as a subfield of mathematics independent of the traditional study of logic (Ferreirós 2001, p. 443). Before this emergence, logic was studied with rhetoric, through the syllogism, and with philosophy. The first half of the 20th century saw an explosion of fundamental results, accompanied by vigorous debate over the foundations of mathematics.

Early history

Sophisticated theories of logic were developed in many cultures, including China, India, Greece and the Islamic world. In the 18th century, attempts to treat the operations of

formal logic in a symbolic or algebraic way had been made by philosophical mathematicians including Leibniz and Lambert, but their labors remained isolated and little known.

19th century

In the middle of the nineteenth century, George Boole and then Augustus De Morgan presented systematic mathematical treatments of logic. Their work, building on work by algebraists such as George Peacock, extended the traditional Aristotelian doctrine of logic into a sufficient framework for the study of foundations of mathematics (Katz 1998, p. 686).

Charles Sanders Peirce built upon the work of Boole to develop a logical system for relations and quantifiers, which he published in several papers from 1870 to 1885. Gottlob Frege presented an independent development of logic with quantifiers in his Begriffsschrift, published in 1879, a work generally considered as marking a turning point in the history of logic. Frege's work remained obscure, however, until Bertrand Russell began to promote it near the turn of the century. The two-dimensional notation Frege developed was never widely adopted and is unused in contemporary texts.

From 1890 to 1905, Ernst Schröder published Vorlesungen über die Algebra der Logik in three volumes. This work summarized and extended the work of Boole, De Morgan, and Peirce, and was a comprehensive reference to symbolic logic as it was understood at the end of the 19th century.

Foundational theories

Some concerns that mathematics had not been built on a proper foundation led to the development of axiomatic systems for fundamental areas of mathematics such as arithmetic, analysis, and geometry.

In logic, the term arithmetic refers to the theory of the natural numbers. Giuseppe Peano (1888) published a set of axioms for arithmetic that came to bear his name (Peano axioms), using a variation of the logical system of Boole and Schröder but adding quantifiers. Peano was unaware of Frege's work at the time. Around the same time Richard Dedekind showed that the natural numbers are uniquely characterized by their induction properties. Dedekind (1888) proposed a different characterization, which lacked the formal logical character of Peano's axioms. Dedekind's work, however, proved theorems inaccessible in Peano's system, including the uniqueness of the set of natural numbers (up to isomorphism) and the recursive definitions of addition and multiplication from the successor function and mathematical induction.

In the mid-19th century, flaws in Euclid's axioms for geometry became known (Katz 1998, p. 774). In addition to the independence of the parallel postulate, established by Nikolai Lobachevsky in 1826 (Lobachevsky 1840), mathematicians discovered that certain theorems taken for granted by Euclid were not in fact provable from his axioms.

Among these is the theorem that a line contains at least two points, or that circles of the same radius whose centers are separated by that radius must intersect. Hilbert (1899) developed a complete set of axioms for geometry, building on previous work by Pasch (1882). The success in axiomatizing geometry motivated Hilbert to seek complete axiomatizations of other areas of mathematics, such as the natural numbers and the real line. This would prove to be a major area of research in the first half of the 20th century.

The 19th century saw great advances in the theory of real analysis, including theories of convergence of functions and Fourier series. Mathematicians such as Karl Weierstrass began to construct functions that stretched intuition, such as nowhere-differentiable continuous functions. Previous conceptions of a function as a rule for computation, or a smooth graph, were no longer adequate. Weierstrass began to advocate the arithmetization of analysis, which sought to axiomatize analysis using properties of the natural numbers. The modern (ε, δ)-definition of limit and continuous functions was already developed by Bolzano in 1817 (Felscher 2000), but remained relatively unknown. Cauchy in 1821 defined continuity in terms of infinitesimals. In 1858, Dedekind proposed a definition of the real numbers in terms of Dedekind cuts of rational numbers (Dedekind 1872), a definition still employed in contemporary texts.

Georg Cantor developed the fundamental concepts of infinite set theory. His early results developed the theory of cardinality and proved that the reals and the natural numbers have different cardinalities (Cantor 1874). Over the next twenty years, Cantor developed a theory of transfinite numbers in a series of publications. In 1891, he published a new proof of the uncountability of the real numbers that introduced the diagonal argument, and used this method to prove Cantor's theorem that no set can have the same cardinality as its powerset. Cantor believed that every set could be well-ordered, but was unable to produce a proof for this result, leaving it as an open problem in 1895 (Katz 1998, p. 807).

20th century

In the early decades of the 20th century, the main areas of study were set theory and formal logic. The discovery of paradoxes in informal set theory caused some to wonder whether mathematics itself is inconsistent, and to look for proofs of consistency.

In 1900, Hilbert posed a famous list of 23 problems for the next century. The first two of these were to resolve the continuum hypothesis and prove the consistency of elementary arithmetic, respectively; the tenth was to produce a method that could decide whether a multivariate polynomial equation over the integers has a solution. Subsequent work to resolve these problems shaped the direction of mathematical logic, as did the effort to resolve Hilbert's Entscheidungsproblem, posed in 1928. This problem asked for a procedure that would decide, given a formalized mathematical statement, whether the statement is true or false.

Set theory and paradoxes

Ernst Zermelo (1904) gave a proof that every set could be well-ordered, a result Georg Cantor had been unable to obtain. To achieve the proof, Zermelo introduced the axiom of choice, which drew heated debate and research among mathematicians and the pioneers of set theory. The immediate criticism of the method led Zermelo to publish a second exposition of his result, directly addressing criticisms of his proof (Zermelo 1908a). This paper led to the general acceptance of the axiom of choice in the mathematics community.

Skepticism about the axiom of choice was reinforced by recently discovered paradoxes in naive set theory. Cesare Burali-Forti (1897) was the first to state a paradox: the Burali-Forti paradox shows that the collection of all ordinal numbers cannot form a set. Very soon thereafter, Bertrand Russell discovered Russell's paradox in 1901, and Jules Richard (1905) discovered Richard's paradox.

Zermelo (1908b) provided the first set of axioms for set theory. These axioms, together with the additional axiom of replacement proposed by Abraham Fraenkel, are now called Zermelo–Fraenkel set theory (ZF). Zermelo's axioms incorporated the principle of limitation of size to avoid Russell's paradox.

In 1910, the first volume of Principia Mathematica by Russell and Alfred North Whitehead was published. This seminal work developed the theory of functions and cardinality in a completely formal framework of type theory, which Russell and Whitehead developed in an effort to avoid the paradoxes. Principia Mathematica is considered one of the most influential works of the 20th century, although the framework of type theory did not prove popular as a foundational theory for mathematics (Ferreirós 2001, p. 445).

Fraenkel (1922) proved that the axiom of choice cannot be proved from the remaining axioms of Zermelo's set theory with urelements. Later work by Paul Cohen (1966) showed that the addition of urelements is not needed, and the axiom of choice is unprovable in ZF. Cohen's proof developed the method of forcing, which is now an important tool for establishing independence results in set theory.

Symbolic logic

Leopold Löwenheim (1915) and Thoralf Skolem (1920) obtained the Löwenheim–Skolem theorem, which says that first-order logic cannot control the cardinalities of infinite structures. Skolem realized that this theorem would apply to first-order formalizations of set theory, and that it implies any such formalization has a countable model. This counterintuitive fact became known as Skolem's paradox.

In his doctoral thesis, Kurt Gödel (1929) proved the completeness theorem, which establishes a correspondence between syntax and semantics in first-order logic. Gödel used the completeness theorem to prove the compactness theorem, demonstrating the

finitary nature of first-order logical consequence. These results helped establish first-order logic as the dominant logic used by mathematicians.

In 1931, Gödel published On Formally Undecidable Propositions of Principia Mathematica and Related Systems, which proved the incompleteness (in a different meaning of the word) of all sufficiently strong, effective first-order theories. This result, known as Gödel's incompleteness theorem, establishes severe limitations on axiomatic foundations for mathematics, striking a strong blow to Hilbert's program. It showed the impossibility of providing a consistency proof of arithmetic within any formal theory of arithmetic. Hilbert, however, did not acknowledge the importance of the incompleteness theorem for some time.

Gödel's theorem shows that a consistency proof of any sufficiently strong, effective axiom system cannot be obtained in the system itself, if the system is consistent, nor in any weaker system. This leaves open the possibility of consistency proofs that cannot be formalized within the system they consider. Gentzen (1936) proved the consistency of arithmetic using a finitistic system together with a principle of transfinite induction. Gentzen's result introduced the ideas of cut elimination and proof-theoretic ordinals, which became key tools in proof theory. Gödel (1958) gave a different consistency proof, which reduces the consistency of classical arithmetic to that of intutitionistic arithmetic in higher types.

Beginnings of the other branches

Alfred Tarski developed the basics of model theory.

Beginning in 1935, a group of prominent mathematicians collaborated under the pseudonym Nicolas Bourbaki to publish a series of encyclopedic mathematics texts. These texts, written in an austere and axiomatic style, emphasized rigorous presentation and set-theoretic foundations. Terminology coined by these texts, such as the words bijection, injection, and surjection, and the set-theoretic foundations the texts employed, were widely adopted throughout mathematics.

The study of computability came to be known as recursion theory, because early formalizations by Gödel and Kleene relied on recursive definitions of functions. When these definitions were shown equivalent to Turing's formalization involving Turing machines, it became clear that a new concept – the computable function – had been discovered, and that this definition was robust enough to admit numerous independent characterizations. In his work on the incompleteness theorems in 1931, Gödel lacked a rigorous concept of an effective formal system; he immediately realized that the new definitions of computability could be used for this purpose, allowing him to state the incompleteness theorems in generality that could only be implied in the original paper.

Numerous results in recursion theory were obtained in the 1940s by Stephen Cole Kleene and Emil Leon Post. Kleene (1943) introduced the concepts of relative computability, foreshadowed by Turing (1939), and the arithmetical hierarchy. Kleene later generalized

recursion theory to higher-order functionals. Kleene and Kreisel studied formal versions of intuitionistic mathematics, particularly in the context of proof theory.

Subfields and scope

The Handbook of Mathematical Logic makes a rough division of contemporary mathematical logic into four areas:

1. set theory
2. model theory
3. recursion theory, and
4. proof theory and constructive mathematics (considered as parts of a single area).

Each area has a distinct focus, although many techniques and results are shared between multiple areas. The border lines between these fields, and the lines between mathematical logic and other fields of mathematics, are not always sharp. Gödel's incompleteness theorem marks not only a milestone in recursion theory and proof theory, but has also led to Löb's theorem in modal logic. The method of forcing is employed in set theory, model theory, and recursion theory, as well as in the study of intuitionistic mathematics.

The mathematical field of category theory uses many formal axiomatic methods, and includes the study of categorical logic, but category theory is not ordinarily considered a subfield of mathematical logic. Because of its applicability in diverse fields of mathematics, mathematicians including Saunders Mac Lane have proposed category theory as a foundational system for mathematics, independent of set theory. These foundations use toposes, which resemble generalized models of set theory that may employ classical or nonclassical logic.

Formal logical systems

At its core, mathematical logic deals with mathematical concepts expressed using formal logical systems. These systems, though they differ in many details, share the common property of considering only expressions in a fixed formal language, or signature. The system of first-order logic is the most widely studied today, because of its applicability to foundations of mathematics and because of its desirable proof-theoretic properties. Stronger classical logics such as second-order logic or infinitary logic are also studied, along with nonclassical logics such as intuitionistic logic.

Other classical logics

Many logics besides first-order logic are studied. These include infinitary logics, which allow for formulas to provide an infinite amount of information, and higher-order logics, which include a portion of set theory directly in their semantics.

The most well studied infinitary logic is $L_{\omega_1,\omega}$. In this logic, quantifiers may only be nested to finite depths, as in first order logic, but formulas may have finite or countably infinite conjunctions and disjunctions within them. Thus, for example, it is possible to say that an object is a whole number using a formula of $L_{\omega_1,\omega}$ such as

$$(x = 0) \vee (x = 1) \vee (x = 2) \vee \cdots.$$

Higher-order logics allow for quantification not only of elements of the domain of discourse, but subsets of the domain of discourse, sets of such subsets, and other objects of higher type. The semantics are defined so that, rather than having a separate domain for each higher-type quantifier to range over, the quantifiers instead range over all objects of the appropriate type. The logics studied before the development of first-order logic, for example Frege's logic, had similar set-theoretic aspects. Although higher-order logics are more expressive, allowing complete axiomatizations of structures such as the natural numbers, they do not satisfy analogues of the completeness and compactness theorems from first-order logic, and are thus less amenable to proof-theoretic analysis.

Another type of logics are fixed-point logics that allow inductive definitions, like one writes for primitive recursive functions.

One can formally define an extension of first-order logic — a notion which encompasses all logics in this section because they behave like first-order logic in certain fundamental ways, but does not encompass all logics in general, e.g. it does not encompass intuitionistic, modal or fuzzy logic. Lindström's theorem implies that the only extension of first-order logic satisfying both the Compactness theorem and the Downward Löwenheim–Skolem theorem is first-order logic.

Nonclassical and modal logic

Modal logics include additional modal operators, such as an operator which states that a particular formula is not only true, but necessarily true. Although modal logic is not often used to axiomatize mathematics, it has been used to study the properties of first-order provability (Solovay 1976) and set-theoretic forcing (Hamkins and Löwe 2007).

Intuitionistic logic was developed by Heyting to study Brouwer's program of intuitionism, in which Brouwer himself avoided formalization. Intuitionistic logic specifically does not include the law of the excluded middle, which states that each sentence is either true or its negation is true. Kleene's work with the proof theory of intuitionistic logic showed that constructive information can be recovered from intuitionistic proofs. For example, any provably total function in intuitionistic arithmetic is computable; this is not true in classical theories of arithmetic such as Peano arithmetic.

Algebraic logic

Algebraic logic uses the methods of abstract algebra to study the semantics of formal logics. A fundamental example is the use of Boolean algebras to represent truth values in classical propositional logic, and the use of Heyting algebras to represent truth values in intuitionistic propositional logic. Stronger logics, such as first-order logic and higher-order logic, are studied using more complicated algebraic structures such as cylindric algebras.

Set theory

Set theory is the study of sets, which are abstract collections of objects. Many of the basic notions, such as ordinal and cardinal numbers, were developed informally by Cantor before formal axiomatizations of set theory were developed. The first such axiomatization, due to Zermelo (1908b), was extended slightly to become Zermelo–Fraenkel set theory (ZF), which is now the most widely used foundational theory for mathematics.

Other formalizations of set theory have been proposed, including von Neumann–Bernays–Gödel set theory (NBG), Morse–Kelley set theory (MK), and New Foundations (NF). Of these, ZF, NBG, and MK are similar in describing a cumulative hierarchy of sets. New Foundations takes a different approach; it allows objects such as the set of all sets at the cost of restrictions on its set-existence axioms. The system of Kripke-Platek set theory is closely related to generalized recursion theory.

Two famous statements in set theory are the axiom of choice and the continuum hypothesis. The axiom of choice, first stated by Zermelo (1904), was proved independent of ZF by Fraenkel (1922), but has come to be widely accepted by mathematicians. It states that given a collection of nonempty sets there is a single set C that contains exactly one element from each set in the collection. The set C is said to "choose" one element from each set in the collection. While the ability to make such a choice is considered obvious by some, since each set in the collection is nonempty, the lack of a general, concrete rule by which the choice can be made renders the axiom nonconstructive. Stefan Banach and Alfred Tarski (1924) showed that the axiom of choice can be used to decompose a solid ball into a finite number of pieces which can then be rearranged, with no scaling, to make two solid balls of the original size. This theorem, known as the Banach-Tarski paradox, is one of many counterintuitive results of the axiom of choice.

The continuum hypothesis, first proposed as a conjecture by Cantor, was listed by David Hilbert as one of his 23 problems in 1900. Gödel showed that the continuum hypothesis cannot be disproven from the axioms of Zermelo–Fraenkel set theory (with or without the axiom of choice), by developing the constructible universe of set theory in which the continuum hypothesis must hold. In 1963, Paul Cohen showed that the continuum hypothesis cannot be proven from the axioms of Zermelo–Fraenkel set theory (Cohen 1966). This independence result did not completely settle Hilbert's question, however, as it is possible that new axioms for set theory could resolve the hypothesis. Recent work

along these lines has been conducted by W. Hugh Woodin, although its importance is not yet clear (Woodin 2001).

Contemporary research in set theory includes the study of large cardinals and determinacy. Large cardinals are cardinal numbers with particular properties so strong that the existence of such cardinals cannot be proved in ZFC. The existence of the smallest large cardinal typically studied, an inaccessible cardinal, already implies the consistency of ZFC. Despite the fact that large cardinals have extremely high cardinality, their existence has many ramifications for the structure of the real line. Determinacy refers to the possible existence of winning strategies for certain two-player games (the games are said to be determined). The existence of these strategies implies structural properties of the real line and other Polish spaces.

Model theory

Model theory studies the models of various formal theories. Here a theory is a set of formulas in a particular formal logic and signature, while a model is a structure that gives a concrete interpretation of the theory. Model theory is closely related to universal algebra and algebraic geometry, although the methods of model theory focus more on logical considerations than those fields.

The set of all models of a particular theory is called an elementary class; classical model theory seeks to determine the properties of models in a particular elementary class, or determine whether certain classes of structures form elementary classes.

The method of quantifier elimination can be used to show that definable sets in particular theories cannot be too complicated. Tarski (1948) established quantifier elimination for real-closed fields, a result which also shows the theory of the field of real numbers is decidable. (He also noted that his methods were equally applicable to algebraically closed fields of arbitrary characteristic.) A modern subfield developing from this is concerned with o-minimal structures.

Morley's categoricity theorem, proved by Michael D. Morley (1965), states that if a first-order theory in a countable language is categorical in some uncountable cardinality, i.e. all models of this cardinality are isomorphic, then it is categorical in all uncountable cardinalities.

A trivial consequence of the continuum hypothesis is that a complete theory with less than continuum many nonisomorphic countable models can have only countably many. Vaught's conjecture, named after Robert Lawson Vaught, says that this is true even independently of the continuum hypothesis. Many special cases of this conjecture have been established.

Algorithmically unsolvable problems

An important subfield of recursion theory studies algorithmic unsolvability; a decision problem or function problem is **algorithmically unsolvable** if there is no possible computable algorithm which returns the correct answer for all legal inputs to the problem. The first results about unsolvability, obtained independently by Church and Turing in 1936, showed that the Entscheidungsproblem is algorithmically unsolvable. Turing proved this by establishing the unsolvability of the halting problem, a result with far-ranging implications in both recursion theory and computer science.

There are many known examples of undecidable problems from ordinary mathematics. The word problem for groups was proved algorithmically unsolvable by Pyotr Novikov in 1955 and independently by W. Boone in 1959. The busy beaver problem, developed by Tibor Radó in 1962, is another well-known example.

Hilbert's tenth problem asked for an algorithm to determine whether a multivariate polynomial equation with integer coefficients has a solution in the integers. Partial progress was made by Julia Robinson, Martin Davis and Hilary Putnam. The algorithmic unsolvability of the problem was proved by Yuri Matiyasevich in 1970 (Davis 1973).

Connections with computer science

The study of computability theory in computer science is closely related to the study of computability in mathematical logic. There is a difference of emphasis, however. Computer scientists often focus on concrete programming languages and feasible computability, while researchers in mathematical logic often focus on computability as a theoretical concept and on noncomputability.

The theory of semantics of programming languages is related to model theory, as is program verification (in particular, model checking). The Curry–Howard isomorphism between proofs and programs relates to proof theory, especially intuitionistic logic. Formal calculi such as the lambda calculus and combinatory logic are now studied as idealized programming languages.

Computer science also contributes to mathematics by developing techniques for the automatic checking or even finding of proofs, such as automated theorem proving and logic programming.

Descriptive complexity theory relates logics to computational complexity. The first significant result in this area, Fagin's theorem (1974) established that NP is precisely the set of languages expressible by sentences of existential second-order logic.

Foundations of mathematics

In the 19th century, mathematicians became aware of logical gaps and inconsistencies in their field. It was shown that Euclid's axioms for geometry, which had been taught for centuries as an example of the axiomatic method, were incomplete. The use of infinitesimals, and the very definition of function, came into question in analysis, as pathological examples such as Weierstrass' nowhere-differentiable continuous function were discovered.

Cantor's study of arbitrary infinite sets also drew criticism. Leopold Kronecker famously stated "God made the integers; all else is the work of man," endorsing a return to the study of finite, concrete objects in mathematics. Although Kronecker's argument was carried forward by constructivists in the 20th century, the mathematical community as a whole rejected them. David Hilbert argued in favor of the study of the infinite, saying "No one shall expel us from the Paradise that Cantor has created."

Mathematicians began to search for axiom systems that could be used to formalize large parts of mathematics. In addition to removing ambiguity from previously-naive terms such as function, it was hoped that this axiomatization would allow for consistency proofs. In the 19th century, the main method of proving the consistency of a set of axioms was to provide a model for it. Thus, for example, non-Euclidean geometry can be proved consistent by defining point to mean a point on a fixed sphere and line to mean a great circle on the sphere. The resulting structure, a model of elliptic geometry, satisfies the axioms of plane geometry except the parallel postulate.

With the development of formal logic, Hilbert asked whether it would be possible to prove that an axiom system is consistent by analyzing the structure of possible proofs in the system, and showing through this analysis that it is impossible to prove a contradiction. This idea led to the study of proof theory. Moreover, Hilbert proposed that the analysis should be entirely concrete, using the term finitary to refer to the methods he would allow but not precisely defining them. This project, known as Hilbert's program, was seriously affected by Gödel's incompleteness theorems, which show that the consistency of formal theories of arithmetic cannot be established using methods formalizable in those theories. Gentzen showed that it is possible to produce a proof of the consistency of arithmetic in a finitary system augmented with axioms of transfinite induction, and the techniques he developed to so do were seminal in proof theory.

A second thread in the history of foundations of mathematics involves nonclassical logics and constructive mathematics. The study of constructive mathematics includes many different programs with various definitions of constructive. At the most accommodating end, proofs in ZF set theory that do not use the axiom of choice are called constructive by many mathematicians. More limited versions of constructivism limit themselves to natural numbers, number-theoretic functions, and sets of natural numbers (which can be used to represent real numbers, facilitating the study of mathematical analysis). A common idea is that a concrete means of computing the values of the function must be known before the function itself can be said to exist.

In the early 20th century, Luitzen Egbertus Jan Brouwer founded intuitionism as a philosophy of mathematics. This philosophy, poorly understood at first, stated that in order for a mathematical statement to be true to a mathematician, that person must be able to intuit the statement, to not only believe its truth but understand the reason for its truth. A consequence of this definition of truth was the rejection of the law of the excluded middle, for there are statements that, according to Brouwer, could not be claimed to be true while their negations also could not be claimed true. Brouwer's philosophy was influential, and the cause of bitter disputes among prominent mathematicians. Later, Kleene and Kreisel would study formalized versions of intuitionistic logic (Brouwer rejected formalization, and presented his work in unformalized natural language). With the advent of the BHK interpretation and Kripke models, intuitionism became easier to reconcile with classical mathematics.

Chapter- 2

Propositional Calculus

In mathematical logic, a **propositional calculus** or **logic** (also called **sentential calculus** or **sentential logic**) is a formal system in which formulas of a formal language may be interpreted as representing propositions. A system of inference rules and axioms allows certain formulas to be derived, called theorems; which may be interpreted as true propositions. The series of formulas which is constructed within such a system is called a derivation and the last formula of the series is a theorem, whose derivation may be interpreted as a proof of the truth of the proposition represented by the theorem.

