METAMATHEMATICS, MACHINES, AND GÖDEL'S PROOF

N. SHANKAR
SRI International

CAMBRIDGE
UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge CB2 1RP, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, United Kingdom
40 West 20th Street, New York, NY 10011-4211, USA
10 Stamford Road, Oakleigh, Melbourne 3166, Australia

First published 1994
First paperback edition 1997

A catalogue record for this book is available from the British Library

Library of Congress Cataloguing in Publication data available

ISBN 0 521 58533 3 paperback

Transferred to digital printing 2002

To My Parents

Contents

List of Figures

Chapter 1

Introduction

Checking mathematical proofs is potentially one of the most interesting and useful applications of automatic computers.
John McCarthy [McC62]

Very few mathematical statements can be judged to be true or false solely by means of direct observation. Some statements, the *axioms*, have to be accepted as true without too much further argument. Other statements, the *theorems*, are believed because they are seen as logical consequences of the axioms by means of a *proof*. Proofs constitute the only effective mechanism for revealing truth in mathematics. We would naturally hope that all provable statements are indeed true. Most of us would also optimistically believe that any true statement has a proof, but such is not the case. Gödel showed that for any reasonably powerful formal system of axioms and inference rules, there are statements that can neither be proved nor disproved on the basis of the axioms and inference rules, and are therefore *undecidable*. Gödel also showed that for such formal systems, there could be no absolute proof that all provable statements were in fact true. In his proof, Gödel described a machine that could check if a given construction constituted a valid proof. It was hoped that one could similarly define a machine to discover the proof itself, but Church and Turing showed that such a machine could not exist. We show in this book that a machine, the Boyer–Moore theorem prover, can be used to check Gödel's proof of the existence of undecidable sentences. Our mechanical verification of Gödel's proof can be seen as an instance of a machine establishing the limitation of mechanism itself, thus showing in concrete terms that machines are not, in this sense, limited.

The notion of a proof as a *logical deduction* of a theorem from the *axioms* was first popularized by Euclid in his *Elements* [Euc56]. For a long time thereafter, there was no rigorous definition of what constituted a valid logical deduction. In the seventeenth century, Leibniz advanced the notion of a universal symbolic logic that he hoped would be applied to mathematical and scientific reasoning, and also to philosophy, law, and politics.[1] In the middle of

[1]Leibniz's actual words are memorable [Lei65]:

> What must be achieved is in fact this: that every paralogism be recognized as an *error of calculation*, and every *sophism* when expressed in this new kind of notation, appear as a *solecism* or *barbarism*, to be corrected easily by the laws of this philosophical grammar.
>
> Once this is done, then when a controversy arises, disputation will no more be needed between

the nineteenth century, Boole, together with de Morgan, laid out the algebraic laws obeyed by truth-valued propositions and the propositional connectives [Boo54]. In the second half of the nineteenth century, Dedekind developed an axiomatic system for proving properties of numbers, and Cantor developed set theory as a framework for all of mathematics. The first rigorous definition of a valid logical proof was given by Frege in his *Begriffsschrift* [Fre67], where he gave syntactic rules for distinguishing the valid proofs from the invalid ones.[2] He devised a symbolic language, the *predicate calculus*, in which mathematical statements could be unambiguously expressed in terms of variables, functions, predicates, propositional connectives, and universal and existential quantification. In formulations of the predicate calculus, some statements, such as "*A implies A*," are taken to be axioms. Theorems are derived starting from the axioms by repeated application of the rules of inference which prescribe how proofs are to be constructed. The well-known rule of *modus ponens* allows the derivation of a proof of *B* from proofs of *A* and *A implies B*. The derivation of a theorem starting from the axioms and using the rules of inference is termed a *formal proof* since it is defined solely in terms of the *form* or the syntax of the statements involved. The language, axioms, and rules constitute a *formal theory.* Examples of concepts formalized by formal theories include geometry, numbers, and sets.

Once the notion of a proof has been made formal, two important consequences follow. First, a computer can be programmed as an *automated proof checker* to check whether a purported formal proof is correct according to the axioms and inference rules. Second, these formal proofs can themselves be made the objects of mathematical study. This study was given the name *metamathematics* by David Hilbert. Theorems in metamathematics typically analyze the properties of formal theories, their inter-relationships, and the relationship between the form and the provability of statements within specific theories. For example, the tautology theorem asserts that all tautologously true statements of predicate calculus can be formally proved. Similarly, Gödel's first incompleteness theorem asserts of a theory such as formal number theory that it is either inconsistent or contains an undecidable sentence that can neither be proved nor disproved.

This book is about the interplay between automated proof checking and metamathematics. We describe a project aimed at constructing and mechanically verifying several substantial proofs in metamathematics using an automated proof checker known as the Boyer–Moore theorem prover.[3] The proofs thus formally verified include some of the landmarks

two philosophers than between two computers. It will suffice that, pen in hand, they sit down to their abacus and (calling in a friend, if they so wish) say to each other: *let us calculate.*

[2]To quote from van Heijenoort's prefatory remarks in *From Frege to Gödel* [vH67]:

> This is the first work that Frege wrote in the field of logic, and, although a mere booklet of eighty-eight pages, it is perhaps the most important single work ever written in logic. Its fundamental contributions, among lesser points, are the truth-functional propositional calculus, the analysis of the proposition into function and argument(s) instead of subject and predicate, the theory of quantification, a system of logic in which derivations are carried out exclusively according to the form of the expressions, and a logical definition of the notion of a mathematical sequence. Any one of these achievements would suffice to secure the book a permanent place in the logician's library.

[3]It could be argued that the mechanically verified proofs described here are not conventional formal proofs

of metamathematics: the tautology theorem, Gödel's first incompleteness theorem, and the Church–Rosser theorem of the lambda calculus. We thus demonstrate that the technology of automated proof checking is sufficiently well-developed that it is possible to verify substantial proofs in metamathematics. If complex and substantial mathematical arguments can be mechanically verified, then it is conceivable that automated proof checking technology will eventually be used as a reasoning aid and even employed as part of the refereeing process for journal publications. The technology is not yet ripe enough for such a task but there are no obvious insurmountable obstacles. Mechanically verified proofs in metamathematics themselves have significant relevance for automated proof checking. Theorems in metamathematics make it possible to obtain more sophisticated but sound inference procedures from simpler ones. The ability to add such inference rules is essential if high-level mathematical arguments are to be verified with a reasonable amount of effort.[4]

The remainder of this introduction presents the background material in metamathematics and automated reasoning and provides an overview of the chapters to follow. We also give a brief introduction to the Boyer–Moore theorem prover and its logic.

1.1 Background

The relevant literature on logic, metamathematics, and automated reasoning is obviously voluminous and only a few significant sources are enumerated here. One of the more readable outlines of Gödel's incompleteness theorem is Gödel's original paper [Göd67b] entitled "On formally undecidable propositions in Principia mathematica and related systems I."[5] This paper has been widely reprinted and an English translation [Göd92] has recently been published by Dover Publications as a book in 1992. Two good sources for Gödel's paper and related material are:

- *The Undecidable*, edited by Davis [Dav65], and
- *From Frege to Gödel: A Sourcebook in Mathematical Logic, 1879–1931*, edited by van Heijenoort [vH67].

Gödel's paper also appears in *Kurt Gödel: Complete Works Volume I* edited by Feferman *et al.* [FJWDK$^+$86]. Topics related to the incompleteness theorem are discussed by Smorynski [Smo78] and by Smullyan [Smu92] who has also given entertaining explanations of Gödel's theorems in several popular books.

Kleene's ***Introduction to Metamathematics*** [Kle52], Shoenfield's ***Mathematical Logic*** [Sho67], and Cohen's ***Set Theory and the Continuum Hypothesis*** [Coh66], provided some of the primary source material for the present project.

in any well-known formal proof system since the Boyer–Moore theorem prover employs a number of complex decision procedures and heuristics. However, these proofs are, both in principle and in practice, formalizable.