Truth-functional propositional logic is a propositional logic whose interpretation limits the truth values of its propositions to two, usually true and false. Truth-functional propositional logic and systems isomorphic to it are considered to be **zeroth order logic**.

Terminology

In general terms, a calculus is a formal system that consists of a set of syntactic expressions (well-formed formulæ or wffs), a distinguished subset of these expressions (axioms), plus a set of formal rules that define a specific binary relation, intended to be interpreted as logical equivalence, on the space of expressions.

When the formal system is intended to be a logical system, the expressions are meant to be interpreted as statements, and the rules, known as inference rules, are typically intended to be truth-preserving. In this setting, the rules (which may include axioms) can then be used to derive ("infer") formulæ representing true statements from given formulæ representing true statements.

The set of axioms may be empty, a nonempty finite set, a countably infinite set, or be given by axiom schemata. A formal grammar recursively defines the expressions and well-formed formulæ (wffs) of the language. In addition a semantics may be given which defines truth and valuations (or interpretations).

The language of a propositional calculus consists of

1. a set of primitive symbols, variously referred to as atomic formulae, placeholders, proposition letters, or variables, and

2. a set of operator symbols, variously interpreted as logical operators or logical connectives.

A well-formed formula (wff) is any atomic formula, or any formula that can be built up from atomic formulæ by means of operator symbols according to the rules of the grammar.

Mathematicians sometimes distinguish between propositional constants, propositional variables, and schemata. Propositional constants represent some particular proposition, while propositional variables range over the set of all atomic propositions. Schemata, however, range over all propositions. It is common to represent propositional constants by A, B, and C, propositional variables by P, Q, and R, and schematic letters are often Greek letters, most often φ, ψ, and χ.

Basic concepts

The following outlines a standard propositional calculus. Many different formulations exist which are all more or less equivalent but differ in the details of

1. their language, that is, the particular collection of primitive symbols and operator symbols,
2. the set of axioms, or distinguished formulæ, and
3. the set of inference rules.

We may represent any given proposition with a letter which we call a propositional constant, analogous to representing a number by a letter in mathematics, for instance, a = 5. We require that all propositions have exactly one of two truth-values: true or false. To take an example, let P be the proposition that it is raining outside. This will be true if it is raining outside and false otherwise.

- We then define truth-functional operators, beginning with negation. We write $\neg P$ to represent the negation of P, which can be thought of as the denial of P. In the example above, $\neg P$ expresses that it is not raining outside, or by a more standard reading: "It is not the case that it is raining outside." When P is true, $\neg P$ is false; and when P is false, $\neg P$ is true. $\neg\neg P$ always has the same truth-value as P.
- Conjunction is a truth-functional connective which forms a proposition out of two simpler propositions, for example, P and Q. The conjunction of P and Q is written $P \ \& \ Q$, and expresses that each are true. We read $P \ \& \ Q$ as "P and Q". For any two propositions, there are four possible assignments of truth values:
    1. P is true and Q is true
    2. P is true and Q is false
    3. P is false and Q is true
    4. P is false and Q is false

The conjunction of P and Q is true in case 1 and is false otherwise. Where P is the proposition that it is raining outside and Q is the proposition that a cold-front is over Kansas, $P \ \& \ Q$ is true when it is raining outside and there is a cold-front over Kansas. If it is not raining outside, then $P \ \& \ Q$ is false; and if there is no cold-front over Kansas, then $P \ \& \ Q$ is false.

- Disjunction resembles conjunction in that it forms a proposition out of two simpler propositions. We write it $P \vee Q$, and it is read "P or Q". It expresses that either P or Q is true. Thus, in the cases listed above, the disjunction of P and Q is true in all cases except 4. Using the example above, the disjunction expresses that it is either raining outside or there is a cold front over Kansas. (Note, this use of disjunction is supposed to resemble the use of the English word "or". However, it is most like the English inclusive "or", which can be used to express the truth of at least one of two propositions. It is not like the English exclusive "or", which expresses the truth of exactly one of two propositions. That is to say, the exclusive "or" is false when both P and Q are true (case 1). An example of the exclusive or is: You may have a bagel or a pastry, but not both. Sometimes, given the appropriate context, the addendum "but not both" is omitted but implied.)
- Material conditional also joins two simpler propositions, and we write $P \rightarrow Q$, which is read "if P then Q". The proposition to the left of the arrow is called the antecedent and the proposition to the right is called the consequent. (There is no such designation for conjunction or disjunction, since they are commutative operations.) It expresses that Q is true whenever P is true. Thus it is true in every case above except case 2, because this is the only case when P is true but Q is not. Using the example, if P then Q expresses that if it is raining outside then there is a cold-front over Kansas. The material conditional is often confused with physical causation. The material conditional, however, only relates two propositions by their truth-values—which is not the relation of cause and effect. It is contentious in the literature whether the material implication represents logical causation.
- Biconditional joins two simpler propositions, and we write $P \leftrightarrow Q$, which is read "P if and only if Q". It expresses that P and Q have the same truth-value, thus P if and only if Q is true in cases 1 and 4, and false otherwise.

It is extremely helpful to look at the truth tables for these different operators, as well as the method of analytic tableaux.

Closure under operations

Propositional logic is closed under truth-functional connectives. That is to say, for any proposition φ, $\neg\varphi$ is also a proposition. Likewise, for any propositions φ and ψ, $\varphi \ \& \ \psi$ is a proposition, and similarly for disjunction, conditional, and biconditional. This implies that, for instance, $P \ \& \ Q$ is a proposition, and so it can be conjoined with another proposition. In order to represent this, we need to use parentheses to indicate which proposition is conjoined with which. For instance, $P \ \& \ Q \ \& \ R$ is not a well-

formed formula, because we do not know if we are conjoining $P \ \& \ Q$ with R or if we are conjoining P with $Q \ \& \ R$. Thus we must write either $(P \ \& \ Q) \ \& \ R$ to represent the former, or $P \ \& \ (Q \ \& \ R)$ to represent the latter. By evaluating the truth conditions, we see that both expressions have the same truth conditions (will be true in the same cases), and moreover that any proposition formed by arbitrary conjunctions will have the same truth conditions, regardless of the location of the parentheses. This means that conjunction is associative, however, one should not assume that parentheses never serve a purpose. For instance, the sentence $P \ \& \ (Q \vee R)$ does not have the same truth conditions as $(P \ \& \ Q) \vee R$, so they are different sentences distinguished only by the parentheses. One can verify this by the truth-table method referenced above.

Note: For any arbitrary number of propositional constants, we can form a finite number of cases which list their possible truth-values. A simple way to generate this is by truth-tables, in which one writes P, Q, …, Z for any list of k propositional constants—that is to say, any list of propositional constants with k entries. Below this list, one writes 2^k rows, and below P one fills in the first half of the rows with true (or T) and the second half with false (or F). Below Q one fills in one-quarter of the rows with T, then one-quarter with F, then one-quarter with T and the last quarter with F. The next column alternates between true and false for each eighth of the rows, then sixteenths, and so on, until the last propositional constant varies between T and F for each row. This will give a complete listing of cases or truth-value assignments possible for those propositional constants.

Argument

The propositional calculus then defines an argument as a set of propositions. A valid argument is a set of propositions, the last of which follows from—or is implied by—the rest. All other arguments are invalid. The simplest valid argument is modus ponens, one instance of which is the following set of propositions:

$$\begin{array}{ll} 1. & P \rightarrow Q \\ 2. & P \\ \hline \therefore & Q \end{array}$$

This is a set of three propositions, each line is a proposition, and the last follows from the rest. The first two lines are called premises, and the last line the conclusion. We say that any proposition C follows from any set of propositions $(P_1,...,P_n)$, if C must be true whenever every member of the set $(P_1,...,P_n)$ is true. In the argument above, for any P and Q, whenever $P \rightarrow Q$ and P are true, necessarily Q is true. Notice that, when P is true, we cannot consider cases 3 and 4 (from the truth table). When $P \rightarrow Q$ is true, we cannot consider case 2. This leaves only case 1, in which Q is also true. Thus Q is implied by the premises.

This generalizes schematically. Thus, where φ and ψ may be any propositions at all,

$$\begin{array}{ll} 1. & \varphi \rightarrow \psi \\ 2. & \varphi \\ \hline \therefore & \psi \end{array}$$

Other argument forms are convenient, but not necessary. Given a complete set of axioms, modus ponens is sufficient to prove all other argument forms in propositional logic, and so we may think of them as derivative. Note, this is not true of the extension of propositional logic to other logics like first-order logic. First-order logic requires at least one additional rule of inference in order to obtain completeness.

The significance of argument in formal logic is that one may obtain new truths from established truths. In the first example above, given the two premises, the truth of Q is not yet known or stated. After the argument is made, Q is deduced. In this way, we define a deduction system as a set of all propositions that may be deduced from another set of propositions. For instance, given the set of propositions $A = \{P \vee Q, \neg Q \ \& \ R, (P \vee Q) \rightarrow R\}$, we can define a deduction system, Γ, which is the set of all propositions which follow from A. Reiteration is always assumed, so $P \vee Q, \neg Q \ \& \ R, (P \vee Q) \rightarrow R \in \Gamma$. Also, from the first element of A, last element, as well as modus ponens, R is a consequence, and so $R \in \Gamma$. Because we have not included sufficiently complete axioms, though, nothing else may be deduced. Thus, even though most deduction systems studied in propositional logic are able to deduce $(P \vee Q) \leftrightarrow (\neg P \rightarrow Q)$, this one is too weak to prove such a proposition.

Generic description of a propositional calculus

A **propositional calculus** is a formal system $\mathcal{L} = \mathcal{L}(\mathrm{A},\ \Omega,\ \mathrm{Z},\ \mathrm{I})$, where:

- The alpha set A is a finite set of elements called proposition symbols or propositional variables. Syntactically speaking, these are the most basic elements of the formal language $\mathcal{L}$, otherwise referred to as atomic formulæ or terminal elements. In the examples to follow, the elements of A are typically the letters p, q, r, and so on.

- The omega set Ω is a finite set of elements called operator symbols or logical connectives. The set Ω is partitioned into disjoint subsets as follows:

  $\Omega = \Omega_0 \cup \Omega_1 \cup \ldots \cup \Omega_j \cup \ldots \cup \Omega_m.$
  In this partition, Ω_j is the set of operator symbols of arity j.
  In the more familiar propositional calculi, Ω is typically partitioned as follows:
  $\Omega_1 = \{\neg\},$
  $\Omega_2 \subseteq \{\wedge, \vee, \rightarrow, \leftrightarrow\}.$
  A frequently adopted convention treats the constant logical values as operators of arity zero, thus:
  $\Omega_0 = \{0, 1\}.$

Some writers use the tilde (~) instead of ¬; and some use the ampersand (&) or ⋅ instead of ∧. Notation varies even more for the set of logical values, with symbols like {false, true}, {F, T}, or $\{\bot, \top\}$ all being seen in various contexts instead of {0, 1}.

- The zeta set Z is a finite set of transformation rules that are called inference rules when they acquire logical applications.

- The iota set I is a finite set of initial points that are called axioms when they receive logical interpretations.

The language of $\mathcal{L}$, also known as its set of formulæ, well-formed formulas or wffs, is inductively or recursively defined by the following rules:

1. Base: Any element of the alpha set A is a formula of $\mathcal{L}$.
2. If $p_1, p_2, \ldots, p_j$ are formulæ and f is in Ω_j, then $(f(p_1, p_2, \ldots, p_j))$ is a formula.
3. Closed: Nothing else is a formula of $\mathcal{L}$.

Repeated applications of these rules permits the construction of complex formulæ. For example:

1. By rule 1, p is a formula.
2. By rule 2, $\neg p$ is a formula.
3. By rule 1, q is a formula.
4. By rule 2, $(\neg p \vee q)$ is a formula.

Example 1. Simple axiom system

Let $\mathcal{L}_1 = \mathcal{L}(\mathrm{A}, \Omega, \mathrm{Z}, \mathrm{I})$, where A, Ω, Z, I are defined as follows:

- The alpha set A, is a finite set of symbols that is large enough to supply the needs of a given discussion, for example:

  $$\mathrm{A} = \{p, q, r, s, t, u\}.$$

- Of the three connectives for conjunction, disjunction, and implication (∧, ∨, and →), one can be taken as primitive and the other two can be defined in terms of it and negation (¬). Indeed, all of the logical connectives can be defined in terms of a sole sufficient operator. The biconditional (↔) can of course be defined in terms of conjunction and implication, with $a \leftrightarrow b$ defined as $(a \rightarrow b) \wedge (b \rightarrow a)$.
  Adopting negation and implication as the two primitive operations of a

propositional calculus is tantamount to having the omega set $\Omega = \Omega_1 \cup \Omega_2$partition as follows:

$$\Omega_1 = \{\neg\},$$
$$\Omega_2 = \{\rightarrow\}.$$

- An axiom system discovered by Jan Łukasiewicz formulates a propositional calculus in this language as follows. The axioms are all substitution instances of:
  - $(p \rightarrow (q \rightarrow p))$
  - $((p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r)))$
  - $((\neg p \rightarrow \neg q) \rightarrow (q \rightarrow p))$
- The rule of inference is modus ponens (i.e. from p and $(p \rightarrow q)$, infer q). Then $a \vee b$is defined as $\neg a \rightarrow b$, and $a \wedge b$is defined as $\neg(a \rightarrow \neg b)$.

Example 2. Natural deduction system

Let $\mathcal{L}_2 = \mathcal{L}(\mathrm{A}, \Omega, \mathrm{Z}, \mathrm{I})$, where A, Ω, Z, I are defined as follows:

- The alpha set A, is a finite set of symbols that is large enough to supply the needs of a given discussion, for example:

  $$\mathrm{A} = \{p, q, r, s, t, u\}.$$

- The omega set $\Omega = \Omega_1 \cup \Omega_2$partitions as follows:

  $$\Omega_1 = \{\neg\},$$
  $$\Omega_2 = \{\wedge, \vee, \rightarrow, \leftrightarrow\}.$$

In the following example of a propositional calculus, the transformation rules are intended to be interpreted as the inference rules of a so-called natural deduction system. The particular system presented here has no initial points, which means that its interpretation for logical applications derives its theorems from an empty axiom set.

- The set of initial points is empty, that is, $I = \varnothing$.
- The set of transformation rules, Z, is described as follows:

Our propositional calculus has ten inference rules. These rules allow us to derive other true formulae given a set of formulae that are assumed to be true. The first nine simply state that we can infer certain wffs from other wffs. The last rule however uses hypothetical reasoning in the sense that in the premise of the rule we temporarily assume

an (unproven) hypothesis to be part of the set of inferred formulae to see if we can infer a certain other formula. Since the first nine rules don't do this they are usually described as non-hypothetical rules, and the last one as a hypothetical rule.

Reductio ad absurdum (negation introduction)
: From p and [accepting q leads to a proof that $\neg p$], infer $\neg q$.

Double negative elimination
: From $\neg\neg p$, infer p.

Conjunction introduction
: From p and q, infer $(p \wedge q)$.
: From p and q, infer $(q \wedge p)$.

Conjunction elimination
: From $(p \wedge q)$, infer p.
: From $(p \wedge q)$, infer q.

Disjunction introduction
: From p, infer $(p \vee q)$.
: From p, infer $(q \vee p)$.

Disjunction elimination
: From $(p \vee q)$and $(p \rightarrow r)$and $(q \rightarrow r)$, infer r.

Biconditional introduction
: From $(p \rightarrow q)$and $(q \rightarrow p)$, infer $(p \leftrightarrow q)$.

Biconditional elimination
: From $(p \leftrightarrow q)$, infer $(p \rightarrow q)$.
: From $(p \leftrightarrow q)$, infer $(q \rightarrow p)$.

Modus ponens (conditional elimination)
: From p and $(p \rightarrow q)$, infer q.

Conditional proof (conditional introduction)
: From [accepting p allows a proof of q], infer $(p \rightarrow q)$.

Basic and derived argument forms

Basic and Derived Argument Forms		
Name	**Sequent**	**Description**
Modus Ponens	$((p \rightarrow q) \wedge p) \vdash q$	If p then q; p; therefore q
Modus Tollens	$((p \rightarrow q) \wedge \neg q) \vdash \neg p$	If p then q; not q; therefore

		not p
Hypothetical Syllogism	$((p \rightarrow q) \wedge (q \rightarrow r)) \vdash (p \rightarrow r)$	If p then q; if q then r; therefore, if p then r
Disjunctive Syllogism	$((p \vee q) \wedge \neg p) \vdash q$	Either p or q, or both; not p; therefore, q
Constructive Dilemma	$((p \rightarrow q) \wedge (r \rightarrow s) \wedge (p \vee r)) \vdash (q \vee s)$	If p then q; and if r then s; but p or r; therefore q or s
Destructive Dilemma	$((p \rightarrow q) \wedge (r \rightarrow s) \wedge (\neg q \vee \neg s)) \vdash (\neg p \vee \neg r)$	If p then q; and if r then s; but not q or not s; therefore not p or not r
Bidirectional Dilemma	$((p \rightarrow q) \wedge (r \rightarrow s) \wedge (p \vee \neg s)) \vdash (q \vee \neg r)$	If p then q; and if r then s; but p or not s; therefore q or not r
Simplification	$(p \wedge q) \vdash p$	p and q are true; therefore p is true
Conjunction	$p, q \vdash (p \wedge q)$	p and q are true separately; therefore they are true conjointly
Addition	$p \vdash (p \vee q)$	p is true;

		therefore the disjunction (p or q) is true
Composition	$((p \rightarrow q) \wedge (p \rightarrow r)) \vdash (p \rightarrow (q \wedge r))$	If p then q; and if p then r; therefore if p is true then q and r are true
De Morgan's Theorem (1)	$\neg(p \wedge q) \vdash (\neg p \vee \neg q)$	The negation of (p and q) is equiv. to (not p or not q)
De Morgan's Theorem (2)	$\neg(p \vee q) \vdash (\neg p \wedge \neg q)$	The negation of (p or q) is equiv. to (not p and not q)
Commutation (1)	$(p \vee q) \vdash (q \vee p)$	(p or q) is equiv. to (q or p)
Commutation (2)	$(p \wedge q) \vdash (q \wedge p)$	(p and q) is equiv. to (q and p)
Commutation (3)	$(p \leftrightarrow q) \vdash (q \leftrightarrow p)$	(p is equiv. to q) is equiv. to (q is equiv. to p)
Association (1)	$(p \vee (q \vee r)) \vdash ((p \vee q) \vee r)$	p or (q or r) is equiv. to (p or q) or r
Association	$(p \wedge (q \wedge r)) \vdash ((p \wedge q) \wedge r)$	p and (q

(2)		and r) is equiv. to (p and q) and r
Distribution (1)	$(p \wedge (q \vee r)) \vdash ((p \wedge q) \vee (p \wedge r))$	p and (q or r) is equiv. to (p and q) or (p and r)
Distribution (2)	$(p \vee (q \wedge r)) \vdash ((p \vee q) \wedge (p \vee r))$	p or (q and r) is equiv. to (p or q) and (p or r)
Double Negation	$p \vdash \neg\neg p$	p is equivalent to the negation of not p
Transposition	$(p \rightarrow q) \vdash (\neg q \rightarrow \neg p)$	If p then q is equiv. to if not q then not p
Material Implication	$(p \rightarrow q) \vdash (\neg p \vee q)$	If p then q is equiv. to not p or q
Material Equivalence (1)	$(p \leftrightarrow q) \vdash ((p \rightarrow q) \wedge (q \rightarrow p))$	(p is equiv. to q) means (if p is true then q is true) and (if q is true then p is true)
Material Equivalence (2)	$(p \leftrightarrow q) \vdash ((p \wedge q) \vee (\neg p \wedge \neg q))$	(p is equiv. to q) means either (p and q are true) or (both p and q are false)
Material	$(p \leftrightarrow q) \vdash ((p \vee \neg q) \wedge (\neg p \vee q))$	(p is equiv.

Equivalence (3)		to q) means, both (p or not q is true) and (not p or q is true)
Exportation	$((p \wedge q) \rightarrow r) \vdash (p \rightarrow (q \rightarrow r))$	from (if p and q are true then r is true) we can prove (if q is true then r is true, if p is true)
Importation	$(p \rightarrow (q \rightarrow r)) \vdash ((p \wedge q) \rightarrow r)$	
Tautology (1)	$p \vdash (p \vee p)$	p is true is equiv. to p is true or p is true
Tautology (2)	$p \vdash (p \wedge p)$	p is true is equiv. to p is true and p is true
Tertium non datur (Law of Excluded Middle)	$\vdash (p \vee \neg p)$	p or not p is true
Law of Non-Contradiction	$\vdash \neg(p \wedge \neg p)$	p and not p is false, is a true statement

Proofs in propositional calculus

One of the main uses of a propositional calculus, when interpreted for logical applications, is to determine relations of logical equivalence between propositional formulæ. These relationships are determined by means of the available transformation rules, sequences of which are called derivations or proofs.

In the discussion to follow, a proof is presented as a sequence of numbered lines, with each line consisting of a single formula followed by a reason or justification for introducing that formula. Each premise of the argument, that is, an assumption introduced as an hypothesis of the argument, is listed at the beginning of the sequence and is marked as a "premise" in lieu of other justification. The conclusion is listed on the last line. A proof is complete if every line follows from the previous ones by the correct application of a transformation rule.

Example of a proof

- To be shown that $A \rightarrow A$.
- One possible proof of this (which, though valid, happens to contain more steps than are necessary) may be arranged as follows:

Example of a Proof		
Number	**Formula**	**Reason**
1	A	premise
2	$A \vee A$	From (1) by disjunction introduction
3	$(A \vee A) \wedge A$	From (1) and (2) by conjunction introduction
4	A	From (3) by conjunction elimination
5	$A \vdash A$	Summary of (1) through (4)
6	$\vdash A \rightarrow A$	From (5) by conditional proof

Interpret $A \vdash A$as "Assuming A, infer A". Read $\vdash A \rightarrow A$as "Assuming nothing, infer that A implies A", or "It is a tautology that A implies A", or "It is always true that A implies A".

Soundness and completeness of the rules

The crucial properties of this set of rules are that they are sound and complete. Informally this means that the rules are correct and that no other rules are required. These claims can be made more formal as follows.

We define a truth assignment as a function that maps propositional variables to **true** or **false**. Informally such a truth assignment can be understood as the description of a possible state of affairs (or possible world) where certain statements are true and others are not. The semantics of formulae can then be formalized by defining for which "state of affairs" they are considered to be true, which is what is done by the following definition.

We define when such a truth assignment Asatisfies a certain wff with the following rules:

- Asatisfies the propositional variable Pif and only if $A(P) = \text{true}$
- Asatisfies $\neg\phi$if and only if Adoes not satisfy ϕ
- Asatisfies $(\phi \wedge \psi)$if and only if Asatisfies both $\phi$and ψ
- Asatisfies $(\phi \vee \psi)$if and only if Asatisfies at least one of either ϕor ψ
- Asatisfies $(\phi \rightarrow \psi)$if and only if it is not the case that Asatisfies ϕbut not ψ
- Asatisfies $(\phi \leftrightarrow \psi)$if and only if Asatisfies both $\phi$and ψor satisfies neither one of them

With this definition we can now formalize what it means for a formula $\phi$to be implied by a certain set $S$of formulae. Informally this is true if in all worlds that are possible given the set of formulae $S$the formula ϕalso holds. This leads to the following formal definition: We say that a set $S$of wffs semantically entails (or implies) a certain wff ϕif all truth assignments that satisfy all the formulae in Salso satisfy ϕ.