[4]Frege foresaw this possibility when he wrote [Fre67]:

> ...when the *foundations* for such an ideography are laid, the primitive components must be taken as simple as possible, if perspicuity and order are to be created. This does not preclude the possibility that *later* certain transitions from several judgments to a new one, transitions that this one mode of inference would not allow us to carry out except mediately, will be abbreviated in immediate ones. In fact this would be advisable in case of eventual application.

[5]Since Gödel's results were quickly accepted, the planned second part of the paper was never written.

The main references on the Boyer–Moore theorem prover are the books *A Computational Logic* [BM79] and *A Computational Logic Handbook* [BM88], by Boyer and Moore. The book *Symbolic Logic and Mechanical Theorem Proving* by Chang and Lee [CL73] is a good reference for material on resolution theorem proving.

Other general texts on mathematical logic include *A Mathematical Introduction to Logic* by Enderton [End72], and *Computability and Logic* by Boolos and Jeffrey [BJ89].

Books on the lambda calculus include Church's *The Calculi of Lambda-Conversion* [Chu41], Barendregt's *The Lambda Calculus* [Bar78a], and *Introduction to Combinators and λ-Calculus* by Hindley and Seldin [HS86].

The rest of this section informally presents the relevant background on language, logic, metamathematics, and automated reasoning.

1.1.1 The Paradoxes

Dedekind, Cantor, Peano, and Frege were seeking a uniform foundational framework for mathematics in terms of a language and a deductive system. The difficulty of this task can be illustrated by means of several well-known paradoxes. Many of these exploit the confusion between language, metalanguage, and semantics. The Liar paradox [Mar84] has a Cretan asserting, "Cretans always lie." If indeed Cretans always lie, then the above statement is true. Then, however, since this statement is true and spoken by a Cretan, it cannot be the case that Cretans always lie. So if the sentence is true, then it is also false. On the other hand, if there is exactly one Cretan whose only utterance is the above statement, then if this statement is false, it is also true. The Liar paradox is a semantical paradox since it employs notions of meaning, truth, and falsity in its statement. The paradox suggests that these notions are not accurately definable for a language within the language itself [Tar83].

Berry's paradox defines a number as "the least natural number not describable in fewer than ninety letters from the English alphabet." Since there are only finitely many numbers describable in fewer than ninety letters from the English alphabet, a least such number must exist. However, if such a number does exist, then we have in fact succeeded in describing it in fewer than ninety letters by the phrase in quotes. Berry's paradox is another semantical paradox.

Russell's paradox evokes the idea of a library catalog that lists all those library catalogs that do not list themselves. If this catalog does not list itself, then it would be incomplete and hence would not have listed *all* of those catalogs that do not list themselves. On the other hand, if it does list itself, then it is erroneous since it has then listed a catalog that lists itself. Russell's paradox can actually be put into a mathematical form. It does not employ any obviously semantical notions, and it yields a contradiction in Frege's theory of arithmetic [GB80, Rus67].

Thus, paradoxes do appear in seemingly natural formalizations of mathematical reasoning and they lead, as shown above, to contradictions. The challenge for a foundation of mathematics is to formalize all of mathematics while avoiding the contradictions resulting from such paradoxes. The constructions of undecidable sentences in proofs of Gödel's incompleteness theorem are also inspired by the paradoxes.