Finally we define syntactical entailment such that $\phi$is syntactically entailed by Sif and only if we can derive it with the inference rules that were presented above in a finite number of steps. This allows us to formulate exactly what it means for the set of inference rules to be sound and complete:

Soundness

If the set of wffs Ssyntactically entails wff ϕthen Ssemantically entails ϕ

Completeness

If the set of wffs Ssemantically entails wff ϕthen Ssyntactically entails ϕ

For the above set of rules this is indeed the case.

Sketch of a soundness proof

(For most logical systems, this is the comparatively "simple" direction of proof)

Notational conventions: Let G be a variable ranging over sets of sentences. Let A, B, and C range over sentences. For "G syntactically entails A" we write "G proves A". For "G semantically entails A" we write "G implies A".

We want to show: (A)(G)(if G proves A, then G implies A).

We note that "G proves A" has an inductive definition, and that gives us the immediate resources for demonstrating claims of the form "If G proves A, then ...". So our proof proceeds by induction.

I. Basis. Show: If A is a member of G, then G implies A.
II. Basis. Show: If A is an axiom, then G implies A.
III. Inductive step (induction on n, the length of the proof):
    a. Assume for arbitrary G and A that if G proves A in n or fewer steps, then G implies A.
    b. For each possible application of a rule of inference at step n + 1, leading to a new theorem B, show that G implies B.

Notice that Basis Step II can be omitted for natural deduction systems because they have no axioms. When used, Step II involves showing that each of the axioms is a (semantic) logical truth.

The Basis step(s) demonstrate(s) that the simplest provable sentences from G are also implied by G, for any G. (The is simple, since the semantic fact that a set implies any of its members, is also trivial.) The Inductive step will systematically cover all the further sentences that might be provable—by considering each case where we might reach a logical conclusion using an inference rule—and shows that if a new sentence is provable, it is also logically implied. (For example, we might have a rule telling us that from "A" we can derive "A or B". In III.a We assume that if A is provable it is implied. We also know that if A is provable then "A or B" is provable. We have to show that then "A or B" too is implied. We do so by appeal to the semantic definition and the assumption we just made. A is provable from G, we assume. So it is also implied by G. So any semantic valuation making all of G true makes A true. But any valuation making A true makes "A or B" true, by the defined semantics for "or". So any valuation which makes all of G true makes "A or B" true. So "A or B" is implied.) Generally, the Inductive step will consist of a lengthy but simple case-by-case analysis of all the rules of inference, showing that each "preserves" semantic implication.

By the definition of provability, there are no sentences provable other than by being a member of G, an axiom, or following by a rule; so if all of those are semantically implied, the deduction calculus is sound.

Sketch of completeness proof

(This is usually the much harder direction of proof.)

We adopt the same notational conventions as above.

We want to show: If G implies A, then G proves A. We proceed by contraposition: We show instead that if G does **not** prove A then G does **not** imply A.

I. G does not prove A. (Assumption)
II. If G does not prove A, then we can construct an (infinite) "Maximal Set", G^*, which is a superset of G and which also does not prove A.

a. Place an "ordering" on all the sentences in the language (e.g., shortest first, and equally long ones in extended alphabetical ordering), and number them E_1, E_2, …
b. Define a series G_n of sets (G_0, G_1, …) inductively:
   i. $G_0 = G$
   ii. If $G_k \cup \{E_{k+1}\}$ proves A, then $G_{k+1} = G_k$
   iii. If $G_k \cup \{E_{k+1}\}$ does **not** prove A, then $G_{k+1} = G_k \cup \{E_{k+1}\}$
c. Define G^* as the union of all the G_n. (That is, G^* is the set of all the sentences that are in any G_n.)
d. It can be easily shown that
   i. G^* contains (is a superset of) G (by (b.i));
   ii. G^* does not prove A (because if it proves A then some sentence was added to some G_n which caused it to prove 'A; but this was ruled out by definition); and
   iii. G^* is a "Maximal Set" (with respect to A): If any more sentences whatever were added to G^*, it would prove A. (Because if it were possible to add any more sentences, they should have been added when they were encountered during the construction of the G_n, again by definition)

III. If G^* is a Maximal Set (wrt A), then it is "truth-like". This means that it contains the sentence "C" only if it does not contain the sentence not-C; If it contains "C" and contains "If C then B" then it also contains "B"; and so forth.
IV. If G^* is truth-like there is a "G^*-Canonical" valuation of the language: one that makes every sentence in G^* true and everything outside G^* false while still obeying the laws of semantic composition in the language.
V. A G^*-canonical valuation will make our original set G all true, and make A false.
VI. If there is a valuation on which G are true and A is false, then G does not (semantically) imply A.

QED

Another outline for a completeness proof

If a formula is a tautology, then there is a truth table for it which shows that each valuation yields the value true for the formula. Consider such a valuation. By mathematical induction on the length of the subformulae, show that the truth or falsity of the subformula follows from the truth or falsity (as appropriate for the valuation) of each propositional variable in the subformula. Then combine the lines of the truth table together two at a time by using "(P is true implies S) implies ((P is false implies S) implies S)". Keep repeating this until all dependencies on propositional variables have been eliminated. The result is that we have proved the given tautology. Since every tautology is provable, the logic is complete.

Interpretation of a truth-functional propositional calculus

An **interpretation of a truth-functional propositional calculus** $\mathcal{P}$ is an assignment to each propositional symbol of $\mathcal{P}$ of one or the other (but not both) of the truth values truth (**T**) and falsity (**F**), and an assignment to the connective symbols of $\mathcal{P}$ of their usual truth-functional meanings. An interpretation of a truth-functional propositional calculus may also be expressed in terms of truth tables.

For n distinct propositional symbols there are 2^n distinct possible interpretations. For any particular symbol a, for example, there are $2^1 = 2$ possible interpretations:

1. a is assigned **T**, or
2. a is assigned **F**.

For the pair a, b there are $2^2 = 4$ possible interpretations:

1. both are assigned **T**,
2. both are assigned **F**,
3. a is assigned **T** and b is assigned **F**, or
4. a is assigned **F** and b is assigned **T**.

Since $\mathcal{P}$ has $\aleph_0$, that is, denumerably many propositional symbols, there are $2^{\aleph_0} = \mathfrak{c}$, and therefore uncountably many distinct possible interpretations of $\mathcal{P}$.

Interpretation of a sentence of truth-functional propositional logic

If ϕ and ψ are formulas of $\mathcal{P}$ and $\mathcal{I}$ is an interpretation of $\mathcal{P}$ then:

- A sentence of propositional logic is true under an interpretation $\mathcal{I}$ iff $\mathcal{I}$ assigns the truth value **T** to that sentence. If a sentence is true under an interpretation, then that interpretation is called a model of that sentence.
- ϕ is false under an interpretation $\mathcal{I}$ iff ϕ is not true under $\mathcal{I}$.
- A sentence of propositional logic is logically valid iff it is true under every interpretation

  $\models \phi$ means that ϕ is logically valid

- A sentence ψ of propositional logic is a semantic consequence of a sentence ϕ iff there is no interpretation under which ϕ is true and ψ is false.
- A sentence of propositional logic is consistent iff it is true under at least one interpretation. It is inconsistent if it is not consistent.

Some consequences of these definitions:

- For any given interpretation a given formula is either true or false.
- No formula is both true and false under the same interpretation.
- ϕ is false for a given interpretation iff $\neg\phi$ is true for that interpretation; and ϕ is true under an interpretation iff $\neg\phi$ is false under that interpretation.
- If ϕ and $(\phi \rightarrow \psi)$ are both true under a given interpretation, then ψ is true under that interpretation.
- If $\models_{\mathrm{P}} \phi$ and $\models_{\mathrm{P}} (\phi \rightarrow \psi)$, then $\models_{\mathrm{P}} \psi$.
- $\neg\phi$ is true under $\mathcal{I}$ iff ϕ is not true under $\mathcal{I}$.
- $(\phi \rightarrow \psi)$ is true under $\mathcal{I}$ iff either ϕ is not true under $\mathcal{I}$ or ψ is true under $\mathcal{I}$.
- A sentence ψ of propositional logic is a semantic consequence of a sentence ϕ iff $(\phi \rightarrow \psi)$ is logically valid, that is, $\phi \models_{\mathrm{P}} \psi$ iff $\models_{\mathrm{P}} (\phi \rightarrow \psi)$.

Alternative calculus

It is possible to define another version of propositional calculus, which defines most of the syntax of the logical operators by means of axioms, and which uses only one inference rule.

Axioms

Let ϕ, χ and ψ stand for well-formed formulæ. (The wffs themselves would not contain any Greek letters, but only capital Roman letters, connective operators, and parentheses.) Then the axioms are as follows:

Axioms		
Name	**Axiom Schema**	**Description**
THEN-1	$\phi \rightarrow (\chi \rightarrow \phi)$	Add hypothesis χ, implication introduction
THEN-2	$(\phi \rightarrow (\chi \rightarrow \psi)) \rightarrow ((\phi \rightarrow \chi) \rightarrow (\phi \rightarrow \psi))$	Distribute hypothesis ϕ over implication
AND-1	$\phi \wedge \chi \rightarrow \phi$	Eliminate conjunction
AND-2	$\phi \wedge \chi \rightarrow \chi$	
AND-3	$\phi \rightarrow (\chi \rightarrow (\phi \wedge \chi))$	Introduce conjunction

OR-1	$\phi \to \phi \vee \chi$	Introduce disjunction
OR-2	$\chi \to \phi \vee \chi$	
OR-3	$(\phi \to \psi) \to ((\chi \to \psi) \to (\phi \vee \chi \to \psi))$	Eliminate disjunction
NOT-1	$(\phi \to \chi) \to ((\phi \to \neg\chi) \to \neg\phi)$	Introduce negation
NOT-2	$\phi \to (\neg\phi \to \chi)$	Eliminate negation
NOT-3	$\phi \vee \neg\phi$	Excluded middle, classical logic
IFF-1	$(\phi \leftrightarrow \chi) \to (\phi \to \chi)$	Eliminate equivalence
IFF-2	$(\phi \leftrightarrow \chi) \to (\chi \to \phi)$	
IFF-3	$(\phi \to \chi) \to ((\chi \to \phi) \to (\phi \leftrightarrow \chi))$	Introduce equivalence

Axiom THEN-2 may be considered to be a "distributive property of implication with respect to implication."
Axioms AND-1 and AND-2 correspond to "conjunction elimination". The relation between AND-1 and AND-2 reflects the commutativity of the conjunction operator.
Axiom AND-3 corresponds to "conjunction introduction."
Axioms OR-1 and OR-2 correspond to "disjunction introduction." The relation between OR-1 and OR-2 reflects the commutativity of the disjunction operator.
Axiom NOT-1 corresponds to "reductio ad absurdum."
Axiom NOT-2 says that "anything can be deduced from a contradiction."
Axiom NOT-3 is called "tertium non datur" (Latin: "a third is not given") and reflects the semantic valuation of propositional formulae: a formula can have a truth-value of either true or false. There is no third truth-value, at least not in classical logic. Intuitionistic logicians do not accept the axiom NOT-3.

Inference rule

The inference rule is modus ponens:

- $\phi,\ \phi \to \chi \vdash \chi.$

Meta-inference rule

Let a demonstration be represented by a sequence, with hypotheses to the left of the turnstile and the conclusion to the right of the turnstile. Then the deduction theorem can be stated as follows:

> If the sequence
> $\phi_1,\ \phi_2,\ ...,\ \phi_n,\ \chi \vdash \psi$
> has been demonstrated, then it is also possible to demonstrate the sequence
> $\phi_1,\ \phi_2,\ ...,\ \phi_n \vdash \chi \rightarrow \psi$.

This deduction theorem (DT) is not itself formulated with propositional calculus: it is not a theorem of propositional calculus, but a theorem about propositional calculus. In this sense, it is a meta-theorem, comparable to theorems about the soundness or completeness of propositional calculus.

On the other hand, DT is so useful for simplifying the syntactical proof process that it can be considered and used as another inference rule, accompanying modus ponens. In this sense, DT corresponds to the natural conditional proof inference rule which is part of the first version of propositional calculus introduced here.

The converse of DT is also valid:

> If the sequence
> $\phi_1,\ \phi_2,\ ...,\ \phi_n \vdash \chi \rightarrow \psi$
> has been demonstrated, then it is also possible to demonstrate the sequence
> $\phi_1,\ \phi_2,\ ...,\ \phi_n,\ \chi \vdash \psi$

in fact, the validity of the converse of DT is almost trivial compared to that of DT:

> If
> $\phi_1,\ ...,\ \phi_n \vdash \chi \rightarrow \psi$
> then
> 1: $\phi_1,\ ...,\ \phi_n,\ \chi \vdash \chi \rightarrow \psi$
> 2: $\phi_1,\ ...,\ \phi_n,\ \chi \vdash \chi$
> and from (1) and (2) can be deduced
> 3: $\phi_1,\ ...,\ \phi_n,\ \chi \vdash \psi$
> by means of modus ponens, Q.E.D.

The converse of DT has powerful implications: it can be used to convert an axiom into an inference rule. For example, the axiom AND-1,

$$\vdash \phi \wedge \chi \rightarrow \phi$$

can be transformed by means of the converse of the deduction theorem into the inference rule

$$\phi \wedge \chi \vdash \phi$$

which is conjunction elimination, one of the ten inference rules used in the first version of the propositional calculus.

Example of a proof

The following is an example of a (syntactical) demonstration, involving only axioms THEN-1 and THEN-2:

Prove: $A \to A$(Reflexivity of implication).

Proof:

1. $(A \to ((B \to A) \to A)) \to ((A \to (B \to A)) \to (A \to A))$

   Axiom THEN-2 with $\phi = A$, $\chi = B \to A$, $\psi = A$

2. $A \to ((B \to A) \to A)$

   Axiom THEN-1 with $\phi = A$, $\chi = B \to A$

3. $(A \to (B \to A)) \to (A \to A)$

   From (1) and (2) by modus ponens.

4. $A \to (B \to A)$

   Axiom THEN-1 with $\phi = A$, $\chi = B$

5. $A \to A$

   From (3) and (4) by modus ponens.

Equivalence to equational logics

The preceding alternative calculus is an example of a Hilbert-style deduction system. In the case of propositional systems the axioms are terms built with logical connectives and the only inference rule is modus ponens. Equational logic as standardly used informally in high school algebra is a different kind of calculus from Hilbert systems. Its theorems are equations and its inference rules express the properties of equality, namely that it is a congruence on terms that admits substitution.

Classical propositional calculus as described above is equivalent to Boolean algebra, while intuitionistic propositional calculus is equivalent to Heyting algebra. The equivalence is shown by translation in each direction of the theorems of the respective

systems. Theorems ϕ of classical or intuitionistic propositional calculus are translated as equations $\phi = 1$ of Boolean or Heyting algebra respectively. Conversely theorems x = y of Boolean or Heyting algebra are translated as theorems $(x \to y) \wedge (y \to x)$ of classical or propositional calculus respectively, for which $x \equiv y$ is a standard abbreviation. In the case of Boolean algebra x = y can also be translated as $(x \wedge y) \vee (\neg x \wedge \neg y)$, but this translation is incorrect intuitionistically.

In both Boolean and Heyting algebra, inequality $x \leq y$ can be used in place of equality. The equality x = y is expressible as a pair of inequalities $x \leq y$ and $y \leq x$. Conversely the inequality $x \leq y$ is expressible as the equality $x \wedge y = x$, or as $x \vee y = y$. The significance of inequality for Hilbert-style systems is that it corresponds to the latter's deduction or entailment symbol $\vdash$. An entailment

$$\phi_1,\ \phi_2,\ \ldots,\ \phi_n \vdash \psi$$

is translated in the inequality version of the algebraic framework as

$$\phi_1\ \wedge\ \phi_2\ \wedge\ \ldots\ \wedge\ \phi_n\ \ \leq\ \ \psi$$

Conversely the algebraic inequality $x \leq y$ is translated as the entailment

$$x\ \vdash\ y.$$

The difference between implication $x \to y$ and inequality or entailment $x \leq y$ or $x\ \vdash\ y$ is that the former is internal to the logic while the latter is external. Internal implication between two terms is another term of the same kind. Entailment as external implication between two terms expresses a metatruth outside the language of the logic, and is considered part of the metalanguage. Even when the logic under study is intuitionistic, entailment is ordinarily understood classically as two-valued: either the left side entails, or is less-or-equal to, the right side, or it is not.

Similar but more complex translations to and from algebraic logics are possible for natural deduction systems as described above and for the sequent calculus. The entailments of the latter can be interpreted as two-valued, but a more insightful interpretation is as a set, the elements of which can be understood as abstract proofs organized as the morphisms of a category. In this interpretation the cut rule of the sequent calculus corresponds to composition in the category. Boolean and Heyting algebras enter this picture as special categories having at most one morphism per homset, i.e. one proof per entailment, corresponding to the idea that existence of proofs is all that matters: any proof will do and there is no point in distinguishing them.

Graphical calculi

It is possible to generalize the definition of a formal language from a set of finite sequences over a finite basis to include many other sets of mathematical structures, so long as they are built up by finitary means from finite materials. What's more, many of these families of formal structures are especially well-suited for use in logic.

For example, there are many families of graphs that are close enough analogues of formal languages that the concept of a calculus is quite easily and naturally extended to them. Indeed, many species of graphs arise as parse graphs in the syntactic analysis of the corresponding families of text structures. The exigencies of practical computation on formal languages frequently demand that text strings be converted into pointer structure renditions of parse graphs, simply as a matter of checking whether strings are wffs or not. Once this is done, there are many advantages to be gained from developing the graphical analogue of the calculus on strings. The mapping from strings to parse graphs is called parsing and the inverse mapping from parse graphs to strings is achieved by an operation that is called traversing the graph.

Other logical calculi

Propositional calculus is about the simplest kind of logical calculus in any current use. (Aristotelian "syllogistic" calculus, which is largely supplanted in modern logic, is in some ways simpler — but in other ways more complex — than propositional calculus.) It can be extended in several ways.

The most immediate way to develop a more complex logical calculus is to introduce rules that are sensitive to more fine-grained details of the sentences being used. When the "atomic sentences" of propositional logic are broken up into terms, variables, predicates, and quantifiers, they yield first-order logic, or first-order predicate logic, which keeps all the rules of propositional logic and adds some new ones. (For example, from "All dogs are mammals" we may infer "If Rover is a dog then Rover is a mammal".) It makes sense to refer to propositional logic as "zeroth-order logic", when comparing it with first-order logic and second-order logic.

With the tools of first-order logic it is possible to formulate a number of theories, either with explicit axioms or by rules of inference, that can themselves be treated as logical calculi. Arithmetic is the best known of these; others include set theory and mereology.

Modal logic also offers a variety of inferences that cannot be captured in propositional calculus. For example, from "Necessarily p" we may infer that p. From p we may infer "It is possible that p". The translation between modal logics and algebraic logics is as for classical and intuitionistic logics but with the introduction of a unary operator on Boolean or Heyting algebras, different from the Boolean operations, interpreting the possibility modality, and in the case of Heyting algebra a second operator interpreting necessity (for Boolean algebra this is redundant since necessity is the De Morgan dual of possibility).

The first operator preserves 0 and disjunction while the second preserves 1 and conjunction.

Many-valued logics are those allowing sentences to have values other than true and false. (For example, neither and both are standard "extra values"; "continuum logic" allows each sentence to have any of an infinite number of "degrees of truth" between true and false.) These logics often require calculational devices quite distinct from propositional calculus. When the values form a Boolean algebra (which may have more than two or even infinitely many values), many-valued logic reduces to classical logic; many-valued logics are therefore only of independent interest when the values form an algebra that is not Boolean.

Solvers

Finding solutions to propositional logic formulas is an NP-complete problem. However, practical methods exist (e.g. DPLL algorithm, 1962; Chaff algorithm, 2001) that are very fast for many useful cases. Recent work has extended the SAT solver algorithms to work with propositions containing arithmetic expressions; these are the SMT solvers.

Chapter- 3

Modal Logic

Modal logic is a type of formal logic that extends the standards of formal logic to include the elements of modality (for example, possibility and necessity). Modals qualify the truth of a judgment. For example, if it is true that "John is happy," we might qualify this statement by saying that "John is very happy," in which case the term "very" would be a modality. Traditionally, there are three "modes" or "moods" or "modalities" represented in modal logic, namely, possibility, probability, and necessity.

A formal modal logic represents modalities using modal operators. For example, "It might rain today" and "It is possible that rain will fall today" both contain the notion of possibility. In a modal logic this is represented as an operator, Possibly, attached to the sentence It will rain today.

The basic unary (1-place) modal operators are usually written $\Box$for Necessarily and $\Diamond$for Possibly. In a classical modal logic, each can be expressed by the other with negation:

$$\Diamond P \leftrightarrow \neg\Box\neg P;$$
$$\Box P \leftrightarrow \neg\Diamond\neg P.$$

Thus it is possible that it will rain today if and only if it is not necessary that it will not rain today;
and it is necessary that it will rain today if and only if it is not possible that it will not rain today.

Development of modal logic

Although Aristotle's logic is almost entirely concerned with the theory of the categorical syllogism, there are passages in his work, such as the famous sea-battle argument in De Interpretatione § 9, that are now seen as anticipations of modal logic and its connection with potentiality and time. Modal logic as a self-aware subject owes much to the writings of the Scholastics, in particular William of Ockham and John Duns Scotus, who reasoned informally in a modal manner, mainly to analyze statements about essence and accident.

C. I. Lewis founded modern modal logic in his 1910 Harvard thesis and in a series of scholarly articles beginning in 1912. This work culminated in his 1932 book Symbolic Logic (with C. H. Langford), which introduced the five systems S1 through S5.

Ruth C. Barcan (later Ruth Barcan Marcus) developed the first axiomatic systems of quantified modal logic — first and second order extensions of Lewis's "S2", "S4", and "S5".

The contemporary era in modal semantics began in 1959, when Saul Kripke (then only a 19 year old Harvard University undergraduate) introduced the now-standard Kripke semantics for modal logics. These are commonly referred to as "possible worlds" semantics. Kripke and A. N. Prior had previously corresponded at some length.

A. N. Prior created modern temporal logic, closely related to modal logic, in 1957 by adding modal operators [F] and [P] meaning "henceforth" and "hitherto". Vaughan Pratt introduced dynamic logic in 1976. In 1977, Amir Pnueli proposed using temporal logic to formalise the behaviour of continually operating concurrent programs. Flavors of temporal logic include propositional dynamic logic (PDL), propositional linear temporal logic (PLTL), linear temporal logic (LTL), computational tree logic (CTL), Hennessy–Milner logic, and T.

The mathematical structure of modal logic, namely Boolean algebras augmented with unary operations (often called "modal algebras"), began to emerge with J. C. C. McKinsey's 1941 proof that S2 and S4 are decidable, and reached full flower in the work of Alfred Tarski and his student Bjarni Jonsson (Jonsson and Tarski 1951–52). This work revealed that S4 and S5 are models of interior algebra, a proper extension of Boolean algebra originally designed to capture the properties of the interior and closure operators of topology. Texts on modal logic typically do little more than mention its connections with the study of Boolean algebras and topology. For a thorough survey of the history of formal modal logic and of the associated mathematics.