1.1.2 Foundations of Mathematics

If mathematics relies on logical deduction rather than direct observation as a means of grasping the truth, then, as the above paradoxes demonstrate, it is crucial to identify the principles of correct mathematical reasoning. Euclid [Euc56] identified such a collection of "self-evident" postulates for proving theorems in plane geometry, but used informal mathematical reasoning to construct proofs. The discovery of non-Euclidean geometry in the early nineteenth century cast doubt on the self-evidence of Euclid's postulates. This discovery and the general increase in mathematical rigor led several mathematicians to carefully identify certain correct reasoning principles and to ensure that these were in some sense minimal. The hope was that such a systematization of mathematical reasoning would also make it possible to find the errors in a mathematical proof in a systematic manner. Dedekind [Ded63] carried out such a development for number theory where he carefully developed axioms for the natural numbers and reduced the theory of rational and real numbers to a combination of set theory and number theory. Peano [Pea67] gave number-theoretic axioms similar to those of Dedekind and employed an elegant logical notation that has since become standard in mathematics. Cantor [Can55] went even further and reduced a great deal of mathematics to a set theory that contained transfinite ordinal and cardinal numbers.

Logicism. Towards the end of the nineteenth century, the question arose as to whether all mathematical truths could be deduced purely from simple, self-evident logical principles. This was Frege's aim when he first set out the predicate calculus in his *Begriffsschrift* [Fre67] in 1879, and later attempted to develop arithmetic from purely logical principles. Russell found an inconsistency in Frege's axiomatization of arithmetic. Whitehead and Russell stuck to the logicist line and attempted to develop a purely logical theory of mathematics in the *Principia Mathematica* [WR25]. They developed a somewhat baroque theory of types and showed that a significant amount of mathematics could be rigorously derived from their axioms. It would be difficult to argue that the system of *Principia Mathematica* employed only logical principles. There have been attempts to revive logicism but none that are very convincing.

Intuitionism. In the late nineteenth and early twentieth century, Kronecker, then Poincaré, and eventually Brouwer, became heavily skeptical about the purity of the methods of proof employed in large parts of mathematics. The crux of the attack was with the use of infinite entities such as the set of natural numbers. Mathematics is replete with proofs where the existence of a mathematical object satisfying a property is demonstrated by showing that its non-existence yields a contradiction.[6] This, the critics argued, is fine when the domain is finite since all possible candidates can be examined in order to find one satisfying the required property. No such exhaustive search is possible with infinite domains and hence the argument by contradiction does not yield a constructive demonstration of existence.

Brouwer's critique led him to formulate intuitionism as a pure approach to mathematics that only used constructive methods and treated infinite structures as potential rather

[6]The classic example is a proof that there exist irrational numbers x and y such that x^y is rational. The argument is that either $\sqrt{2}^{\sqrt{2}}$ is rational, in which case x and y can both be taken as $\sqrt{2}$, or we pick x to be $\sqrt{2}^{\sqrt{2}}$ and y to be $\sqrt{2}$ so that x^y is just $\sqrt{2}^2$ which simplifies to 2.

than completed infinities. The intuitionistic approach to mathematics has been studied and further developed by a number of mathematicians including Kolmogorov, Heyting, Bishop, Markov, Shanin, and many others [TvD88]. The use of constructive proof techniques has also had a significant influence on computer science, particularly through proof checking tools such as Nuprl [Con86]. Intuitionism does place serious and pervasive limitations on the methods of mathematics and has not yet had much of an impact on mathematical practice.

Formalism. Hilbert reacted to the intuitionistic critique by initiating a research programme to demonstrate that conventional mathematical methods were in fact valid. He wished to prove this using rigorous proof methods of mathematics that were acceptable beyond any shadow of doubt. His claim was that the use of infinities in mathematics was a convenient idealization and that any concrete conclusions about numbers that were drawn from such idealizations could be shown to be correct. Hilbert hoped that such a metamathematical study would show that mathematics was free of contradiction and hence *consistent*. In developing his metamathematics, he identified mathematics with a formal game played on paper with symbols and syntactic rules for forming statements and proofs. Hilbert also posed several other important problems in metamathematics including the question of whether a machine could correctly identify whether a given formal statement was a theorem in a given formal theory of mathematics.