Formalizations

Semantics

The semantics for modal logic are usually given like so: First we define a frame, which consists of a non-empty set, G, whose members are generally called possible worlds, and a binary relation, R, that holds (or not) between the possible worlds of G. This binary relation is called the accessibility relation. For example, w R v means that the world v is accessible from world w. That is to say, the state of affairs known as v is a live possibility for w. This gives a pair, <G, R>.

Next, the frame is extended to a model by specifying the truth-values of all propositions at each of the worlds in G. We do so by defining a relation $\vDash$ between possible worlds

and propositional letters. If there is a world w such that w ⊨ P, then P is true at w. A model is thus an ordered triple, <G, R, ⊨>.

Then we define truth in a model:

- w ⊨ ¬P if and only if w ⊭ P
- w ⊨ (P & Q) if and only if w ⊨ P and w ⊨ Q
- w ⊨ □P if and only if for every element v of G, if w R v then v ⊨ P
- w ⊨ ◊P if and only if for some element v of G, it holds that w R v and v ⊨ P

According to these semantics, a truth is necessary with respect to a possible world w if it is true at every world that is accessible to w, and possible if it is true at some world that is accessible to w. Possibility thereby depends upon the accessibility relation R, which allows us to express the relative nature of possibility. For example, we might say that given our laws of physics it is not possible for humans to travel faster than the speed of light, but that given other circumstances it could have been possible to do so. Using the accessibility relation we can translate this scenario as follows: At all of the worlds accessible to our own world, it is not the case that humans can travel faster than the speed of light, but at one of these accessible worlds there is another world accessible from those worlds but not accessible from our own at which humans can travel faster than the speed of light.

It should also be noted that the definition of □makes vacuously true certain sentences, since when it speaks of "every world that is accesible to w" it takes for granted the usual mathematical interpretation of the word "every". Hence, if a world w doesn't have any accesible worlds, any sentence beginning with □is true.

The different systems of modal logic are distinguished by the properties of their corresponding accessibility relations. There are several systems that have been espoused (often called frame conditions). An accessibility relation is:

- reflexive iff w R w, for every w in G
- symmetric iff w R v implies v R w, for all w and v in G
- transitive iff w R v and v R q together imply w R q, for all w, v, q in G.
- serial iff, for each w in G there is some v in G such that w R v.

The logics that stem from these frame conditions are:

- K := no conditions
- D := serial
- T := reflexive
- S4 := reflexive and transitive
- S5 := reflexive, symmetric and transitive

S5 is the strongest logic. A sentence is S5-valid if it is valid in all frames where R is an equivalence relation (reflexive, symmetric and transitive). We can prove that these frames produce the same set of valid sentences as do any frames where all worlds can see all other worlds (i.e., where R is a "total" relation). Since all worlds can see all other worlds in S5, the S5 system's semantics can be defined without the use of an accessibility relation, R.

For example, in S4:

> w ⊨ ◇P if and only if for some element v of G, it holds that v ⊨ P and w R v.

However, in S5, we can just say that

> w ⊨ ◇P if and only if for some element v of G, it holds that v ⊨ P.

We can drop the accessibility clause from the latter stipulation because it is trivially true of all S5 frames that w R v.

All of these logical systems can also be defined axiomatically, as is shown in the next section. For example, in S5, the axioms P → □◇P, □P → □□P, and □P → P (corresponding to symmetry, transitivity and reflexivity, respectively) hold, whereas at least one of these axioms does not hold in each of the other, weaker logics.

Axiomatic systems

The first formalizations of modal logic were axiomatic. Numerous variations with very different properties have been proposed since C. I. Lewis began working in the area in 1910. Hughes and Cresswell (1996), for example, describe 42 normal and 25 non-normal modal logics. Zeman (1973) describes some systems Hughes and Cresswell omit.

Modern treatments of modal logic begin by augmenting the propositional calculus with two unary operations, one denoting "necessity" and the other "possibility". The notation of Lewis, much employed since, denotes "necessarily p" by a prefixed "box" ($\Box p$) whose scope is established by parentheses. Likewise, a prefixed "diamond" ($\Diamond p$) denotes "possibly p". Regardless of notation, each of these operators is definable in terms of the other:

- $\Box p$(necessarily p) is equivalent to $\neg\Diamond\neg p$("not possible that not-p")
- $\Diamond p$(possibly p) is equivalent to $\neg\Box\neg p$("not necessarily not-p")

Hence □and ◇form a dual pair of operators.

In many modal logics, the necessity and possibility operators satisfy the following analogs of de Morgan's laws from Boolean algebra:

"It is **not necessary that** X" is logically equivalent to "It is **possible that not** X".
"It is **not possible that** X" is logically equivalent to "It is **necessary that not** X".

Precisely what axioms and rules must be added to the propositional calculus to create a usable system of modal logic is a matter of philosophical opinion, often driven by the theorems one wishes to prove; or, in computer science, it is a matter of what sort of computational or deductive system one wishes to model. Many modal logics, known collectively as normal modal logics, include the following rule and axiom:

- **N**, Necessitation Rule: If p is a theorem (of any system invoking **N**), then $\Box p$ is likewise a theorem.
- **K**, Distribution Axiom: $\Box(p \to q) \to (\Box p \to \Box q)$.

The weakest normal modal logic, named K in honor of Saul Kripke, is simply the propositional calculus augmented by $\Box$, the rule **N**, and the axiom **K**. K is weak in that it fails to determine whether a proposition can be necessary but only contingently necessary. That is, it is not a theorem of K that if $\Box p$ is true then $\Box\Box p$ is true, i.e., that necessary truths are "necessarily necessary". If such perplexities are deemed forced and artificial, this defect of K is not a great one. In any case, different answers to such questions yield different systems of modal logic.

Adding axioms to K gives rise to other well-known modal systems. One cannot prove in K that if "p is necessary" then p is true. The axiom **T** remedies this defect:

- **T**, Reflexivity Axiom: $\Box p \to p$ (If p is necessary, then p is the case.) **T** holds in most but not all modal logics. Zeman (1973) describes a few exceptions, such as $S1^0$.

Other well-known elementary axioms are:

- **4**: $\Box p \to \Box\Box p$
- **B**: $p \to \Box\Diamond p$
- **D**: $\Box p \to \Diamond p$

These yield the systems (axioms in bold):

- K := **K** + **N**
- T := K + **T**
- S4 := T + **4**
- S5 := S4 + **B**
- D := K + **D**.

K through S5 form a nested hierarchy of systems, making up the core of normal modal logic. But specific rules or sets of rules may be appropriate for specific systems. For example, in deontic logic, $\Box p \to \Diamond p$ (If it ought to be that p, then it is permitted that p)

seems appropriate, but we should probably not include that $p \rightarrow \Box\Diamond p$. In fact, to do so is to commit the naturalistic fallacy.

The commonly employed system S5 simply makes all modal truths necessary. For example, if p is possible, then it is "necessary" that p is possible. Also, if p is necessary, then it is necessary that p is necessary. Other systems of modal logic have been formulated, in part because S5 does not describe every kind of modality of interest.

Alethic logic

Modalities of necessity and possibility are called alethic modalities. They are also sometimes called special modalities, from the Latin species. Modal logic was first developed to deal with these concepts, and only afterward was extended to others. For this reason, or perhaps for their familiarity and simplicity, necessity and possibility are often casually treated as the subject matter of modal logic. Moreover it is easier to make sense of relativizing necessity, e.g. to legal, physical, nomological, epistemic, and so on, than it is to make sense of relativizing other notions.

In classical modal logic, a proposition is said to be

- **possible** if and only if it is not necessarily false (regardless of whether it is actually true or actually false);
- **necessary** if and only if it is not possibly false; and
- **contingent** if and only if it is not necessarily false and not necessarily true (i.e. possible but not necessarily true).

In classical modal logic, therefore, either the notion of possibility or necessity may be taken to be basic, where these other notions are defined in terms of it in the manner of De Morgan duality. Intuitionistic modal logic treats possibility and necessity as not perfectly symmetric.

For those with difficulty with the concept of something being possible but not true, the meaning of these terms may be made more comprehensible by thinking of multiple "possible worlds" (in the sense of Leibniz) or "alternate universes"; something "necessary" is true in all possible worlds, something "possible" is true in at least one possible world. These "possible world semantics" are formalized with Kripke semantics.

Physical possibility

Something is physically possible if it is permitted by the laws of physics. For example, current theory allows for there to be an atom with an atomic number of 150, though there may not in fact be any such atoms in existence. Similarly, while it is logically possible to accelerate beyond the speed of light, modern science stipulates that it is not physically possible for material particles or information.

Metaphysical possibility

Philosophers ponder the properties that objects have independently of those dictated by scientific laws. For example, it might be metaphysically necessary, as some have thought, that all thinking beings have bodies and can experience the passage of time, or that God exists (or does not). Saul Kripke has argued that every person necessarily has the parents they do have: anyone with different parents would not be the same person.

Metaphysical possibility is generally thought to be more restricting than bare logical possibility (i.e., fewer things are metaphysically possible than are logically possible). Its exact relation to physical possibility is a matter of some dispute. Philosophers also disagree over whether metaphysical truths are necessary merely "by definition", or whether they reflect some underlying deep facts about the world, or something else entirely.

Confusion with epistemic modalities

Alethic modalities and epistemic modalities (see below) are often expressed in English using the same words. "It is possible that bigfoot exists" can mean either "Bigfoot could exist, whether or not bigfoot does in fact exist" (alethic), or more likely, "For all I know, bigfoot exists" (epistemic).

Epistemic logic

Epistemic modalities (from the Greek episteme, knowledge), deal with the certainty of sentences. The $\Box$operator is translated as "x knows that…", and the $\Diamond$operator is translated as "For all x knows, it may be true that…" In ordinary speech both metaphysical and epistemic modalities are often expressed in similar words; the following contrasts may help:

A person, Jones, might reasonably say both: (1) "No, it is not possible that Bigfoot exists; I am quite certain of that"; and, (2) "Sure, Bigfoot possibly could exist". What Jones means by (1) is that given all the available information, there is no question remaining as to whether Bigfoot exists. This is an epistemic claim. By (2) he makes the metaphysical claim that it is possible for Bigfoot to exist, even though he does not (which is not equivalent to "it is possible that Bigfoot exists – for all I know", which contradicts (1)).

From the other direction, Jones might say, (3) "It is possible that Goldbach's conjecture is true; but also possible that it is false", and also (4) "if it is true, then it is necessarily true, and not possibly false". Here Jones means that it is epistemically possible that it is true or false, for all he knows (Goldbach's conjecture has not been proven either true or false), but if there is a proof (heretofore undiscovered), then it would show that it is not logically possible for Goldbach's conjecture to be false—there could be no set of numbers that violated it. Logical possibility is a form of alethic possibility; (4) makes a claim about whether it is possible (i.e., logically speaking) that a mathematical truth to have been

false, but (3) only makes a claim about whether it is possible, for all Jones knows, (i.e., speaking of certitude) that the mathematical claim is specifically either true or false, and so again Jones does not contradict himself. It is worthwhile to observe that Jones is not necessarily correct: It is possible (epistemically) that Goldbach's conjecture is both true and unprovable.

Epistemic possibilities also bear on the actual world in a way that metaphysical possibilities do not. Metaphysical possibilities bear on ways the world might have been, but epistemic possibilities bear on the way the world may be (for all we know). Suppose, for example, that I want to know whether or not to take an umbrella before I leave. If you tell me "it is possible that it is raining outside" – in the sense of epistemic possibility – then that would weigh on whether or not I take the umbrella. But if you just tell me that "it is possible for it to rain outside" – in the sense of metaphysical possibility – then I am no better off for this bit of modal enlightenment.

Some features of epistemic modal logic are in debate. For example, if x knows that p, does x know that it knows that p? That is to say, should $\Box P \rightarrow \Box\Box P$ be an axiom in these systems? While the answer to this question is unclear, there is at least one axiom that must be included in epistemic modal logic, because it is minimally true of all modal logics:

- **K**, Distribution Axiom: $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$.

But this is disconcerting, because with **K**, we can prove that we know all the logical consequences of our beliefs: If q is a logical consequence of p, then $\Box(p \rightarrow q)$. And if so, then we can deduce that $(\Box p \rightarrow \Box q)$ using **K**. When we translate this into epistemic terms, this says that if q is a logical consequence of p, then we know that it is, and if we know p, we know q. That is to say, we know all the logical consequences of our beliefs. This must be true for all possible Kripkean modal interpretations of epistemic cases where $\Box$ is translated as "knows that". But then, for example, if x knows that prime numbers are divisible only by themselves and the number one, then x knows that 86833176188118864955181944012799999999 is prime (since this number is only divisible by itself and the number one). That is to say, under the modal interpretation of knowledge, anyone who knows the definition of a prime number knows that this number is prime. This shows that epistemic modal logic is an idealized account of knowledge, and explains objective, rather than subjective knowledge (if anything).

Temporal logic

Temporal logic is an approach to the semantics of expressions with tense, that is, expressions with qualifications of when. Some expressions, such as '2 + 2 = 4', are true at all times, while tensed expressions such as 'John is happy' are only true sometimes.

In temporal logic, tense constructions are treated in terms of modalities, where a standard method for formalizing talk of time is to use two pairs of operators, one for the past and one for the future (P will just mean 'it is presently the case that P'). For example:

FP : It will sometime be the case that P
GP : It will always be the case that P
PP : It was sometime the case that P
HP : It has always been the case that P

There are then at least three modal logics that we can develop. For example, we can stipulate that,

$\Diamond P$= P is the case at some time t
$\Box P$= P is the case at every time t

Or we can trade these operators to deal only with the future (or past). For example,

$\Diamond_1 P$= **F**P
$\Box_1 P$= **G**P

or,

$\Diamond_2 P$= P and/or **F**P
$\Box_2 P$= P and **G**P

The operators **F** and **G** may seem initially foreign, but they create normal modal systems. Note that **F**P is the same as ¬**G**¬P. We can combine the above operators to form complex statements. For example, **P**P → □**P**P says (effectively), Everything that is past and true is necessary.

It seems reasonable to say that possibly it will rain tomorrow, and possibly it won't; on the other hand, seeing as how we can't change the past, if it is true that it rained yesterday, it probably isn't true that it may not have rained yesterday. It seems the past is "fixed", or necessary, in a way the future is not. This is sometimes referred to as accidental necessity. But if the past is "fixed", and everything that is in the future will eventually be in the past, then it seems plausible to say that future events are necessary too.

Similarly, the problem of future contingents considers the semantics of assertions about the future: is either of the propositions 'There will be a sea battle tomorrow', or 'There will not be a sea battle tomorrow' now true? Considering this thesis led Aristotle to reject the principle of bivalence for assertions concerning the future.

Additional binary operators are also relevant to temporal logics, q.v. Linear Temporal Logic.

Versions of temporal logic can be used in computer science to model computer operations and prove theorems about them. In one version, $\Diamond P$ means "at a future time in the computation it is possible that the computer state will be such that P is true"; $\Box P$ means "at all future times in the computation P will be true". In another version, $\Diamond P$ means "at the immediate next state of the computation, P might be true"; $\Box P$ means "at the immediate next state of the computation, P will be true". These differ in the choice of Accessibility relation. (P always means "P is true at the current computer state".) These two examples involve nondeterministic or not-fully-understood computations; there are many other modal logics specialized to different types of program analysis. Each one naturally leads to slightly different axioms.

Deontic logic

Likewise talk of morality, or of obligation and norms generally, seems to have a modal structure. The difference between "You must do this" and "You may do this" looks a lot like the difference between "This is necessary" and "This is possible". Such logics are called deontic, from the Greek for "duty".

Deontic logics commonly lack the axiom **T** semantically corresponding to the reflexivity of the accessibility relation in Kripke semantics: in symbols, $\Box\phi \rightarrow \phi$. Interpreting $\Box$ as "it is obligatory that", **T** informally says that every obligation is true. For example, if it is obligatory not to kill others (i.e. killing is morally forbidden), then **T** implies that people actually do not kill others. The consequent is obviously false.

Instead, using Kripke semantics, we say that though our own world does not realize all obligations, the worlds accessible to it do (i.e., **T** holds at these worlds). These worlds are called idealized worlds. P is obligatory with respect to our own world if at all idealized worlds accessible to our world, P holds. Though this was one of the first interpretations of the formal semantics, it has recently come under criticism.

One other principle that is often (at least traditionally) accepted as a deontic principle is D, $\Box\phi \rightarrow \Diamond\phi$, which corresponds to the seriality (or extendability or unboundedness) of the accessibility relation. It is an embodiment of the Kantian idea that "ought implies can". (Clearly the "can" can be interpreted in various senses, e.g. in a moral or alethic sense.)

Intuitive problems with deontic logic

When we try and formalize ethics with standard modal logic, we run into some problems. Suppose that we have a proposition K: you kill the victim, and another, Q: you kill the victim quickly. Now suppose we want to express the thought that "if you do kill the victim, you ought to kill him quickly". There are two likely candidates,

(1) $(K \rightarrow \Box Q)$

(2) $\Box(K \rightarrow Q)$

But (1) says that if you kill the victim, then it ought to be the case that you kill him quickly. This surely isn't right, because you ought not to have killed him at all. And (2) doesn't work either. If the right representation of "if you kill the victim then you ought to kill him quickly" is (2), then the right representation of (3) "if you kill the victim then you ought to kill him slowly" is $\Box(K \rightarrow \neg Q)$. Now suppose (as seems reasonable) that you should not kill the victim, or $\Box\neg K$. But then we can deduce $\Box(K \rightarrow \neg Q)$, which would express sentence (3). So if you should not kill the victim, then if you kill him, you should kill him slowly. But that can't be right, and is not right when we use natural language. Telling someone they should not kill the victim certainly does not imply that they should kill the victim slowly if they do kill him.

Doxastic logic

Doxastic logic concerns the logic of belief (of some set of agents). The term doxastic is derived from the ancient Greek doxa which means "belief". Typically, a doxastic logic uses $\Box$, often written "B", to mean "It is believed that", or when relativized to a particular agent s, "It is believed by s that".

Other modal logics

Significantly, modal logics can be developed to accommodate most of these idioms; it is the fact of their common logical structure (the use of "intensional" sentential operators) that make them all varieties of the same thing.

The ontology of possibility

In the most common interpretation of modal logic, one considers "logically possible worlds". If a statement is true in all possible worlds, then it is a necessary truth. If a statement happens to be true in our world, but is not true in all possible worlds, then it is a contingent truth. A statement that is true in some possible world (not necessarily our own) is called a possible truth.

Under this "possible worlds idiom," to maintain that Bigfoot's existence is possible but not actual, one says, "There is some possible world in which Bigfoot exists; but in the actual world, Bigfoot does not exist". However, it is unclear what this claim commits us to. Are we really alleging the existence of possible worlds, every bit as real as our actual world, just not actual? Saul Kripke believes that 'possible world' is something of a misnomer – that the term 'possible world' is just a useful way of visualizing the concept of possibility. For him, the sentences "you could have rolled a 4 instead of a 6" and "there is a possible world where you rolled a 4, but you rolled a 6 in the actual world" are not significantly different statements, and neither commit us to the existence of a possible world. David Lewis, on the other hand, made himself notorious by biting the bullet,

asserting that all merely possible worlds are as real as our own, and that what distinguishes our world as actual is simply that it is indeed our world – this world. That position is a major tenet of "modal realism". Some philosophers decline to endorse any version of modal realism, considering it ontologically extravagant, and prefer to seek various ways to paraphrase away these ontological commitments. Robert Adams holds that 'possible worlds' are better thought of as 'world-stories', or consistent sets of propositions. Thus, it is possible that you rolled a 4 if such a state of affairs can be described coherently.

Computer scientists will generally pick a highly specific interpretation of the modal operators specialized to the particular sort of computation being analysed. In place of "all worlds", you may have "all possible next states of the computer", or "all possible future states of the computer".

Chapter- 4

First-Order Logic

First-order logic is a formal logical system used in mathematics, philosophy, linguistics, and computer science. It goes by many names, including: **first-order predicate calculus**, the **lower predicate calculus**, **quantification theory**, and predicate logic. First-order logic is distinguished from propositional logic by its use of quantifiers; each interpretation of first-order logic includes a domain of discourse over which the quantifiers range. Briefly, first-order logic is distinguished from higher-order logics in that quantification is allowed only over atomic entities (individuals but not sets).

There are many deductive systems for first-order logic that are sound (only deriving correct results) and complete (able to derive any logically valid implication). Although the logical consequence relation is only semidecidable, much progress has been made in automated theorem proving in first-order logic. First-order logic also satisfies several metalogical theorems that make it amenable to analysis in proof theory, such as the Löwenheim–Skolem theorem and the compactness theorem.

First-order logic is of great importance to the foundations of mathematics, where it has become the standard formal logic for axiomatic systems. It has sufficient expressive power to formalize two important mathematical theories: Zermelo–Fraenkel set theory (ZF) and first-order Peano arithmetic. However, no axiom system in first order logic is strong enough to fully (categorically) describe infinite structures such as the natural numbers or the real line. Categorical axiom systems for these structures can be obtained in stronger logics such as second-order logic.

A history of first-order logic and an account of its emergence over other formal logics is provided by Ferreirós (2001).

Introduction

While propositional logic deals with simple declarative propositions, first-order logic additionally covers predicates and quantification.

A predicate resembles a function that returns either True or False. Consider the following sentences: "Socrates is a philosopher", "Plato is a philosopher". In propositional logic these are treated as two unrelated propositions, denoted for example by p and q. In first-

order logic, however, the sentences can be expressed in a more parallel manner using the predicate Phil(a), which asserts that the object represented by a is a philosopher. Thus if a represents Socrates then Phil(a) asserts the first proposition, p; if a instead represents Plato then Phil(a) asserts the second proposition, q. A key aspect of first-order logic is visible here: the string "Phil" is a syntactic entity which is given semantic meaning by declaring that Phil(a) holds exactly when a is a philosopher. An assignment of semantic meaning is called an interpretation.

First-order logic allows reasoning about properties that are shared by many objects, through the use of variables. For example, let Phil(a) assert that a is a philosopher and let Schol(a) assert that a is a scholar. Then the formula

$$\mathrm{Phil}(a) \rightarrow \mathrm{Schol}(a)$$

asserts that if a is a philosopher then a is a scholar. The symbol $\rightarrow$is used to denote a conditional (if/then) statement. The hypothesis lies to the left of the arrow and the conclusion to the right. The truth of this formula depends on which object is denoted by a, and on the interpretations of "Phil" and "Schol".

Assertions of the form "for every a, if a is a philosopher then a is a scholar" require both the use of variables and the use of a quantifier. Again, let Phil(a) assert a is a philosopher and let Schol(a) assert that a is a scholar. Then the first-order sentence

$$\forall a(\mathrm{Phil}(a) \rightarrow \mathrm{Schol}(a))$$

asserts that no matter what a represents, if a is a philosopher then a is scholar. Here $\forall$, the universal quantifier, expresses the idea that the claim in parentheses holds for all choices of a.

To show that the claim "If a is a philosopher then a is a scholar" is false, one would show there is some philosopher who is not a scholar. This counterclaim can be expressed with the existential quantifier $\exists$:

$$\exists a(\mathrm{Phil}(a) \wedge \neg \mathrm{Schol}(a)).$$

Here:

- $\neg$is the negation operator: $\neg \mathrm{Schol}(a)$is true if and only if $\mathrm{Schol}(a)$is false, in other words if and only if a is not a scholar.
- $\wedge$is the conjunction operator: $\mathrm{Phil}(a) \wedge \neg \mathrm{Schol}(a)$asserts that a is a philosopher and also not a scholar.