Metamathematics has had a great many successes but Hilbert's original goal was not fulfilled. His own technique of formalizing and arithmetizing the syntax of mathematics led to Gödel's discovery that any reasonable formal theory contained sentences that could not be proved or disproved. As a consequence of this, Gödel showed that any consistency proof of a sufficiently powerful formal theory for mathematics would have to rely on methods that were stronger and more open to doubt than those permitted by the theory. Church [Chu36] and Turing [Tur65] showed that no machine could recognize the theorems in the predicate calculus. Since most formal mathematical theories build on the predicate calculus, the prospect of mechanical *decision procedures* for these theories is not great. There are interesting decidable theories such as Presburger arithmetic which is a form of Peano arithmetic where only addition is defined but not multiplication.

1.1.3 Language and Metalanguage

The quest for a foundation for mathematics begins with the search for a precise and unambiguous notation for expressing mathematics. Modern mathematical notation owes a good part of its precision and polish to the work of logicians such as Frege and Peano. A mathematical language provides a *syntax* for expressing statements about a particular conceptual domain. A *semantics* for a language characterizes those statements that are true of the conceptual domain. The language of *informal* mathematics consists of *ad hoc* notation and everyday language embellished with colloquial mathematical usages. The notation of logic underlies the formalized syntax of mathematics. For example, the axiom of *extensionality* in set theory would be written informally as, "Two sets are equal if they contain exactly the same elements." In formal notation, the same statement would be

$$(\forall x, y.\ (\forall z.\ z \in x \iff z \in y) \supset x = y).$$

Such a language can be given a precise grammar so that it is possible to automatically check an expression for syntactic well-formedness. We can characterize expressions according to their syntactic properties as formulas, atomic formulas, quantified formulas, sentences, and so on. It is possible then to study the connection between the syntactic and semantic properties of expressions. The language in which the syntax and semantics of a formalized language is discussed is called the *metalanguage*. The formal language itself is called the *object language*. Informal mathematical discourse often employs metalinguistic assertions such as, "There are exactly two free variables, 'x' and 'y', in the statement of extensionality above." Thus the metalanguage used in mathematical discourse is informal natural language such as English or German.

The work described in the present monograph employs a formal object language, namely that of first-order logic. In contrast to most logic textbooks where the metalanguage is informal, we employ a formal metalanguage that is based on the programming language pure Lisp (see Section 1.1.8).

1.1.4 Logic

Given a *language* for expressing statements about a particular conceptual domain, and a *semantics* for identifying the true statements, a *logic* is a system of *axioms* and *rules of inference* for constructing proofs of statements. Examples of logics include classical and intuitionistic propositional logics, temporal and modal logics, and first-order logic. A *theory* such as number theory or set theory can be formalized within the framework of a logic such as first-order logic by providing additional axioms. A statement is *provable* in a logic if it has a proof constructed from the axioms using the rules of inference. The provable statements are the *theorems* of the logic. A statement is *disprovable* if its negation is provable. A logic is *sound* if all provable statements are semantically true. It is *complete* if all of the semantically true statements are provable. The logic is *consistent* if no statement and its negation are both provable. A logic is *decidable* if there is an effective algorithm to determine whether a given statement is a theorem.

Propositional logic, for example, is about propositions, which as the dictionary indicates are expressions "in language or signs of something that can be believed, doubted, or denied or is either true or false." There are many ways that propositional logic can be presented; the presentation below follows Shoenfield [Sho67]. *Formulas* in the language consist of:

- the *propositional atoms* such as p, q, and r,
- *negations*: $(\neg A)$ (read as "not A"), where A is a formula, and
- *disjunctions*: $(A \vee B)$ (read as "A or B"), where A and B are themselves formulas.

Except for the quantified formulas to be introduced below, the surrounding parentheses will be dropped when they contribute nothing to the readability of the formula. Note that negation binds the tightest, and that conjunction and disjunction bind more tightly than implication and equivalence. The other propositional connectives can easily be defined in terms of negation and disjunction:

- *Implication*: $A \supset B \equiv \neg A \vee B$ (read as "A implies B").