The predicates Phil(a) and Schol(a) take only one parameter each. First-order logic can also express predicates with more than one parameter. For example, "there is someone who can be fooled every time" can be expressed as:

$$\exists x(\mathrm{Person}(x) \wedge \forall y(\mathrm{Time}(y) \rightarrow \mathrm{Canfool}(x, y))).$$

Here Person(x) is interpreted to mean x is a person, Time(y) to mean that y is a moment of time, and Canfool(x,y) to mean that (person) x can be fooled at (time) y. For clarity, this statement asserts that there is one person who can be fooled at all times, which is stronger than asserting that at all times at least one person exists who can be fooled. Asserting the latter (that there is always at least one foolable person) does not signify whether this foolable person is always the same for all moments of time.

The **range** of the quantifiers is the set of objects that can be used to satisfy them. (In the informal examples in this section, the range of the quantifiers was left unspecified.) In addition to specifying the meaning of predicate symbols such as Person and Time, an interpretation must specify a nonempty set, known as the domain of discourse or universe, as a range for the quantifiers. Thus a statement of the form $\exists a \mathrm{Phil}(a)$ is said to be true, under a particular interpretation, if there is some object in the domain of discourse of that interpretation that satisfies the predicate that the interpretation uses to assign meaning to the symbol Phil.

Syntax

There are two key parts of first order logic. The syntax determines which collections of symbols are legal expressions in first-order logic, while the semantics determine the meanings behind these expressions.

Alphabet

Unlike natural languages, such as English, the language of first-order logic is completely formal, so that it can be mechanically determined whether a given expression is legal. There are two key types of legal expressions: **terms**, which intuitively represent objects, and **formulas**, which intuitively express predicates that can be true or false. The terms and formulas of first-order logic are strings of **symbols** which together form the **alphabet** of the language. As with all formal languages, the nature of the symbols themselves is outside the scope of formal logic; they are often regarded simply as letters and punctuation symbols.

It is common to divide the symbols of the alphabet into **logical symbols**, which always have the same meaning, and **non-logical symbols**, whose meaning varies by interpretation. For example, the logical symbol $\wedge$ always represents "and"; it is never interpreted as "or". On the other hand, a non-logical predicate symbol such as Phil(x) could be interpreted to mean "x is a philosopher", "x is a cat", or any other unary predicate, depending on the interpretation at hand.

Logical symbols

There are several logical symbols in the alphabet, which vary by author but usually include:

- The quantifier symbols $\forall$ and $\exists$
- The logical connectives: $\wedge$ for conjunction, $\vee$ for disjunction, $\rightarrow$ for implication, $\leftrightarrow$ for biconditional, $\neg$ for negation. Occasionally other logical connective symbols are included. Some authors use $\Rightarrow$ and $\Leftrightarrow$ instead of $\rightarrow$ and $\leftrightarrow$, especially in contexts where $\rightarrow$ is used for other purposes. Moreover, $\supset$, tilde (~) and & may replace $\rightarrow$, $\neg$ and $\wedge$, especially if these symbols are not available for technical reasons.
- Parentheses, brackets, and other punctuation symbols. The choice of such symbols varies depending on context.
- An infinite set of **variables**, often denoted by lowercase letters at the end of the alphabet x, y, z, … . Subscripts are often used to distinguish variables: x_0, x_1, x_2, … .
- An **equality symbol** (sometimes, **identity symbol**) =.

It should be noted that not all of these symbols are required - only one of the quantifiers, negation and conjunction, variables, brackets and equality suffice. There are numerous minor variations that may define additional logical symbols:

- Sometimes the truth constants T or $\top$ for "true" and F or $\bot$ for "false" are included. Without any such logical operators of valence 0, these two constants can only be expressed using quantifiers.
- Sometimes additional logical connectives, such as the Sheffer stroke (NAND) and exclusive or operators are included.

Non-logical symbols

The non-logical symbols represent predicates (relations), functions and constants on the domain of discourse. It used to be standard practice to use a fixed, infinite set of non-logical symbols for all purposes. A more recent practice is to use different non-logical symbols according to the application one has in mind. Therefore it has become necessary to name the set of all non-logical symbols used in a particular application. This choice is made via a **signature**.

The traditional approach is to have only one, infinite, set of non-logical symbols (one signature) for all applications. Consequently, under the traditional approach there is only one language of first-order logic. This approach is still common, especially in philosophically oriented books.

1. For every integer $n \geq 0$ there is a collection of n-**ary**, or n-**place**, **predicate symbols**. Because they represent relations between n elements, they are also called **relation symbols**. For each arity n we have an infinite supply of them:

$P^n_0, P^n_1, P^n_2, P^n_3, \ldots$

2. For every integer $n \geq 0$ there are infinitely many n-ary **function symbols**:

   $f^n_0, f^n_1, f^n_2, f^n_3, \ldots$

In contemporary mathematical logic, the signature varies by application. Typical signatures in mathematics are {1, ×} or just {×} for groups, or {0, 1, +, ×, <} for ordered fields. There are no restrictions on the number of non-logical symbols. The signature can be empty, finite, or infinite, even uncountable. Uncountable signatures occur for example in modern proofs of the Löwenheim-Skolem theorem.

In this approach, every non-logical symbol is of one of the following types.

1. A **predicate symbol** (or **relation symbol**) with some **valence** (or **arity**, number of arguments) greater than or equal to 0. These which are often denoted by uppercase letters P, Q, R,... .
   - Relations of valence 0 can be identified with propositional variables. For example, P, which can stand for any statement.
   - For example, P(x) is a predicate variable of valence 1. One possible interpretation is "x is a man".
   - Q(x,y) is a predicate variable of valence 2. Possible interpretations include "x is greater than y" and "x is the father of y".
2. A **function symbol**, with some valence greater than or equal to 0. These are often denoted by lowercase letters f, g, h,... .
   - Examples: f(x) may be interpreted as for "the father of x". In arithmetic, it may stand for "-x". In set theory, it may stand for "the power set of x". In arithmetic, g(x,y) may stand for "x+y". In set theory, it may stand for "the union of x and y".
   - Function symbols of valence 0 are called **constant symbols**, and are often denoted by lowercase letters at the beginning of the alphabet a, b, c,... . The symbol a may stand for Socrates. In arithmetic, it may stand for 0. In set theory, such a constant may stand for the empty set.

The traditional approach can be recovered in the modern approach by simply specifying the "custom" signature to consist of the traditional sequences of non-logical symbols.

Formation rules

The formation rules define the terms and formulas of first order logic. When terms and formulas are represented as strings of symbols, these rules can be used to write a formal grammar for terms and formulas. These rules are generally context-free (each production has a single symbol on the left side), except that the set of symbols may be allowed to be infinite and there may be many start symbols, for example the variables in the case of terms.

Terms

The set of **terms** is inductively defined by the following rules:

1. **Variables.** Any variable is a term.
2. **Functions.** Any expression $f(t_1,...,t_n)$ of n arguments (where each argument t_i is a term and f is a function symbol of valence n) is a term.

Only expressions which can be obtained by finitely many applications of rules 1 and 2 are terms. For example, no expression involving a predicate symbol is a term.

Formulas

The set of **formulas** (also called **well-formed formulas** or **wff**s) is inductively defined by the following rules:

1. **Predicate symbols.** If P is an n-ary predicate symbol and t_1, ..., t_n are terms then $P(t_1,...,t_n)$ is a formula.
2. **Equality.** If the equality symbol is considered part of logic, and t_1 and t_2 are terms, then $t_1 = t_2$ is a formula.
3. **Negation.** If φ is a formula, then ¬φ is a formula.
4. **Binary connectives.** If φ and ψ are formulas, then (φ ⟶ψ) is a formula. Similar rules apply to other binary logical connectives.
5. **Quantifiers.** If φ is a formula and x is a variable, then $\forall x \varphi$ and $\exists x \varphi$ are formulas.

Only expressions which can be obtained by finitely many applications of rules 1–5 are formulas. The formulas obtained from the first two rules are said to be **atomic formulas**.

For example,

$$\forall x \forall y (P(f(x)) \rightarrow \neg (P(x) \rightarrow Q(f(y), x, z)))$$

is a formula, if f is a unary function symbol, P a unary predicate symbol, and Q a ternary predicate symbol. On the other hand, $\forall x\, x \rightarrow$ is not a formula, although it is a string of symbols from the alphabet.

The role of the parentheses in the definition is to ensure that any formula can only be obtained in one way by following the inductive definition (in other words, there is a unique parse tree for each formula). This property is known as **unique readability** of formulas. There are many conventions for where parentheses are used in formulas. For example, some authors use colons or full stops instead of parentheses, or change the places in which parentheses are inserted. Each author's particular definition must be accompanied by a proof of unique readability.

This definition of a formula does not support defining an if-then-else function ite(c,a,b) where "c" is a condition expressed as a formula, that would return "a" if c is true, and "b" if it is false. This is because both predicates and functions can only accept terms as parameters, but the first parameter is a formula. Some languages built on first-order logic, such as SMT-LIB 2.0, add this.

Notational conventions

For convenience, conventions have been developed about the precedence of the logical operators, to avoid the need to write parentheses in some cases. These rules are similar to the order of operations in arithmetic. A common convention is:

- ¬is evaluated first
- ∧and ∨are evaluated next
- Quantifiers are evaluated next
- →is evaluated last.

Moreover, extra punctuation not required by the definition may be inserted to make formulas easier to read. Thus the formula

$$(\neg \forall x P(x) \rightarrow \exists x \neg P(x))$$

might be written as

$$(\neg [\forall x P(x)]) \rightarrow \exists x [\neg P(x)].$$

In some fields, it is common to use infix notation for binary relations and functions, instead of the prefix notation defined above. For example, in arithmetic, one typically writes "2 + 2 = 4" instead of "=(+(2,2),4)". It is common to regard formulas in infix notation as abbreviations for the corresponding formulas in prefix notation.

The definitions above use infix notation for binary connectives such as →. A less common convention is Polish notation, in which one writes →, ∧, and so on in front of their arguments rather than between them. This convention allows all punctuation symbols to be discarded. Polish notation is compact and elegant, but rarely used in practice because it is hard for humans to read it. In Polish notation, the formula

$$\forall x \forall y (P(f(x)) \rightarrow \neg (P(x) \rightarrow Q(f(y), x, z)))$$

becomes " ∀x∀y→Pfx¬→ PxQfyxz ".

Example

In mathematics the language of ordered abelian groups has one constant symbol 0, one unary function symbol −, one binary function symbol +, and one binary relation symbol ≤. Then:

- The expressions +(x, y) and +(x, +(y, −(z))) are **terms**. These are usually written as x + y and x + y − z.
- The expressions +(x, y) = 0 and ≤(+(x, +(y, −(z))), +(x, y)) are **atomic formulas**.

  These are usually written as x + y = 0 and x + y − z ≤ x + y.

- The expression $(\forall x \forall y \leq (+(x, y), z) \rightarrow \forall x\, \forall y + (x, y) = 0)$ is a **formula**, which is usually written as $\forall x \forall y (x + y \leq z) \rightarrow \forall x \forall y (x + y = 0)$.

Free and bound variables

In a formula, a variable may occur **free** or **bound**. Intuitively, a variable is free in a formula if it is not quantified: in $\forall y\, P(x, y)$, variable x is free while y is bound. The free and bound variables of a formula are defined inductively as follows.

1. **Atomic formulas.** If φ is an atomic formula then x is free in φ if and only if x occurs in φ. Moreover, there are no bound variables in any atomic formula.
2. **Negation.** x is free in ¬φ if and only if x is free in φ. x is bound in ¬φ if and only if x is bound in φ.
3. **Binary connectives.** x is free in (φ ⟶ψ) if and only if x is free in either φ or ψ. x is bound in (φ ⟶ψ) if and only if x is bound in either φ or ψ. The same rule applies to any other binary connective in place of ⟶.
4. **Quantifiers.** x is free in ∀y φ if and only if x is free in φ and x is a different symbol from y. Also, x is bound in ∀y φ if and only if x is y or x is bound in φ. The same rule holds with ∃in place of ∀.

For example, in ∀x ∀y (P(x)⟶ Q(x,f(x),z)), x and y are bound variables, z is a free variable, and w is neither because it does not occur in the formula.

Freeness and boundness can be also specialized to specific occurrences of variables in a formula. For example, in $P(x) \rightarrow \forall x\, Q(x)$, the first occurrence of x is free while the second is bound. In other words, the x in P(x) is free while the x in $\forall x\, Q(x)$ is bound.

A formula in first-order logic with no free variables is called a **first-order sentence**. These are the formulas that will have well-defined truth values under an interpretation. For example, whether a formula such as Phil(x) is true must depend on what x represents. But the sentence $\exists x\, \mathrm{Phil}(x)$ will be either true or false in a given interpretation.

Semantics

An interpretation of a first-order language assigns a denotation to all non-logical constants in that language. It also determines a domain of discourse that specifies the range of the quantifiers. The result is that each term is assigned an object that it

represents, and each sentence is assigned a truth value. In this way, an interpretation provides semantic meaning to the terms and formulas of the language. The study of the interpretations of formal languages is called formal semantics.

The domain of discourse D is a nonempty set of "objects" of some kind. Intuitively, a first-order formula is a statement about these objects; for example, $\exists x P(x)$ states the existence of an object x such that the predicate P is true where referred to it. The domain of discourse is the set of considered objects. For example, one can take D to be the set of integer numbers.

The interpretation of a function symbol is a function. For example, if the domain of discourse consists of integers, a function symbol f of arity 2 can be interpreted as the function that gives the sum of its arguments. In other words, the symbol f is associated with the function I(f) which, in this interpretation, is addition.

The interpretation of a constant symbol is a function from the one-element set D^0 to D, which can be simply identified with an object in D. For example, an interpretation may assign the value I(c) = 10 to the constant symbol c.

The interpretation of an n-ary predicate symbol is a set of n-tuples of elements of the domain of discourse. This means that, given an interpretation, a predicate symbol, and n elements of the domain of discourse, one can tell whether the predicate is true of those elements according to the given interpretation. For example, an interpretation I(P) of a binary predicate symbol P may be the set of pairs of integers such that the first one is less than the second. According to this interpretation, the predicate P would be true if its first argument is less than the second.

First-order structures

The most common way of specifying an interpretation (especially in mathematics) is to specify a **structure** (also called a **model**; see below). The structure consists of a nonempty set D that forms the domain of discourse and an interpretation I of the non-logical terms of the signature. This interpretation is itself a function:

- Each function symbol f of arity n is assigned a function I(f) from D^n to D. In particular, each constant symbol of the signature is assigned an individual in the domain of discourse.
- Each predicate symbol P of arity n is assigned a relation I(P) over D^n or, equivalently, a function from D^n to {true,false}. Thus each predicate symbol is interpreted by a Boolean-valued function on D.

Evaluation of truth values

A formula evaluates to true or false given an interpretation, and a **variable assignment** μ that associates an element of the domain of discourse with each variable. The reason that a variable assignment is required is to give meanings to formulas with free variables,

such as y = x. The truth value of this formula changes depending on whether x and y denote the same individual.

First, the variable assignment μ can be extended to all terms of the language, with the result that each term maps to a single element of the domain of discourse. The following rules are used to make this assignment:

1. **Variables.** Each variable x evaluates to μ(x)
2. **Functions.** Given terms $t_1, \ldots, t_n$ that have been evaluated to elements $d_1, \ldots, d_n$ of the domain of discourse, and a n-ary function symbol f, the term $f(t_1, \ldots, t_n)$ evaluates to $(I(f))(d_1, \ldots, d_n)$.

Next, each formula is assigned a truth value. The inductive definition used to make this assignment is called the T-schema.

1. **Atomic formulas (1).** A formula $P(t_1, \ldots, t_n)$ is associated the value true or false depending on whether $\langle v_1, \ldots, v_n \rangle \in I(P)$, where $v_1, \ldots, v_n$ are the evaluation of the terms $t_1, \ldots, t_n$ and I(P) is the interpretation of P, which by assumption is a subset of D^n.
2. **Atomic formulas (2).** A formula $t_1 = t_2$ is assigned true if t_1 and t_2 evaluate to the same object of the domain of discourse.
3. **Logical connectives.** A formula in the form $\neg\phi$, $\phi \rightarrow \psi$, etc. is evaluated according to the truth table for the connective in question, as in propositional logic.
4. **Existential quantifiers.** A formula $\exists x \phi(x)$ is true according to M and μ if there exists an evaluation μ' of the variables that only differs from μ regarding the evaluation of x and such that φ is true according to the interpretation M and the variable assignment μ'. This formal definition captures the idea that $\exists x \phi(x)$ is true if and only if there is a way to choose a value for x such that φ(x) is satisfied.
5. **Universal quantifiers.** A formula $\forall x \phi(x)$ is true according to M and μ if φ(x) is true for every pair composed by the interpretation M and some variable assignment μ' that differs from μ only on the value of x. This captures the idea that $\forall x \phi(x)$ is true if every possible choice of a value for x causes φ(x) to be true.

If a formula does not contain free variables, and so is a sentence, then the initial variable assignment does not affect its truth value. In other words, a sentence is true according to M and μ if and only if is true according to M and any other variable assignment μ'.

There is a second common approach to defining truth values that does not rely on variable assignment functions. Instead, given an interpretation M, one first adds to the signature a collection of constant symbols, one for each element of the domain of discourse in M; say that for each d in the domain the constant symbol c_d is fixed. The

interpretation is extended so that each new constant symbol is assigned to its corresponding element of the domain. One now defines truth for quantified formulas syntactically, as follows:

1. **Existential quantifiers (alternate).** A formula $\exists x \phi(x)$ is true according to M if there is some d in the domain of discourse such that $\varphi(c_d)$ holds. Here $\varphi(c_d)$ is the result of substituting c_d for every free occurrence of x in φ.
2. **Universal quantifiers (alternate).** A formula $\forall x \phi(x)$ is true according to M if, for every d in the domain of discourse, $\varphi(c_d)$ is true according to M.

This alternate approach gives exactly the same truth values to all sentences as the approach via variable assignments.

Validity, satisfiability, and logical consequence

If a sentence φ evaluates to True under a given interpretation M, one says that M **satisfies** φ; this is denoted $M \vDash \phi$. A sentence is **satisfiable** if there is some interpretation under which it is true.

Satisfiability of formulas with free variables is more complicated, because an interpretation on its own does not determine the truth value of such a formula. The most common convention is that a formula with free variables is said to be satisfied by an interpretation if the formula remains true regardless which individuals from the domain of discourse are assigned to its free variables. This has the same effect as saying that a formula is satisfied if and only if its universal closure is satisfied.

A formula is **logically valid** (or simply **valid**) if it is true in every interpretation. These formulas play a role similar to tautologies in propositional logic.

A formula φ is a **logical consequence** of a formula ψ if every interpretation that makes ψ true also makes φ true. In this case one says that φ is logically implied by ψ.

Algebraizations

An alternate approach to the semantics of first-order logic proceeds via abstract algebra. This approach generalizes the Lindenbaum–Tarski algebras of propositional logic. There are three ways of eliminating quantified variables from first-order logic, that do not involve replacing quantifiers with other variable binding term operators:

- Cylindric algebra, by Alfred Tarski and his coworkers;
- Polyadic algebra, by Paul Halmos;
- Predicate functor logic, mainly due to Willard Quine.

These algebras are all lattices that properly extend the two-element Boolean algebra.

Tarski and Givant (1987) showed that the fragment of first-order logic that has no atomic sentence lying in the scope of more than three quantifiers, has the same expressive power as relation algebra. This fragment is of great interest because it suffices for Peano arithmetic and most axiomatic set theory, including the canonical ZFC. They also prove that first-order logic with a primitive ordered pair is equivalent to a relation algebra with two ordered pair projection functions.

First-order theories, models, and elementary classes

A **first-order theory** consists of a set of axioms in a particular first-order signature. The set of axioms is often finite or recursively enumerable, in which case the theory is called **effective**. Some authors require theories to also include all logical consequences of the axioms.

A first-order structure that satisfies all sentences in a given theory is said to be a **model** of the theory. An **elementary class** is the set of all structures satisfying a particular theory. These classes are a main subject of study in model theory.

Many theories have an intended interpretation, a certain model that is kept in mind when studying the theory. For example, the intended intended interpretation of Peano arithmetic consists of the usual natural numbers with their usual operations. However, the Löwenheim–Skolem theorem shows that most first-order theories will also have other, nonstandard models.

A theory is **consistent** if it is not possible to prove a contradiction from the axioms of the theory. A theory is **complete** if, for every formula in its signature, either that formula or its negation is a logical consequence of the axioms of the theory. Gödel's incompleteness theorem shows that effective first-order theories that include a sufficient portion of the theory of the natural numbers can never be both consistent and complete.

Empty domains

The definition above requires that the domain of discourse of any interpretation must be a nonempty set. There are settings, such as inclusive logic, where empty domains are permitted. Moreover, if a class of algebraic structures includes an empty structure (for example, there is an empty poset), that class can only be an elementary class in first-order logic if empty domains are permitted or the empty structure is removed from the class.

There are several difficulties with empty domains, however:

- Many common rules of inference are only valid when the domain of discourse is required to be nonempty. One example is the rule stating that $\phi \vee \exists x \psi$ implies $\exists x(\phi \vee \psi)$ when x is not a free variable in φ. This rule, which is used to put formulas into prenex normal form, is sound in nonempty domains, but unsound if the empty domain is permitted.

- The definition of truth in an interpretation that uses a variable assignment function cannot work with empty domains, because there are no variable assignment functions whose range is empty. (Similarly, one cannot assign interpretations to constant symbols.) This truth definition requires that one must select a variable assignment function (μ above) before truth values for even atomic formulas can be defined. Then the truth value of a sentence is defined to be its truth value under any variable assignment, and it is proved that this truth value does not depend on which assignment is chosen. This technique does not work if there are no assignment functions at all; it must be changed to accommodate empty domains.

Thus, when the empty domain is permitted, it must often be treated as a special case. Most authors, however, simply exclude the empty domain by definition.

Deductive systems

A **deductive system** is used to demonstrate, on a purely syntactic basis, that one formula is a logical consequence of another formula. There are many such systems for first-order logic, including Hilbert-style deductive systems, natural deduction, the sequent calculus, the tableaux method, and resolution. These share the common property that a deduction is a finite syntactic object; the format of this object, and the way it is constructed, vary widely. These finite deductions themselves are often called **derivations** in proof theory. They are also often called proofs, but are completely formalized unlike natural-language mathematical proofs.

A deductive system is **sound** if any formula that can be derived in the system is logically valid. Conversely, a deductive system is **complete** if every logically valid formula is derivable. All of the systems discussed here are both sound and complete. They also share the property that it is possible to effectively verify that a purportedly valid deduction is actually a deduction; such deduction systems are called **effective**.

A key property of deductive systems is that they are purely syntactic, so that derivations can be verified without considering any interpretation. Thus a sound argument is correct in every possible interpretation of the language, regardless whether that interpretation is about mathematics, economics, or some other area.

In general, logical consequence in first-order logic is only semidecidable: if a sentence A logically implies a sentence B then this can be discovered (for example, by searching for a proof until one is found, using some effective, sound, complete proof system). However, if A does not logically imply B, this does not mean that A logically implies the negation of B. There is no effective procedure that, given formulas A and B, always correctly decides whether A logically implies B.

Rules of inference

A **rule of inference** states that, given a particular formula (or set of formulas) with a certain property as a hypothesis, another specific formula (or set of formulas) can be

derived as a conclusion. The rule is sound (or truth-preserving) if it preserves validity in the sense that whenever any interpretation satisfies the hypothesis, that interpretation also satisfies the conclusion.