A	$\neg A$
true	**false**
false	**true**

A	B	$A \vee B$
true	**true**	**true**
true	**false**	**true**
false	**true**	**true**
false	**false**	**false**

Figure 1.1: Truth Tables for Negation and Disjunction

$$\frac{}{A \vee \neg A}\ \textit{Axiom}$$

$$\frac{B}{A \vee B}\ \textit{Weakening}$$

$$\frac{A \vee A}{A}\ \textit{Contraction}$$

$$\frac{A \vee (B \vee C)}{(A \vee B) \vee C}\ \textit{Associativity}$$

$$\frac{A \vee B \quad \neg A \vee C}{B \vee C}\ \textit{Cut}$$

Figure 1.2: Proof Rules for Propositional Logic

- *Conjunction*: $A \wedge B \equiv \neg(\neg A \vee \neg B)$ (read as "A and B").
- *Equivalence*: $(A \iff B) \equiv (A \supset B) \wedge (B \supset A)$ (read as "A equivales B" or "A if and only if B").

The classical semantics for propositional formulas is given by assigning *truth values*, **true** or **false**, to the propositional atoms and evaluating the statements for each such assignment using the *truth-table interpretation* of negation and disjunction. These truth tables are displayed in Figure 1.1. The statements of propositional logic are just its formulas. The true statements are those that evaluate to **true** under any assignment of truth values to the propositional atoms. A typical proof rule of the propositional logic, the *law of the excluded middle*, asserts that any formula of the form $A \vee \neg A$ is an *axiom*. Such a proof rule is an axiom scheme since it asserts $A \vee \neg A$ to be an axiom for any formula A. Any formula of the form $A \vee \neg A$ always evaluates to **true** regardless of whether A is assigned **true** or **false**.

The excluded middle axiom and the other proof rules of propositional logic are shown in Figure 1.2. For example, the proof rule of *weakening* would yield a proof of $(A \vee B)$ from a proof of B. Propositional logic can easily be shown to be sound relative to the truth-table interpretation, and also to be consistent and complete.[7]

Furthermore, propositional logic is *decidable*: there is a straightforward algorithm to decide if a given statement is a theorem by simply checking if every truth-table evaluation of the statement is **true**. The consistency, completeness, and decidability of propositional logic

[7]Soundness typically implies consistency. A sound theory is consistent if the semantics does not judge some statement and its negation to both be true. A theory could be consistent in not proving some statement and its negation but could be unsound relative to the intended semantics.

constitute the earliest significant contributions to metamathematics and were first proved in Post's doctoral dissertation [Pos21], and independently by Bernays [Ber26].

First-order logic or the predicate calculus is a refinement of propositional logic where the truth values of propositional atoms are allowed to vary depending on the values assigned to the *individual variables.* The syntax of first-order logic contains individual variables, function symbols (e.g., the addition and multiplication operations of arithmetic), predicate symbols (e.g., the symbols for the ordering relations $<$ and $\leq$ on numbers), and quantifiers. First-order *terms* are either

- individual variables denoted by x, y, and z, or
- of the form $f(t_1, \ldots, t_n)$, where f is an n-ary *function symbol* and the t_i are terms.

An *atomic formula* has the form $p(t_1, \ldots, t_n)$, where p is an n-ary predicate symbol and the t_i are terms. The arities of predicates and function symbols can be zero. The *formulas* include

- atomic formulas,
- the *negation* $\neg A$ of any formula A,
- the *disjunction* $A \vee B$ of any two formulas A and B, and
- the *existential quantification* $(\exists x.\ A)$ (read as "for some x, A") of any formula A with respect to a variable x.

The form $(\forall x.\ A)$ (read as "forall x, A) can be defined as $\neg(\exists x.\ \neg A)$. Unlike propositional logic, where the meaning of an atomic proposition in a given interpretation is either **true** or **false**, the meaning of $p(t_1, \ldots, t_n)$ varies according to the meanings of the t_i. A *first-order language* is a collection of function and predicate symbols obeying the above syntactic rules.