For example, one common rule of inference is the **rule of substitution**. If t is a term and φ is a formula possibly containing the variable x, then φ[t/x] (often denoted φ[x/t]) is the result of replacing all free instances of x by t in φ. The substitution rule states that for any φ and any term t, one can conclude φ[t/x] from φ provided that no free variable of t becomes bound during the substitution process. (If some free variable of t becomes bound, then to substitute t for x it is first necessary to change the bound variables of φ to differ from the free variables of t.)

To see why the restriction on bound variables is necessary, consider the logically valid formula φ given by $\exists x(x = y)$, in the signature of (0,1,+,×,=) of arithmetic. If t is the term "x + 1", the formula φ[t/y] is $\exists x(x = x + 1)$, which will be false in many interpretations. The problem is that the free variable x of t became bound during the substitution. The intended replacement can be obtained by renaming the bound variable x of φ to something else, say z, so that the formula after substitution is $\exists z(z = x + 1)$, which is again logically valid.

The substitution rule demonstrates several common aspects of rules of inference. It is entirely syntactical; one can tell whether it was correctly applied without appeal to any interpretation. It has (syntactically-defined) limitations on when it can be applied, which must be respected to preserve the correctness of derivations. Moreover, as is often the case, these limitations are necessary because of interactions between free and bound variables that occur during syntactic manipulations of the formulas involved in the inference rule.

Hilbert-style systems and natural deduction

A deduction in a Hilbert-style deductive system is a list of formulas, each of which is a **logical axiom**, a hypothesis that has been assumed for the derivation at hand, or follows from previous formulas via a rule of inference. The logical axioms consist of several axiom schemes of logically valid formulas; these encompass a significant amount of propositional logic. The rules of inference enable the manipulation of quantifiers. Typical Hilbert-style systems have a small number of rules of inference, along with several infinite schemes of logical axioms. It is common to have only modus ponens and universal generalization as rules of inference.

Natural deduction systems resemble Hilbert-style systems in that a deduction is a finite list of formulas. However, natural deduction systems have no logical axioms; they compensate by adding additional rules of inference that can be used to manipulate the logical connectives in formulas in the proof.

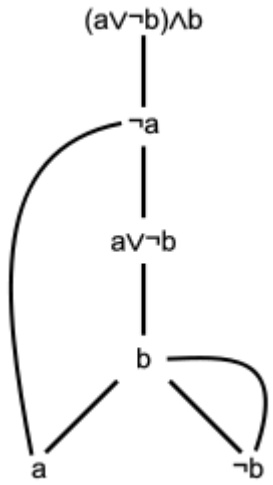
Sequent calculus

The sequent calculus was developed to study the properties of natural deduction systems. Instead of working with one formula at a time, it uses **sequents**, which are expressions of the form

$$A_1, \ldots, A_n \vdash B_1, \ldots, B_k,$$

where A_1, ..., A_n, B_1, ..., B_k are formulas and the turnstile symbol $\vdash$ is used as punctuation to separate the two halves. Intuitively, a sequent expresses the idea that $(A_1 \wedge \cdots \wedge A_n)$ implies $(B_1 \vee \cdots \vee B_k)$.

Tableaux method



A tableau proof for the propositional formula ((a ∨ ~b) & b) → a

Unlike the methods just described, the derivations in the tableaux method are not lists of formulas. Instead, a derivation is a tree of formulas. To show that a formula A is provable, the tableaux method attempts to demonstrate that the negation of A is unsatisfiable. The tree of the derivation has $\neg A$ at its root; the tree branches in a way that reflects the structure of the formula. For example, to show that $C \vee D$ is unsatisfiable requires showing that C and D are each unsatisfiable; the corresponds to a branching point in the tree with parent $C \vee D$ and children C and D.

Resolution

The resolution rule is a single rule of inference that, together with unification, is sound and complete for first-order logic. As with the tableaux method, a formula is proved by showing that the negation of the formula is unsatisfiable. Resolution is commonly used in automated theorem proving.

The resolution method works only with formulas that are disjunctions of atomic formulas; arbitrary formulas must first be converted to this form through Skolemization. The resolution rule states that from the hypotheses $A_1 \vee \cdots \vee A_k \vee C$ and $B_1 \vee \cdots \vee B_l \vee \neg C$, the conclusion $A_1 \vee \cdots \vee A_k \vee B_1 \vee \cdots \vee B_l$ can be obtained.

Provable identities

The following sentences can be called "identities" because the main connective in each is the biconditional.

$$\neg\forall x\, P(x) \Leftrightarrow \exists x\, \neg P(x)$$
$$\neg\exists x\, P(x) \Leftrightarrow \forall x\, \neg P(x)$$
$$\forall x\, \forall y\, P(x,y) \Leftrightarrow \forall y\, \forall x\, P(x,y)$$
$$\exists x\, \exists y\, P(x,y) \Leftrightarrow \exists y\, \exists x\, P(x,y)$$
$$\forall x\, P(x) \wedge \forall x\, Q(x) \Leftrightarrow \forall x\, (P(x) \wedge Q(x))$$
$$\exists x\, P(x) \vee \exists x\, Q(x) \Leftrightarrow \exists x\, (P(x) \vee Q(x))$$
$$P \wedge \exists x\, Q(x) \Leftrightarrow \exists x\, (P \wedge Q(x)) \quad \text{(where x must not occur free in P)}$$
$$P \vee \forall x\, Q(x) \Leftrightarrow \forall x\, (P \vee Q(x)) \quad \text{(where x must not occur free in P)}$$

Equality and its axioms

There are several different conventions for using equality (or identity) in first-order logic. The most common convention, known as **first-order logic with equality**, includes the equality symbol as a primitive logical symbol which is always interpreted as the real equality relation between members of the domain of discourse, such that the "two" given members are the same member. This approach also adds certain axioms about equality to the deductive system employed. These equality axioms are:

1. **Reflexivity**. For each variable x, x = x.
2. **Substitution for functions.** For all variables x and y, and any function symbol f,

   x = y → f(...,x,...) = f(...,y,...).

3. **Substitution for formulas**. For any variables x and y and any formula φ(x), if φ' is obtained by replacing any number of free occurrences of x in φ with y, such that these remain free occurrences of y, then

   x = y → (φ → φ').

These are axiom schemes, each of which specifies an infinite set of axioms. The third scheme is known as **Leibniz's law**, "the principle of substitutivity", "the indiscernibility of identicals", or "the replacement property". The second scheme, involving the function symbol f, is (equivalent to) a special case of the third scheme, using the formula

$x = y \rightarrow (f(...,x,...) = z \rightarrow f(...,y,...) = z)$.

Many other properties of equality are consequences of the axioms above, for example:

1. **Symmetry.** If x = y then y = x.
2. **Transitivity.** If x = y and y = z then x = z.

First-order logic without equality

An alternate approach considers the equality relation to be a non-logical symbol. This convention is known as **first-order logic without equality**. If an equality relation is included in the signature, the axioms of equality must now be added to the theories under consideration, if desired, instead of being considered rules of logic. The main difference between this method and first-order logic with equality is that an interpretation may now interpret two distinct individuals as "equal" (although, by Leibniz's law, these will satisfy exactly the same formulas under any interpretation). That is, the equality relation may now be interpreted by an arbitrary equivalence relation on the domain of discourse that is congruent with respect to the functions and relations of the interpretation.

When this second convention is followed, the term **normal model** is used to refer to an interpretation where no distinct individuals a and b satisfy a = b. In first-order logic with equality, only normal models are considered, and so there is no term for a model other than a normal model. When first-order logic without equality is studied, it is necessary to amend the statements of results such as the Löwenheim–Skolem theorem so that only normal models are considered.

First-order logic without equality is often employed in the context of second-order arithmetic and other higher-order theories of arithmetic, where the equality relation between sets of natural numbers is usually omitted.

Defining equality within a theory

If a theory has a binary formula A(x,y) which satisfies reflexivity and Leibniz's law, the theory is said to have equality, or to be a theory with equality. The theory may not have all instances of the above schemes as axioms, but rather as derivable theorems. For example, in theories with no function symbols and a finite number of relations, it is possible to define equality in terms of the relations, by defining the two terms s and t to be equal if any relation is unchanged by changing s to t in any argument.

Some theories allow other ad hoc definitions of equality:

- In the theory of partial orders with one relation symbol $\leq$, one could define s = t to be an abbreviation for $s \leq t \wedge t \leq s$.
- In set theory with one relation $\in$, one may define s = t to be an abbreviation for $\forall x\ (s \in x \leftrightarrow t \in x) \wedge \forall x\ (x \in s \leftrightarrow x \in t)$. This definition of equality then automatically satisfies the axioms for equality. In this case, one should replace the

usual axiom of extensionality, $\forall x \forall y [\forall z (z \in x \Leftrightarrow z \in y) \Rightarrow x = y]$, by $\forall x \forall y [\forall z (z \in x \Leftrightarrow z \in y) \Rightarrow \forall z (x \in z \Leftrightarrow y \in z)]$, i.e. if x and y have the same elements, then they belong to the same sets.

Metalogical properties

One motivation for the use of first-order logic, rather than higher-order logic, is that first-order logic has many metalogical properties that stronger logics do not have. These results concern general properties of first-order logic itself, rather than properties of individual theories. They provide fundamental tools for the construction of models of first-order theories.

Completeness and undecidability

Gödel's completeness theorem, proved by Kurt Gödel in 1929, establishes that there are sound, complete, effective deductive systems for first-order logic, and thus the first-order logical consequence relation is captured by finite provability. Naively, the statement that a formula φ logically implies a formula ψ depends on every model of φ; these models will in general be of arbitrarily large cardinality, and so logical consequence cannot be effectively verified by checking every model. However, it is possible to enumerate all finite derivations and search for a derivation of ψ from φ. If ψ is logically implied by φ, such a derivation will eventually be found. Thus first-order logical consequence is semidecidable: it is possible to make an effective enumeration of all pairs of sentences (φ,ψ) such that ψ is a logical consequence of φ.

Unlike propositional logic, first-order logic is undecidable (although semidecidable), provided that the language has at least one predicate of arity at least 2 (other than equality). This means that there is no decision procedure that determines whether arbitrary formulas are logically valid. This result was established independently by Alonzo Church and Alan Turing in 1936 and 1937, respectively, giving a negative answer to the Entscheidungsproblem posed by David Hilbert in 1928. Their proofs demonstrate a connection between the unsolvability of the decision problem for first-order logic and the unsolvability of the halting problem.

There are systems weaker than full first-order logic for which the logical consequence relation is decidable. These include propositional logic and monadic predicate logic, which is first-order logic restricted to unary predicate symbols and no function symbols. The Bernays–Schönfinkel class of first-order formulas is also decidable.

The Löwenheim–Skolem theorem

The Löwenheim–Skolem theorem shows that if a first-order theory of cardinality λ has any infinite model then it has models of every infinite cardinality greater than or equal to λ. One of the earliest results in model theory, it implies that it is not possible to characterize countability or uncountability in a first-order language. That is, there is no

first-order formula φ(x) such that an arbitrary structure M satisfies φ if and only if the domain of discourse of M is countable (or, in the second case, uncountable).

The Löwenheim–Skolem theorem implies that infinite structures cannot be categorically axiomatized in first-order logic. For example, there is no first-order theory whose only model is the real line: any first-order theory with an infinite model also has a model of cardinality larger than the continuum. Since the real line is infinite, any theory satisfied by the real line is also satisfied by some nonstandard models. When the Löwenheim–Skolem theorem is applied to first-order set theories, the nonintuitive consequences are known as Skolem's paradox.

The compactness theorem

The compactness theorem states that a set of first-order sentences has a model if and only if every finite subset of it has a model. This implies that if a formula is a logical consequence of an infinite set of first-order axioms, then it is a logical consequence of some finite number of those axioms. This theorem was proved first by Kurt Gödel as a consequence of the completeness theorem, but many additional proofs have been obtained over time. It is a central tool in model theory, providing a fundamental method for constructing models.

The compactness theorem has a limiting effect on which collections of first-order structures are elementary classes. For example, the compactness theorem implies that any theory that has arbitrarily large finite models has an infinite model. Thus the class of all finite graphs is not an elementary class (the same holds for many other algebraic structures).

There are also more subtle limitations of first-order logic that are implied by the compactness theorem. For example, in computer science, many situations can be modeled as a directed graph of states (nodes) and connections (directed edges). Validating such a system may require showing that no "bad" state can be reached from any "good" state. Thus one seeks to determine if the good and bad states are in different connected components of the graph. However, the compactness theorem can be used to show that connected graphs are not an elementary class in first-order logic, and there is no formula φ(x,y) of first-order logic, in the signature of graphs, that expresses the idea that there is a path from x to y. Connectedness can be expressed in second-order logic, however.

Lindström's theorem

Per Lindström showed that the metalogical properties just discussed actually characterize first-order logic in the sense that no stronger logic has the properties. Lindström defined a class of abstract logical systems, so that it makes sense to say that one system is stronger than another. He established two theorems for systems of this type:

- A logical system satisfying Lindström's definition that contains first-order logic and satisfies both the Löwenheim–Skolem theorem and the compactness theorem must be equivalent to first-order logic.
- A logical system satisfying Lindström's definition that has a semidecidable logical consequence relation and satisfies the Löwenheim–Skolem theorem must be equivalent to first-order logic.

Restrictions, extensions and variations

There are many variations of first-order logic. Some of these are inessential in the sense that they merely change notation without affecting the semantics. Others change the expressive power more significantly, by extending the semantics through additional quantifiers or other new logical symbols. For example, infinitary logics permit formulas of infinite size, and modal logics add symbols for possibility and necessity.

Restricted languages

First-order logic can be studied in languages with fewer logical symbols than were described above.

- Because $\exists x \phi(x)$ can be expressed as $\neg \forall x \neg \phi(x)$, and $\forall x \phi(x)$ can be expressed as $\neg \exists x \neg \phi(x)$, either of the two quantifiers $\exists$ and $\forall$ can be dropped.
- Since $\phi \vee \psi$ can be expressed as $\neg(\neg \phi \wedge \neg \psi)$ and $\phi \wedge \psi$ can be expressed as $\neg(\neg \phi \vee \neg \psi)$, either $\vee$ or $\wedge$ can be dropped. In other words, it is sufficient to have $\neg$ and $\vee$, or $\neg$ and $\wedge$, as the only logical connectives.
- Similarly, it is sufficient to have only $\neg$ and $\rightarrow$ as logical connectives, or to have only the Sheffer stroke (NAND) or the Peirce arrow (NOR) operator.
- It is possible to entirely avoid function symbols and constant symbols, rewriting them via predicate symbols in a appropriate way. For example, instead of using a constant symbol 0 one may use a predicate $0(x)$ (interpreted as $x = 0$), and replace every predicate such as $P(0, y)$ with $\forall x \ (0(x) \rightarrow P(x, y))$. A function such as $f(x_1,x_2,...,x_n)$ will similarly be replaced by a predicate $F(x_1,x_2,...,x_n,y)$ interpreted as $y = f(x_1,x_2,...,x_n)$. This change requires adding additional axioms to the theory at hand, so that interpretations of the predicate symbols used have the correct semantics.

Restrictions such as these are useful as a technique to reduce the number of inference rules or axiom schemes in deductive systems, which leads to shorter proofs of metalogical results. The cost of the restrictions is that it becomes more difficult to express natural-language statements in the formal system at hand, because the logical connectives used in the natural language statements must be replaced by their (longer) definitions in terms of the restricted collection of logical connectives. Similarly, derivations in the limited systems may be longer than derivations in systems that include additional

connectives. There is thus a trade-off between the ease of working within the formal system and the ease of proving results about the formal system.

It is also possible to restrict the arities of function symbols and predicate symbols, in sufficiently expressive theories. One can in principle dispense entirely with functions of arity greater than 2 and predicates of arity greater than 1 in theories that include a pairing function. This is a function of arity 2 that takes pairs of elements of the domain and returns an ordered pair containing them. It is also sufficient to have two predicate symbols of arity 2 that define projection functions from an ordered pair to its components. In either case it is necessary that the natural axioms for a pairing function and its projections are satisfied.

Many-sorted logic

Ordinary first-order interpretations have a single domain of discourse over which all quantifiers range. **Many-sorted first-order logic** allows variables to have different **sorts**, which have different domains. This is also called **typed first-order logic**, and the sorts called **types** (as in data type), but it is not the same as first-order type theory. Many-sorted first-order logic is often used in the study of second-order arithmetic.

When there are only finitely many sorts in a theory, many-sorted first-order logic can be reduced to single-sorted first-order logic. One introduces into the single-sorted theory a unary predicate symbol for each sort in the many-sorted theory, and adds an axiom saying that these unary predicates partition the domain of discourse. For example, if there are two sorts, one adds predicate symbols $P_1(x)$ and $P_2(x)$ and the axiom

$$\forall x(P_1(x) \vee P_2(x)) \wedge \neg\exists x(P_1(x) \wedge P_2(x))$$

Then the elements satisfying P_1 are thought of as elements of the first sort, and elements satisfying P_2 as elements of the second sort. One can quantify over each sort by using the corresponding predicate symbol to limit the range of quantification. For example, to say there is an element of the first sort satisfying formula $\varphi(x)$, one writes

$$\exists x(P_1(x) \wedge \phi(x))$$

Additional quantifiers

Additional quantifiers can be added to first-order logic.

- Sometimes it is useful to say that "P(x) holds for exactly one x", which can be expressed as $\exists!$x P(x). This notation, called uniqueness quantification, may be taken to abbreviate a formula such as $\exists$x (P(x) $\wedge\forall$y (P(y) $\rightarrow$(x = y))).
- **First-order logic with extra quantifiers** has new quantifiers Qx,..., with meanings such as "there are many x such that ...". Also see branching quantifiers and the plural quantifiers of George Boolos and others.

- **Bounded quantifiers** are often used in the study of set theory or arithmetic.

Infinitary logics

Infinitary logic allows infinitely long sentences. For example, one may allow a conjunction or disjunction of infinitely many formulas, or quantification over infinitely many variables. Infinitely long sentences arise in areas of mathematics including topology and model theory.

Infinitary logic generalizes first-order logic to allow formulas of infinite length. The most common way in which formulas can become infinite is through infinite conjunctions and disjunctions. However, it is also possible to admit generalized signatures in which function and relation symbols are allowed to have infinite arities, or in which quantifiers can bind infinitely many variables. Because an infinite formula cannot be represented by a finite string, it is necessary to choose some other representation of formulas; the usual representation in this context is a tree. Thus formulas are, essentially, identified with their parse trees, rather than with the strings being parsed.

The most commonly studied infinitary logics are denoted $L_{\alpha\beta}$, where α and β are each either cardinal numbers or the symbol ∞. In this notation, ordinary first-order logic is $L_{\omega\omega}$. In the logic $L_{\infty\omega}$, arbitrary conjunctions or disjunctions are allowed when building formulas, and there is an unlimited supply of variables. More generally, the logic that permits conjunctions or disjunctions with less than κ constituents is known as $L_{\kappa\omega}$. For example, $L_{\omega 1\omega}$ permits countable conjunctions and disjunctions.

The set of free variables in a formula of $L_{\kappa\omega}$ can have any cardinality strictly less than κ, yet only finitely many of them can be in the scope of any quantifier when a formula appears as a subformula of another. In other infinitary logics, a subformula may be in the scope of infinitely many quantifiers. For example, in $L_{\kappa\infty}$, a single universal or existential quantifier may bind arbitrarily many variables simultaneously. Similarly, the logic $L_{\kappa\lambda}$ permits simultaneous quantification over fewer than λ variables, as well as conjunctions and disjunctions of size less than κ.

Non-classical and modal logics

- **Intuitionistic first-order logic** uses intuitionistic rather than classical propositional calculus; for example, $\neg\neg\varphi$ need not be equivalent to φ.

- First-order **modal logic** allows one to describe other possible worlds as well as this contingently true world which we inhabit. In some versions, the set of possible worlds varies depending on which possible world one inhabits. Modal logic has extra modal operators with meanings which can be characterized informally as, for example "it is necessary that φ" (true in all possible worlds) and "it is possible that φ" (true in some possible world). With standard first-order logic we have a single domain and each predicate is assigned one extension. With first-order modal logic we have a domain function that assigns each possible

world its own domain, so that each predicate gets an extension only relative to these possible worlds. This allows us to model cases where, for example, Alex is a Philosopher, but may have been a Mathematician, and may not have existed at all. In the first possible world P(a) is true, in the second P(a) is false, and in the third possible world there is no a in the domain at all.

- **first-order fuzzy logics** are first-order extensions of propositional fuzzy logics rather than classical propositional calculus.

Higher-order logics

The characteristic feature of first-order logic is that individuals can be quantified, but not predicates. Thus

$$\exists a(\mathrm{Phil}(a))$$

is a legal first-order formula, but

$$\exists \mathrm{Phil}(\mathrm{Phil}(a))$$

is not. Second-order logic extends first-order logic by adding the latter type of quantification. Other higher-order logics allow quantification over even higher types than second-order logic permits. These higher types include relations between relations, functions from relations to relations between relations, and other higher-type objects. Thus the "first" in first-order logic describes the type of objects that can be quantified.

Unlike first-order logic, for which only one semantics is studied, there are several possible semantics for second-order logic. The most commonly employed semantics for second-order and higher-order logic, known as **full semantics**, is much stronger than the semantics for first-order logic. In particular, the (semantic) logical consequence relation for second-order and higher-order logic is not semidecidable; there is no effective deduction system for second-order logic that is sound and complete under full semantics.

Second-order logic with full semantics is more expressive than first-order logic. For example, it is possible to create axiom systems in second-order logic that uniquely characterize the natural numbers and the real line. The cost of this expressiveness is that second-order and higher-order logics have fewer attractive metalogical properties than first-order logic. The Löwenheim–Skolem theorem and compactness theorem become false when generalized to stronger logics.

Automated theorem proving and formal methods

Automated theorem proving refers to the development of computer programs that search and find derivations (formal proofs) of mathematical theorems. Finding derivations is a difficult task because the search space can be very large; an exhaustive search of every

possible derivation is theoretically possible but computationally infeasible for many systems of interest in mathematics. Thus complicated heuristic functions are developed to attempt to find a derivation in less time than a blind search.

The related area of automated proof verification uses computer programs to check that human-created proofs are correct. Unlike complicated automated theorem provers, verification systems may be small enough that their correctness can be checked both by hand and through automated software verification. This validation of the proof verifier is needed to give confidence that any derivation labeled as "correct" is actually correct.

Some proof verifiers, such as Metamath, insist on having a complete derivation as input. Others, such as Mizar and Isabelle, take a well-formatted proof sketch (which may still be very long and detailed) and fill in the missing pieces by doing simple proof searches or applying known decision procedures: the resulting derivation is then verified by a small, core "kernel". Many such systems are primarily intended for interactive use by human mathematicians: these are known as proof assistants. They may also use formal logics that are stronger than first-order logic, such as type theory. Because a full derivation of any nontrivial result in a first-order deductive system will be extremely long for a human to write, results are often formalized as a series of lemmas, for which derivations can be constructed separately.

Automated theorem provers are also used to implement formal methods in computer science. In this setting, theorem provers are used to verify the correctness of programs and of hardware such as processors with respect to a formal specification. Because such analysis is time-consuming and thus expensive, it is usually reserved for projects in which a malfunction would have grave human or financial consequences.

Chapter- 5

Computability Theory

Computability theory, also called **recursion theory**, is a branch of mathematical logic that originated in the 1930s with the study of computable functions and Turing degrees. The field has grown to include the study of generalized computability and definability. In these areas, recursion theory overlaps with proof theory and effective descriptive set theory.

The basic questions addressed by recursion theory are "What does it mean for a function from the natural numbers to themselves to be computable?" and "Can noncomputable functions be classified into a hierarchy based on their level of noncomputability?". The answers to these questions have led to a rich theory that is still being actively researched.

The field is also closely related to computer science. Recursion theorists in mathematical logic often study the theory of relative computability, reducibility notions and degree structures described here. This contrasts with the theory of subrecursive hierarchies, formal methods and formal languages that is common in the study of computability theory in computer science. There is considerable overlap in knowledge and methods between these two research communities, however, and no firm line can be drawn between them.

Computable and uncomputable sets

Recursion theory originated with work of Kurt Gödel, Alonzo Church, Alan Turing, Stephen Kleene and Emil Post in the 1930s.

The fundamental results the researchers obtained established Turing computability as the correct formalization of the informal idea of effective calculation. These results led Stephen Kleene (1952) to coin the two names "Church's thesis" (Kleene 1952:300) and "Turing's Thesis" (p. 376). Nowadays these are often considered as a single hypothesis, the **Church–Turing thesis**, which states that any function that is computable by an algorithm is a computable function. Although initially skeptical, by 1946 Gödel argued in favor of this thesis.

"Tarski has stressed in his lecture (and I think justly) the great importance of the concept of general recursiveness (or Turing's computability). It seems to me that this importance

is largely due to the fact that with this concept one has for the first time succeeded in giving an absolute notion to an interesting epistemological notion, i.e., one not depending on the formalism chosen."(Gödel 1946 in Davis 1965: 84)

With a definition of effective calculation came the first proofs that there are problems in mathematics that cannot be effectively decided. Church (1936a, 1936b) and Turing (1936), inspired by techniques used by Gödel (1931) to prove his incompleteness theorems, independently demonstrated that the Entscheidungsproblem is not effectively decidable. This result showed that there is no algorithmic procedure that can correctly decide whether arbitrary mathematical propositions are true or false.

Many problems of mathematics have been shown to be undecidable after these initial examples were established. In 1947, Markov and Post published independent papers showing that the word problem for semigroups cannot be effectively decided. Extending this result, Pyotr Novikov and William Boone showed independently in the 1950s that the word problem for groups is not effectively solvable: there is no effective procedure that, given a word in a finitely presented group, will decide whether the element represented by the word is the identity element of the group. In 1970, Yuri Matiyasevich proved Matiyasevich's theorem, which implies that Hilbert's tenth problem has no effective solution; this problem asked whether there is an effective procedure to decide whether a Diophantine equation over the integers has a solution in the integers. The list of undecidable problems gives additional examples of problems with no computable solution.

The study of which mathematical constructions can be effectively performed is sometimes called **recursive mathematics**; the Handbook of Recursive Mathematics (Ershov et al. 1998) covers many of the known results in this field.

Turing computability

The main form of computability studied in recursion theory was introduced by Turing (1936). A set of natural numbers is said to be a **computable set** (also called a **decidable**, **recursive**, or **Turing computable** set) if there is a Turing machine that, given a number n, halts with output 1 if n is in the set and halts with output 0 if n is not in the set. A function f from the natural numbers to themselves is a **recursive** or **(Turing) computable function** if there is a Turing machine that, on input n, halts and returns output f(n). The use of Turing machines here is not necessary; there are many other models of computation that have the same computing power as Turing machines; for example the μ-recursive functions obtained from primitive recursion and the μ operator.

The terminology for recursive functions and sets is not completely standardized. The definition in terms of μ-recursive functions as well as a different definition of rekursiv functions by Gödel led to the traditional name recursive for sets and functions computable by a Turing machine. The word **decidable** stems from the German word **Entscheidungsproblem** which was used in the original papers of Turing and others. In contemporary use, the term "computable function" has various definitions: according to

Cutland (1980), it is a partial recursive function (which can be undefined for some inputs), while according to Soare (1987) it is a total recursive (equivalently, general recursive) function.

Not every set of natural numbers is computable. The halting problem, which is the set of (descriptions of) Turing machines that halt on input 0, is a well known example of a noncomputable set. The existence of many noncomputable sets follows from the facts that there are only countably many Turing machines, and thus only countably many computable sets, but there are uncountably many sets of natural numbers.

Although the Halting problem is not computable, it is possible to simulate program execution and produce an infinite list of the programs that do halt. Thus the halting problem is an example of a **recursively enumerable set**, which is a set that can be enumerated by a Turing machine (other terms for recursively enumerable include **computably enumerable** and **semidecidable**). Equivalently, a set is recursively enumerable if and only if it is the range of some computable function. The recursively enumerable sets, although not decidable in general, have been studied in detail in recursion theory.

Areas of research in recursion theory

Beginning with the theory of recursive sets and functions described above, the field of recursion theory has grown to include the study of many closely related topics. These are not independent areas of research: each of these areas draws ideas and results from the others, and most recursion theorists are familiar with the majority of them.

Relative computability and the Turing degrees

Recursion theory in mathematical logic has traditionally focused on **relative computability**, a generalization of Turing computability defined using oracle Turing machines, introduced by Turing (1939). An oracle Turing machine is a hypothetical device which, in addition to performing the actions of a regular Turing machine, is able to ask questions of an **oracle**, which is a particular set of natural numbers. The oracle machine may only ask questions of the form "Is n in the oracle set?". Each question will be immediately answered correctly, even if the oracle set is not computable. Thus an oracle machine with a noncomputable oracle will be able to compute sets that are not computable without an oracle.

Informally, a set of natural numbers A is **Turing reducible** to a set B if there is an oracle machine that correctly tells whether numbers are in A when run with B as the oracle set (in this case, the set A is also said to be (**relatively**) **computable from** B and **recursive in** B). If a set A is Turing reducible to a set B and B is Turing reducible to A then the sets are said to have the same **Turing degree** (also called **degree of unsolvability**). The Turing degree of a set gives a precise measure of how uncomputable the set is.

The natural examples of sets that are not computable, including many different sets that encode variants of the halting problem, have two properties in common:

1. They are recursively enumerable, and
2. Each can be translated into any other via a many-one reduction. That is, given such sets A and B, there is a total computable function f such that $A = \{x : f(x) \in B\}$. These sets are said to be **many-one equivalent** (or **m-equivalent**).

Many-one reductions are "stronger" than Turing reductions: if a set A is many-one reducible to a set B, then A is Turing reducible to B, but the converse does not always hold. Although the natural examples of noncomputable sets are all many-one equivalent, it is possible to construct recursively enumerable sets A and B such that A is Turing reducible to B but not many-one reducible to B. It can be shown that every recursively enumerable set is many-one reducible to the halting problem, and thus the halting problem is the most complicated recursively enumerable set with respect to many-one reducibility and with respect to Turing reducibility. Post (1944) asked whether every recursively enumerable set is either computable or Turing equivalent to the halting problem, that is, whether there is no recursively enumerable set with a Turing degree intermediate between those two.

As intermediate results, Post defined natural types of recursively enumerable sets like the simple, hypersimple and hyperhypersimple sets. Post showed that these sets are strictly between the computable sets and the halting problem with respect to many-one reducibility. Post also showed that some of them are strictly intermediate under other reducibility notions stronger than Turing reducibility. But Post left open the main problem of the existence of recursively enumerable sets of intermediate Turing degree; this problem became known as **Post's problem**. After ten years, Kleene and Post showed in 1954 that there are intermediate Turing degrees between those of the computable sets and the halting problem, but they failed to show that any of these degrees contains a recursively enumerable set. Very soon after this, Friedberg and Muchnik independently solved Post's problem by establishing the existence of recursively enumerable sets of intermediate degree. This groundbreaking result opened a wide study of the Turing degrees of the recursively enumerable sets which turned out to possess a very complicated and non-trivial structure.

There are uncountably many sets that are not recursively enumerable, and the investigation of the Turing degrees of all sets is as central in recursion theory as the investigation of the recursively enumerable Turing degrees. Many degrees with special properties were constructed: **hyperimmune-free degrees** where every function computable relative to that degree is majorized by a (unrelativized) computable function; **high degrees** relative to which one can compute a function f which dominates every computable function g in the sense that there is a constant c depending on g such that $g(x) < f(x)$ for all $x > c$; **random degrees** containing algorithmically random sets; **1-generic** degrees of 1-generic sets; and the degrees below the halting problem of limit-recursive sets.

The study of arbitrary (not necessarily recursively enumerable) Turing degrees involves the study of the Turing jump. Given a set A, the **Turing jump** of A is a set of natural numbers encoding a solution to the halting problem for oracle Turing machines running with oracle A. The Turing jump of any set is always of higher Turing degree than the original set, and a theorem of Friedburg shows that any set that computes the Halting problem can be obtained as the Turing jump of another set. Post's theorem establishes a close relationship between the Turing jump operation and the arithmetical hierarchy, which is a classification of certain subsets of the natural numbers based on their definability in arithmetic.

Much recent research on Turing degrees has focused on the overall structure of the set of Turing degrees and the set of Turing degrees containing recursively enumerable sets. A deep theorem of Shore and Slaman (1999) states that the function mapping a degree x to the degree of its Turing jump is definable in the partial order of the Turing degrees. A recent survey by Ambos-Spies and Fejer (2006) gives an overview of this research and its historical progression.

Other reducibilities

An ongoing area of research in recursion theory studies reducibility relations other than Turing reducibility. Post (1944) introduced several **strong reducibilities**, so named because they imply truth-table reducibility. A Turing machine implementing a strong reducibility will compute a total function regardless of which oracle it is presented with. **Weak reducibilities** are those where a reduction process may not terminate for all oracles; Turing reducibility is one example.

The strong reducibilities include:

- One-one reducibility: A is **one-one reducible** (or **1-reducible**) to B if there is a total computable injective function f such that each n is in A if and only if f(n) is in B.
- Many-one reducibility: This is essentially one-one reducibility without the constraint that f be injective. A is **many-one reducible** (or **m-reducible**) to B if there is a total computable function f such that each n is in A if and only if f(n) is in B.
- Truth-table reducibility: A is truth-table reducible to B if A is Turing reducible to B via an oracle Turing machine that computes a total function regardless of the oracle it is given. Because of compactness of Cantor space, this is equivalent to saying that the reduction presents a single list of questions (depending only on the input) to the oracle simultaneously, and then having seen their answers is able to produce an output without asking additional questions regardless of the oracle's answer to the initial queries. Many variants of truth-table reducibility have also been studied.

The major research on strong reducibilities has been to compare their theories, both for the class of all recursively enumerable sets as well as for the class of all subsets of the

natural numbers. Furthermore, the relations between the reducibilities has been studied. For example, it is known that every Turing degree is either a truth-table degree or is the union of infinitely many truth-table degrees.

Reducibilities weaker than Turing reducibility (that is, reducibilities that are implied by Turing reducibility) have also been studied. The most well known are arithmetical reducibility and hyperarithmetical reducibility. These reducibilities are closely connected to definability over the standard model of arithmetic.

Rice's theorem and the arithmetical hierarchy

Rice showed that for every nontrivial class C (which contains some but not all r.e. sets) the index set E = {e: the eth r.e. set W_e is in C} has the property that either the halting problem or its complement is many-one reducible to E, that is, can be mapped using a many-one reduction to E. But, many of these index sets are even more complicated than the halting problem. These type of sets can be classified using the arithmetical hierarchy. For example, the index set FIN of class of all finite sets is on the level Σ_2, the index set REC of the class of all recursive sets is on the level Σ_3, the index set COFIN of all cofinite sets is also on the level Σ_3 and the index set COMP of the class of all Turing-complete sets Σ_4. These hierarchy levels are defined inductively, Σ_{n+1} contains just all sets which are recursively enumerable relative to Σ_n; Σ_1 contains the recursively enumerable sets. The index sets given here are even complete for their levels, that is, all the sets in these levels can be many-one reduced to the given index sets.

Reverse mathematics

The program of **reverse mathematics** asks which set-existence axioms are necessary to prove particular theorems of mathematics in subsystems of second-order arithmetic. This study was initiated by Harvey Friedman and was studied in detail by Stephen Simpson and others; Simpson (1999) gives a detailed discussion of the program. The set-existence axioms in question correspond informally to axioms saying that the powerset of the natural numbers is closed under various reducibility notions. The weakest such axiom studied in reverse mathematics is **recursive comprehension**, which states that the powerset of the naturals is closed under Turing reducibility.

Numberings

A numbering is an enumeration of functions; it has two parameters, e and x and outputs the value of the e-th function in the numbering on the input x. Numberings can be partial-recursive although some of its members are total recursive, that is, computable functions. Acceptable or Gödel numberings are those into which all others can be translated. A Friedberg numbering (named after its discoverer) is a one-one numbering of all partial-recursive functions; it is necessarily not an acceptable numbering. Later research dealt also with numberings of other classes like classes of recursively enumerable sets. Goncharov discovered for example a class of recursively enumerable sets for which the numberings fall into exactly two classes with respect to recursive isomorphisms.

The priority method

Post's problem was solved with a method called the **priority method**; a proof using this method is called a **priority argument**. This method is primarily used to construct recursively enumerable sets with particular properties. To use this method, the desired properties of the set to be constructed are broken up into an infinite list of goals, known as **requirements**, so that satisfying all the requirements will cause the set constructed to have the desired properties. Each requirement is assigned to a natural number representing the priority of the requirement; so 0 is assigned to the most important priority, 1 to the second most important, and so on. The set is then constructed in stages, each stage attempting to satisfy one of more of the requirements by either adding numbers to the set or banning numbers from the set so that the final set will satisfy the requirement. It may happen that satisfying one requirement will cause another to become unsatisfied; the priority order is used to decide what to do in such an event.

Priority arguments have been employed to solve many problems in recursion theory, and have been classified into a hierarchy based on their complexity (Soare 1987). Because complex priority arguments can be technical and difficult to follow, it has traditionally been considered desirable to prove results without priority arguments, or to see if results proved with priority arguments can also be proved without them. For example, Kummer published a paper on a proof for the existence of Friedberg numberings without using the priority method.

The lattice of recursively enumerable sets

When Post defined the notion of a simple set as an r.e. set with an infinite complement not containing any infinite r.e. set, he started to study the structure of the recursively enumerable sets under inclusion. This lattice became a well-studied structure. Recursive sets can be defined in this structure by the basic result that a set is recursive if and only if the set and its complement are both recursively enumerable. Infinite r.e. sets have always infinite recursive subsets; but on the other hand, simple sets exist but do not have a coinfinite recursive superset. Post (1944) introduced already hypersimple and hyperhypersimple sets; later maximal sets were constructed which are r.e. sets such that every r.e. superset is either a finite variant of the given maximal set or is co-finite. Post's original motivation in the study of this lattice was to find a structural notion such that every set which satisfies this property is neither in the Turing degree of the recursive sets nor in the Turing degree of the halting problem. Post did not find such a property and the solution to his problem applied priority methods instead; Harrington and Soare (1991) found eventually such a property.

Automorphism problems

Another important question is the existence of automorphisms in recursion-theoretic structures. One of these structures is that one of recursively enumerable sets under inclusion modulo finite difference; in this structure, A is below B if and only if the set difference B − A is finite. Maximal sets (as defined in the previous paragraph) have the

property that they cannot be automorphic to non-maximal sets, that is, if there is an automorphism of the recursive enumerable sets under the structure just mentioned, then every maximal set is mapped to another maximal set. Soare (1974) showed that also the converse holds, that is, every two maximal sets are automorphic. So the maximal sets form an orbit, that is, every automorphism preserves maximality and any two maximal sets are transformed into each other by some automorphism. Harrington gave a further example of an automorphic property: that of the creative sets, the sets which are many-one equivalent to the halting problem.

Besides the lattice of recursively enumerable sets, automorphisms are also studied for the structure of the Turing degrees of all sets as well as for the structure of the Turing degrees of r.e. sets. In both cases, Cooper claims to have constructed nontrivial automorphisms which map some degrees to other degrees; this construction has, however, not been verified and some colleagues believe that the construction contains errors and that the question of whether there is a nontrivial automorphism of the Turing degrees is still one of the main unsolved questions in this area (Slaman and Woodin 1986, Ambos-Spies and Fejer 2006).

Kolmogorov complexity

The field of Kolmogorov complexity and algorithmic randomness was developed during the 1960s and 1970s by Chaitin, Kolmogorov, Levin, Martin-Löf and Solomonoff (the names are given here in alphabetical order; much of the research was independent, and the unity of the concept of randomness was not understood at the time). The main idea is to consider a universal Turing machine U and to measure the complexity of a number (or string) x as the length of the shortest input p such that U(p) outputs x. This approach revolutionized earlier ways to determine when an infinite sequence (equivalently, characteristic function of a subset of the natural numbers) is random or not by invoking a notion of randomness for finite objects. Kolmogorov complexity became not only a subject of independent study but is also applied to other subjects as a tool for obtaining proofs. There are still many open problems in this area. For that reason, a recent research conference in this area was held in January 2007 and a list of open problems is maintained by Joseph Miller and Andre Nies.

Frequency computation

This branch of recursion theory analyzed the following question: For fixed m and n with $0 < m < n$, for which functions A is it possible to compute for any different n inputs $x_1, x_2, ..., x_n$ a tuple of n numbers $y_1,y_2,...,y_n$ such that at least m of the equations $A(x_k) = y_k$ are true. Such sets are known as (m, n)-recursive sets. The first major result in this branch of Recursion Theory is Trakhtenbrot's result that a set is computable if it is (m, n)-recursive for some m, n with $2m > n$. On the other hand, Jockusch's semirecursive sets (which were already known informally before Jockusch introduced them 1968) are examples of a set which is (m, n)-recursive if and only if $2m < n + 1$. There are uncountably many of these sets and also some recursively enumerable but noncomputable sets of this type. Later, Degtev established a hierarchy of recursively

enumerable sets that are (1, n + 1)-recursive but not (1, n)-recursive. After a long phase of research by Russian scientists, this subject became repopularized in the west by Beigel's thesis on bounded queries, which linked frequency computation to the above mentioned bounded reducibilities and other related notions. One of the major results was Kummer's Cardinality Theory which states that a set A is computable if and only if there is an n such that some algorithm enumerates for each tuple of n different numbers up to n many possible choices of the cardinality of this set of n numbers intersected with A; these choices must contain the true cardinality but leave out at least one false one.

Inductive inference

This is the recursion-theoretic branch of learning theory. It is based on Gold's model of learning in the limit from 1967 and has developed since then more and more models of learning. The general scenario is the following: Given a class S of computable functions, is there a learner (that is, recursive functional) which outputs for any input of the form (f(0),f(1),...,f(n)) a hypothesis. A learner M learns a function f if almost all hypotheses are the same index e of f with respect to a previously agreed on acceptable numbering of all computable functions; M learns S if M learns every f in S. Basic results are that all recursively enumerable classes of functions are learnable while the class REC of all computable functions is not learnable. Many related models have been considered and also the learning of classes of recursively enumerable sets from positive data is a topic studied from Gold's pioneering paper in 1967 onwards.

Generalizations of Turing computability

Recursion theory includes the study of generalized notions of this field such as arithmetic reducibility, hyperarithmetical reducibility and α-recursion theory, as described by Sacks (1990). These generalized notions include reducibilities that cannot be executed by Turing machines but are nevertheless natural generalizations of Turing reducibility. These studies include approaches to investigate the analytical hierarchy which differs from the arithmetical hierarchy by permitting quantification over sets of natural numbers in addition to quantification over individual numbers. These areas are linked to the theories of well-orderings and trees; for example the set of all indices of recursive (nonbinary) trees without infinite branches is complete for level Π^1_1 of the analytical hierarchy. Both Turing reducibility and hyperarithmetical reducibility are important in the field of effective descriptive set theory. The even more general notion of degrees of constructibility is studied in set theory.

Continuous computability theory

Computability theory for digital computation is well developed. Computability theory is less well developed for analog computation that occurs in analog computers, analog signal processing, analog electronics, neural networks and continuous-time control theory, modelled by differential equations and continuous dynamical systems.

Relationships between definability, proof and computability

There are close relationships between the Turing degree of a set of natural numbers and the difficulty (in terms of the arithmetical hierarchy) of defining that set using a first-order formula. One such relationship is made precise by Post's theorem. A weaker relationship was demonstrated by Kurt Gödel in the proofs of his completeness theorem and incompleteness theorems. Gödel's proofs show that the set of logical consequences of an effective first-order theory is a recursively enumerable set, and that if the theory is strong enough this set will be uncomputable. Similarly, Tarski's indefinability theorem can be interpreted both in terms of definability and in terms of computability.

Recursion theory is also linked to second order arithmetic, a formal theory of natural numbers and sets of natural numbers. The fact that certain sets are computable or relatively computable often implies that these sets can be defined in weak subsystems of second order arithmetic. The program of reverse mathematics uses these subsystems to measure the noncomputability inherent in well known mathematical theorems. Simpson (1999) discusses many aspects of second-order arithmetic and reverse mathematics.

The field of proof theory includes the study of second-order arithmetic and Peano arithmetic, as well as formal theories of the natural numbers weaker than Peano arithmetic. One method of classifying the strength of these weak systems is by characterizing which computable functions the system can prove to be total. For example, in primitive recursive arithmetic any computable function that is provably total is actually primitive recursive, while Peano arithmetic proves that functions like the Ackerman function, which are not primitive recursive, are total. Not every total computable function is provably total in Peano arithmetic, however; an example of such a function is provided by Goodstein's theorem.

Name of the subject

The field of mathematical logic dealing with computability and its generalizations has been called "recursion theory" since its early days. Robert I. Soare, a prominent researcher in the field, has proposed (Soare 1996) that the field should be called "computability theory" instead. He argues that Turing's terminology using the word "computable" is more natural and more widely understood than the terminology using the word "recursive" introduced by Kleene. Many contemporary researchers have begun to use this alternate terminology. These researchers also use terminology such as partial computable function and computably enumerable (c.e.) set instead of partial recursive function and recursively enumerable (r.e.) set. Not all researchers have been convinced, however, as explained by Fortnow and Simpson. Some commentators argue that both the names recursion theory and computability theory fail to convey the fact that most of the objects studied in recursion theory are not computable.

Rogers (1967) has suggested that a key property of recursion theory is that its results and structures should be invariant under computable bijections on the natural numbers (this suggestion draws on the ideas of the Erlangen program in geometry). The idea is that a computable bijection merely renames numbers in a set, rather than indicating any structure in the set, much as a rotation of the Euclidean plane does not change any geometric aspect of lines drawn on it. Since any two infinite computable sets are linked by a computable bijection, this proposal identifies all the infinite computable sets (the finite computable sets are viewed as trivial). According to Rogers, the sets of interest in recursion theory are the noncomputable sets, partitioned into equivalence classes by computable bijections of the natural numbers.

Chapter- 6

Proof Theory and Model Theory

Proof theory

Proof theory is a branch of mathematical logic that represents proofs as formal mathematical objects, facilitating their analysis by mathematical techniques. Proofs are typically presented as inductively-defined data structures such as plain lists, boxed lists, or trees, which are constructed according to the axioms and rules of inference of the logical system. As such, proof theory is syntactic in nature, in contrast to model theory, which is semantic in nature. Together with model theory, axiomatic set theory, and recursion theory, proof theory is one of the so-called four pillars of the foundations of mathematics.

Proof theory is important in philosophical logic, where the primary interest is in the idea of a proof-theoretic semantics, an idea which depends upon technical ideas in structural proof theory to be feasible.

History

Although the formalisation of logic was much advanced by the work of such figures as Gottlob Frege, Giuseppe Peano, Bertrand Russell, and Richard Dedekind, the story of modern proof theory is often seen as being established by David Hilbert, who initiated what is called Hilbert's program in the foundations of mathematics. Kurt Gödel's seminal work on proof theory first advanced, then refuted this program: his completeness theorem initially seemed to bode well for Hilbert's aim of reducing all mathematics to a finitist formal system; then his incompleteness theorems showed that this is unattainable. All of this work was carried out with the proof calculi called the Hilbert systems.

In parallel, the foundations of structural proof theory were being founded. Jan Łukasiewicz suggested in 1926 that one could improve on Hilbert systems as a basis for the axiomatic presentation of logic if one allowed the drawing of conclusions from assumptions in the inference rules of the logic. In response to this Stanisław Jaśkowski (1929) and Gerhard Gentzen (1934) independently provided such systems, called calculi of natural deduction, with Gentzen's approach introducing the idea of symmetry between the grounds for asserting propositions, expressed in introduction rules, and the

consequences of accepting propositions in the elimination rules, an idea that has proved very important in proof theory . Gentzen (1934) further introduced the idea of the sequent calculus, a calculus advanced in a similar spirit that better expressed the duality of the logical connectives , and went on to make fundamental advances in the formalisation of intuitionistic logic, and provide the first combinatorial proof of the consistency of Peano arithmetic. Together, the presentation of natural deduction and the sequent calculus introduced the fundamental idea of analytic proof to proof theory,

Formal and informal proof

The informal proofs of everyday mathematical practice are unlike the formal proofs of proof theory. They are rather like high-level sketches that would allow an expert to reconstruct a formal proof at least in principle, given enough time and patience. For most mathematicians, writing a fully formal proof is too pedantic and long-winded to be in common use.

Formal proofs are constructed with the help of computers in interactive theorem proving. Significantly, these proofs can be checked automatically, also by computer. (Checking formal proofs is usually simple, whereas finding proofs (automated theorem proving) is generally hard.) An informal proof in the mathematics literature, by contrast, requires weeks of peer review to be checked, and may still contain errors.

Kinds of proof calculi

The three most well-known styles of proof calculi are:

- The Hilbert calculi
- The natural deduction calculi
- The sequent calculi

Each of these can give a complete and axiomatic formalization of propositional or predicate logic of either the classical or intuitionistic flavour, almost any modal logic, and many substructural logics, such as relevance logic or linear logic. Indeed it is unusual to find a logic that resists being represented in one of these calculi.

Consistency proofs

As previously mentioned, the spur for the mathematical investigation of proofs in formal theories was Hilbert's program. The central idea of this program was that if we could give finitary proofs of consistency for all the sophisticated formal theories needed by mathematicians, then we could ground these theories by means of a metamathematical argument, which shows that all of their purely universal assertions (more technically their provable Π^0_1 sentences) are finitarily true; once so grounded we do not care about the

non-finitary meaning of their existential theorems, regarding these as pseudo-meaningful stipulations of the existence of ideal entities.

The failure of the program was induced by Kurt Gödel's incompleteness theorems, which showed that any ω-consistent theory that is sufficiently strong to express certain simple arithmetic truths, cannot prove its own consistency, which on Gödel's formulation is a Π^0_1sentence.

Much investigation has been carried out on this topic since, which has in particular led to:

- Refinement of Gödel's result, particularly J. Barkley Rosser's refinement, weakening the above requirement of ω-consistency to simple consistency;
- Axiomatisation of the core of Gödel's result in terms of a modal language, provability logic;
- Transfinite iteration of theories, due to Alan Turing and Solomon Feferman;
- The recent discovery of self-verifying theories, systems strong enough to talk about themselves, but too weak to carry out the diagonal argument that is the key to Gödel's unprovability argument.

Structural proof theory

Structural proof theory is the subdiscipline of proof theory that studies proof calculi that support a notion of analytic proof. The notion of analytic proof was introduced by Gentzen for the sequent calculus; there the analytic proofs are those that are cut-free. His natural deduction calculus also supports a notion of analytic proof, as shown by Dag Prawitz. The definition is slightly more complex: we say the analytic proofs are the normal forms, which are related to the notion of normal form in term rewriting. More exotic proof calculi such as Jean-Yves Girard's proof nets also support a notion of analytic proof.

Structural proof theory is connected to type theory by means of the Curry-Howard correspondence, which observes a structural analogy between the process of normalisation in the natural deduction calculus and beta reduction in the typed lambda calculus. This provides the foundation for the intuitionistic type theory developed by Per Martin-Löf, and is often extended to a three way correspondence, the third leg of which are the cartesian closed categories.

Model theory

In mathematics, **model theory** is the study of (classes of) mathematical structures such as groups, fields, graphs, or even universes of set theory, using tools from mathematical logic. A structure that gives meaning to the sentences of a formal language is called a model for the language. If a model for a language moreover satisfies a particular sentence or theory (set of sentences), it is called a model of the sentence or theory. Model theory has close ties to algebra and universal algebra.

Finite model theory, which concentrates on finite structures, diverges significantly from the study of infinite structures in both the problems studied and the techniques used. Model theory in higher-order logics or infinitary logics is hampered by the fact that completeness does not in general hold for these logics. However, a great deal of study has also been done in such languages.

Introduction

Model theory recognises and is intimately concerned with a duality: It examines semantical elements by means of syntactical elements of a corresponding language. To quote the first page of Chang and Keisler (1990):

universal algebra + logic = **model theory**.

In a similar way to proof theory, model theory is situated in an area of interdisciplinarity between mathematics, philosophy, and computer science. The most important professional organization in the field of model theory is the Association for Symbolic Logic.

An incomplete and somewhat arbitrary subdivision of model theory is into classical model theory, model theory applied to groups and fields, and geometric model theory. A missing subdivision is computable model theory, but this can arguably be viewed as an independent subfield of logic. Examples of early theorems from classical model theory include Gödel's completeness theorem, the upward and downward Löwenheim–Skolem theorems, Vaught's two-cardinal theorem, Scott's isomorphism theorem, the omitting types theorem, and the Ryll-Nardzewski theorem. Examples of early results from model theory applied to fields are Tarski's elimination of quantifiers for real closed fields, Ax's theorem on pseudo-finite fields, and Robinson's development of nonstandard analysis. An important step in the evolution of classical model theory occurred with the birth of stability theory (through Morley's theorem on uncountably categorical theories and Shelah's classification program), which developed a calculus of independence and rank based on syntactical conditions satisfied by theories. During the last several decades applied model theory has repeatedly merged with the more pure stability theory. The result of this synthesis is called geometric model theory here (which is taken to include o-minimality, for example, as well as classical geometric stability theory). An example of a theorem from geometric model theory is Hrushovski's proof of the Mordell–Lang

conjecture for function fields. The ambition of geometric model theory is to provide a geography of mathematics by embarking on a detailed study of definable sets in various mathematical structures, aided by the substantial tools developed in the study of pure model theory.

Example

To illustrate the basic relationship involving syntax and semantics in the context of a non-trivial models, one can start, on the syntactic side, with suitable axioms for the natural numbers such as Peano axioms, and the associated theory. Going on to the semantic side, one has the usual counting numbers as a model. In the 1930s, Skolem developed alternative models satisfying the axioms. This illustrates what is meant by interpretating a language or theory in a particular model. A more traditional example is interpreting the axioms of a particular algebraic system such as a group, in the context of a model provided by a specific group.

Universal algebra

Fundamental concepts in universal algebra are signatures σ and σ-algebras. Since these concepts are formally defined in the article on structures, the present article can content itself with an informal introduction which consists in examples of how these terms are used.

The standard signature of rings is $\sigma_{ring} = \{\times,+,-,0,1\}$, where × and + are binary, − is unary, and 0 and 1 are nullary.
The standard signature of semirings is $\sigma_{smr} = \{\times,+,0,1\}$, where the arities are as above.
The standard signature of (multiplicative) groups is $\sigma_{grp} = \{\times,^{-1},1\}$, where × is binary, $^{-1}$ is unary and 1 is nullary.
The standard signature of monoids is $\sigma_{mnd} = \{\times,1\}$.
A ring is a σ_{ring}-structure which satisfies the identities $u + (v + w) = (u + v) + w$, $u + v = v + u$, $u + 0 = u$, $u + (-u) = 0$, $u \times (v \times w) = (u \times v) \times w$, $u \times 1 = u$, $1 \times u = u$, $u \times (v + w) = (u \times v) + (u \times w)$ and $(v + w) \times u = (v \times u) + (w \times u)$.
A group is a σ_{grp}-structure which satisfies the identities $u \times (v \times w) = (u \times v) \times w$, $u \times 1 = u$, $1 \times u = u$, $u \times u^{-1} = 1$ and $u^{-1} \times u = 1$.
A monoid is a σ_{mnd}-structure which satisfies the identities $u \times (v \times w) = (u \times v) \times w$, $u \times 1 = u$ and $1 \times u = u$.
A semigroup is a σ_{mnd}-structure which satisfies the identity $u \times (v \times w) = (u \times v) \times w$.
A magma is just a $\{\times\}$-structure.

This is a very efficient way to define most classes of algebraic structures, because there is also the concept of σ-homomorphism, which correctly specializes to the usual notions of homomorphism for groups, semigroups, magmas and rings. For this to work, the signature must be chosen well.

Terms such as the σ_{ring}-term t(u,v,w) given by (u + (v × w)) + (−1) are used to define identities t = t', but also to construct free algebras. An equational class is a class of structures which, like the examples above and many others, is defined as the class of all σ-structures which satisfy a certain set of identities. Birkhoff's theorem states:

A class of σ-structures is an equational class if and only if it is not empty and closed under subalgebras, homomorphic images, and direct products.

An important non-trivial tool in universal algebra are ultraproducts $\Pi_{i \in I} A_i / U$, where I is an infinite set indexing a system of σ-structures A_i, and U is an ultrafilter on I.

While model theory is generally considered a part of mathematical logic, universal algebra, which grew out of Alfred North Whitehead's (1898) work on abstract algebra, is part of algebra. This is reflected by their respective MSC classifications. Nevertheless model theory can be seen as an extension of universal algebra.

Finite model theory

Finite model theory is the area of model theory which has the closest ties to universal algebra. Like some parts of universal algebra, and in contrast with the other areas of model theory, it is mainly concerned with finite algebras, or more generally, with finite σ-structures for signatures σ which may contain relation symbols as in the following example:

The standard signature for graphs is σ_{grph}={E}, where E is a binary relation symbol. A graph is a σ_{grph}-structure satisfying the sentences $\forall u \forall v (uEv \rightarrow vEu)$ and $\forall u \neg (uEu)$.

A σ-homomorphism is a map that commutes with the operations and preserves the relations in σ. This definition gives rise to the usual notion of graph homomorphism, which has the interesting property that a bijective homomorphism need not be invertible. Structures are also a part of universal algebra; after all, some algebraic structures such as ordered groups have a binary relation <. What distinguishes finite model theory from universal algebra is its use of more general logical sentences (as in the example above) in place of identities. (In a model-theoretic context an identity t=t' is written as a sentence $\forall u_1 u_2 \ldots u_n (t = t')$.)

The logics employed in finite model theory are often substantially more expressive than first-order logic, the standard logic for model theory of infinite structures.

First-order logic

Whereas universal algebra provides the semantics for a signature, logic provides the syntax. With terms, identities and quasi-identities, even universal algebra has some

limited syntactic tools; first-order logic is the result of making quantification explicit and adding negation into the picture.

A first-order **formula** is built out of atomic formulas such as R(f(x,y),z) or y = x + 1 by means of the Boolean connectives $\neg, \wedge, \vee, \rightarrow$ and prefixing of quantifiers $\forall v$ or $\exists v$. A sentence is a formula in which each occurrence of a variable is in the scope of a corresponding quantifier. Examples for formulas are φ (or φ(x) to mark the fact that at most x is an unbound variable in φ) and ψ defined as follows:

$$\phi = \forall u \forall v (\exists w (x \times w = u \times v) \rightarrow (\exists w (x \times w = u) \vee \exists w (x \times w = v))) \wedge x \neq 0 \wedge x \neq 1,$$
$$\psi = \forall u \forall v ((u \times v = x) \rightarrow (u = x) \vee (v = x)) \wedge x \neq 0 \wedge x \neq 1.$$

(Note that the equality symbol has a double meaning here.) It is intuitively clear how to translate such formulas into mathematical meaning. In the σ_{smr}-structure $\mathcal{N}$ of the natural numbers, for example, an element n **satisfies** the formula φ if and only if n is a prime number. The formula ψ similarly defines irreducibility. Tarski gave a rigorous definition, sometimes called "Tarski's definition of truth", for the satisfaction relation $\models$, so that one easily proves:

$$\mathcal{N} \models \phi(n) \iff n \text{ is a prime number.}$$
$$\mathcal{N} \models \psi(n) \iff n \text{ is irreducible.}$$

A set T of sentences is called a (first-order) theory. A theory is **satisfiable** if it has a **model** $\mathcal{M} \models T$, i.e. a structure (of the appropriate signature) which satisfies all the sentences in the set T. Consistency of a theory is usually defined in a syntactical way, but in first-order logic by the completeness theorem there is no need to distinguish between satisfiability and consistency. Therefore model theorists often use "consistent" as a synonym for "satisfiable".

A theory is called **categorical** if it determines a structure up to isomorphism, but it turns out that this definition is not useful, due to serious restrictions in the expressivity of first-order logic. The Löwenheim–Skolem theorem implies that for every theory T which has an infinite model and for every infinite cardinal number κ, there is a model $\mathcal{M} \models T$ such that the number of elements of $\mathcal{M}$ is exactly κ. Therefore only finite structures can be described by a categorical theory.

Lack of expressivity (when compared to higher logics such as second-order logic) has its advantages, though. For model theorists the Löwenheim–Skolem theorem is an important practical tool rather than the source of Skolem's paradox. First-order logic is in some sense the most expressive logic for which both the Löwenheim–Skolem theorem and the compactness theorem hold:

Compactness theorem
Every unsatisfiable first-order theory has a finite unsatisfiable subset.

This important theorem, due to Gödel, is of central importance in infinite model theory, where the words "by compactness" are commonplace. One way to prove it is by means of ultraproducts. An alternative proof uses the completeness theorem, which is otherwise reduced to a marginal role in most of modern model theory.

Axiomatizability, elimination of quantifiers, and model-completeness

The first step, often trivial, for applying the methods of model theory to a class of mathematical objects such as groups, or trees in the sense of graph theory, is to choose a signature σ and represent the objects as σ-structures. The next step is to show that the class is an elementary class, i.e. axiomatizable in first-order logic (i.e. there is a theory T such that a σ-structure is in the class if and only if it satisfies T). E.g. this step fails for the trees, since connectedness cannot be expressed in first-order logic. Axiomatizability ensures that model theory can speak about the right objects. Quantifier elimination can be seen as a condition which ensures that model theory does not say too much about the objects.

A theory T has quantifier elimination if every first-order formula $\varphi(x_1,...,x_n)$ over its signature is equivalent modulo T to a first-order formula $\psi(x_1,...,x_n)$ without quantifiers, i.e. $\forall x_1 \ldots \forall x_n (\phi(x_1, \ldots, x_n) \leftrightarrow \psi(x_1, \ldots, x_n))$ holds in all models of T. For example the theory of algebraically closed fields in the signature $\sigma_{ring}=(\times,+,-,0,1)$ has quantifier elimination because every formula is equivalent to a Boolean combination of equations between polynomials.

A substructure of a σ-structure is a subset of its domain, closed under all functions in its signature σ, which is regarded as a σ-structure by restricting all functions and relations in σ to the subset. An embedding of a σ-structure $\mathcal{A}$ into another σ-structure $\mathcal{B}$ is a map f: A → B between the domains which can be written as an isomorphism of $\mathcal{A}$ with a substructure of $\mathcal{B}$. Every embedding is an injective homomorphism, but the converse holds only if the signature contains no relation symbols.

If a theory does not have quantifier elimination, one can add additional symbols to its signature so that it does. Early model theory spent much effort on proving axiomatizability and quantifier elimination results for specific theories, especially in algebra. But often instead of quantifier elimination a weaker property suffices:

A theory T is called model-complete if every substructure of a model of T which is itself a model of T is an elementary substructure. There is a useful criterion for testing whether a substructure is an elementary substructure, called the Tarski–Vaught test. It follows from this criterion that a theory T is model-complete if and only if every first-order formula $\varphi(x_1,...,x_n)$ over its signature is equivalent modulo T to an existential first-order formula, i.e. a formula of the following form:

$$\exists v_1 \ldots \exists v_m \psi(x_1, \ldots, x_n, v_1, \ldots, v_m),$$

where ψ is quantifier free. A theory that is not model-complete may or may not have a **model completion**, which is a related model-complete theory that is not, in general, an extension of the original theory. A more general notion is that of **model companions**.

Categoricity

As observed in the section on first-order logic, first-order theories cannot be categorical, i.e. they cannot describe a unique model up to isomorphism, unless that model is finite. But two famous model-theoretic theorems deal with the weaker notion of κ-categoricity for a cardinal κ. A theory T is called **κ-categorical** if any two models of T that are of cardinality κ are isomorphic. It turns out that the question of κ-categoricity depends critically on whether κ is bigger than the cardinality of the language (i.e. $\aleph_0 + |\sigma|$, where $|\sigma|$ is the cardinality of the signature). For finite or countable signatures this means that there is a fundamental difference between $\aleph_0$-cardinality and κ-cardinality for uncountable κ.

The following characterization of $\aleph_0$-categoricity is due independently to Ryll-Nardzewski, Engeler and Svenonius:

Ryll-Nardzewski's theorem
For a complete first-order theory T in a finite or countable signature the following conditions are equivalent:

1. T is $\aleph_0$-categorical.
2. For every natural number n, the Stone space $S_n(T)$ is finite.
3. For every natural number n, the number of formulas $\varphi(x_1, ..., x_n)$ in n free variables, up to equivalence modulo T, is finite.

$\aleph_0$-categorical theories and their countable models have strong ties with oligomorphic groups. They are often constructed as Fraïssé limits.

Michael Morley's highly non-trivial result that (for countable languages) there is only one notion of uncountable categoricity was the starting point for modern model theory, and in particular classification theory and stability theory:

Morley's categoricity theorem
If a first-order theory T in a finite or countable signature is κ-categorical for some uncountable cardinal κ, then T is κ-categorical for all uncountable cardinals κ.

Uncountably categorical (i.e. κ-categorical for all uncountable cardinals κ) theories are from many points of view the most well-behaved theories. A theory that is both $\aleph_0$-categorical and uncountably categorical is called **totally categorical**.

Model theory and set theory

Set theory (which is expressed in a countable language) has a countable model; this is known as Skolem's paradox, since there are sentences in set theory which postulate the existence of uncountable sets and yet these sentences are true in our countable model. Particularly the proof of the independence of the continuum hypothesis requires considering sets in models which appear to be uncountable when viewed from within the model, but are countable to someone outside the model.

The model-theoretic viewpoint has been useful in set theory; for example in Kurt Gödel's work on the constructible universe, which, along with the method of forcing developed by Paul Cohen can be shown to prove the (again philosophically interesting) independence of the axiom of choice and the continuum hypothesis from the other axioms of set theory.

Other basic notions of model theory

Reducts and expansions

A field or a vector space can be regarded as a (commutative) group by simply ignoring some of its structure. The corresponding notion in model theory is that of a **reduct** of a structure to a subset of the original signature. The opposite relation is called an expansion - e.g. the (additive) group of the rational numbers, regarded as a structure in the signature $\{+,0\}$ can be expanded to a field with the signature $\{\times,+,1,0\}$ or to an ordered group with the signature $\{+,0,<\}$.

Similarly, if σ' is a signature that extends another signature σ, then a complete σ'-theory can be restricted to σ by intersecting the set of its sentences with the set of σ-formulas. Conversely, a complete σ-theory can be regarded as a σ'-theory, and one can extend it (in more than one way) to a complete σ'-theory. The terms reduct and expansion are sometimes applied to this relation as well.

Interpretability

Given a mathematical structure, there are very often associated structures which can be constructed as a quotient of part of the original structure via an equivalence relation. An important example is a quotient group of a group.

One might say that to understand the full structure one must understand these quotients. When the equivalence relation is definable, we can give the previous sentence a precise meaning. We say that these structures are **interpretable**.

A key fact is that one can translate sentences from the language of the interpreted structures to the language of the original structure. Thus one can show that if a structure M interprets another whose theory is undecidable, then M itself is undecidable.

Using the compactness and completeness theorems

Gödel's completeness theorem (not to be confused with his incompleteness theorems) says that a theory has a model if and only if it is consistent, i.e. no contradiction is proved by the theory. This is the heart of model theory as it lets us answer questions about theories by looking at models and vice-versa. One should not confuse the completeness theorem with the notion of a complete theory. A complete theory is a theory that contains every sentence or its negation. Importantly, one can find a complete consistent theory extending any consistent theory. However, as shown by Gödel's incompleteness theorems only in relatively simple cases will it be possible to have a complete consistent theory that is also recursive, i.e. that can be described by a recursively enumerable set of axioms. In particular, the theory of natural numbers has no recursive complete and consistent theory. Non-recursive theories are of little practical use, since it is undecidable if a proposed axiom is indeed an axiom, making proof-checking a supertask.

The compactness theorem states that a set of sentences S is satisfiable if every finite subset of S is satisfiable. In the context of proof theory the analogous statement is trivial, since every proof can have only a finite number of antecedents used in the proof. In the context of model theory, however, this proof is somewhat more difficult. There are two well known proofs, one by Gödel (which goes via proofs) and one by Malcev (which is more direct and allows us to restrict the cardinality of the resulting model).

Model theory is usually concerned with first-order logic, and many important results (such as the completeness and compactness theorems) fail in second-order logic or other alternatives. In first-order logic all infinite cardinals look the same to a language which is countable. This is expressed in the Löwenheim–Skolem theorems, which state that any countable theory with an infinite model $\mathfrak{A}$ has models of all infinite cardinalities (at least that of the language) which agree with $\mathfrak{A}$ on all sentences, i.e. they are 'elementarily equivalent'.

Types

Fix an L-structure M, and a natural number n. The set of definable subsets of M^n over some parameters A is a Boolean algebra. By Stone's representation theorem for Boolean algebras there is a natural dual notion to this. One can consider this to be the topological space consisting of maximal consistent sets of formulae over A. We call this the space of (complete) n-types over A, and write $S_n(A)$.

Now consider an element $m \in M^n$. Then the set of all formulae φ with parameters in A in free variables $x_1, \ldots, x_n$ so that $M \models \phi(m)$ is consistent and maximal such. It is called the type of m over A.

One can show that for any n-type p, there exists some elementary extension N of M and some $a \in N^n$ so that p is the type of a over A.

Many important properties in model theory can be expressed with types. Further many proofs go via constructing models with elements that contain elements with certain types and then using these elements.

Illustrative Example: Suppose M is an algebraically closed field. The theory has quantifier elimination . This allows us to show that a type is determined exactly by the polynomial equations it contains. Thus the space of n-types over a subfield A is bijective with the set of prime ideals of the polynomial ring $A[x_1, \ldots, x_n]$. This is the same set as the spectrum of $A[x_1, \ldots, x_n]$. Note however that the topology considered on the type space is the constructible topology: a set of types is basic open iff it is of the form $\{p : f(x) = 0 \in p\}$or of the form $\{p : f(x) \neq 0 \in p\}$. This is finer than the Zariski topology.