Any expression (term or formula) is a *subexpression* of itself; the subexpressions of $f(t_1, \ldots, t_n)$ and $p(t_1, \ldots, t_n)$ also include the subexpressions of the t_i, the subexpressions of $\neg A$ and $(\exists x.\ A)$ include those of A, and the subexpressions of $A \vee B$ include those of A and B. A formula A is a *subformula* of a another formula B if A is a subexpression of B. A variable x is said to *occur* in a term or formula in which it occurs as a subexpression. Any occurrence of x in A is *bound* in $(\exists x.\ A)$. An occurrence of a variable is *free* in a formula if it is not bound in any subformula of the given formula. Any variable that has a free occurrence in a formula is a free variable of the formula. A *statement* or a *sentence* is a formula with no free variables.[8] The notations $[r/x]t$ and $[r/x]A$ refer to the results of substituting the term r for all free occurrences of the variable x in the term t and the formula A, respectively. First-order logic *with equality* contains a special 2-ary predicate symbol for equality that is used in the infix form $a = b$, where a and b are terms.

A *model* for a first-order language assigns meanings to the function and predicate symbols relative to a nonempty domain D. A model $\mathcal{M}$ associates each n-ary function symbol in the language to a mapping from D^n to D, and each n-ary predicate symbol in the language to

[8] Any formula can be seen as a statement where the free variables in the formula are implicitly universally quantified at the outermost level. So if A is a formula in a proof and x is the only free variable in A, then the statement A is read implicitly as $(\forall x.\ A)$.

$$
\begin{aligned}
\mathcal{M}[\![x]\!]\rho &= \rho(x)\\
\mathcal{M}[\![f(t_1,\ldots,t_n)]\!]\rho &= \mathcal{M}(f)(\mathcal{M}[\![t_1]\!]\rho,\ldots,\mathcal{M}[\![t_n]\!]\rho)\\
\mathcal{M}[\![a=b]\!]\rho &= \begin{cases}\textbf{true}, \text{ if } \mathcal{M}[\![a]\!]\rho = \mathcal{M}[\![b]\!]\rho\\ \textbf{false}, \text{ otherwise}\end{cases}\\
\mathcal{M}[\![p(t_1,\ldots,t_n)]\!]\rho &= \mathcal{M}(p)(\mathcal{M}[\![t_1]\!]\rho,\ldots,\mathcal{M}[\![t_n]\!]\rho)\\
\mathcal{M}[\![\neg A]\!]\rho &= \begin{cases}\textbf{false}, \text{ if } \mathcal{M}[\![A]\!]\rho = \textbf{true}\\ \textbf{true}, \text{ otherwise}\end{cases}\\
\mathcal{M}[\![A \vee B]\!]\rho &= \begin{cases}\textbf{true}, \text{ if } \mathcal{M}[\![A]\!]\rho = \textbf{true}\\ \textbf{true}, \text{ if } \mathcal{M}[\![B]\!]\rho = \textbf{true}\\ \textbf{false}, \text{ otherwise}\end{cases}\\
\mathcal{M}[\![(\exists x.\ A)]\!]\rho &= \begin{cases}\textbf{true}, \text{ if } \mathcal{M}[\![A]\!][d/x]\rho = \textbf{true}, \text{ for some } d \text{ in } D\\ \textbf{false}, \text{ otherwise}\end{cases}
\end{aligned}
$$

Figure 1.3: Semantics of First-Order Logic

$$\frac{}{[a/x]A \supset (\exists x.\ A)}\ \textit{Substitution}$$

$$\frac{}{a = a}\ \textit{Identity}$$

$$\frac{}{a_1 = b_1 \supset (\ldots \supset (a_n = b_n \supset f(a_1,\ldots,a_n) = f(b_1,\ldots,b_n)))}\ \textit{Equality}_1$$

$$\frac{}{a_1 = b_1 \supset (\ldots \supset (a_n = b_n \supset p(a_1,\ldots,a_n) \supset p(b_1,\ldots,b_n)))}\ \textit{Equality}_2$$

$$\{x \text{ is not free in } B\}\frac{A \supset B}{(\exists x.\ A) \supset B}\ \exists\textit{-Introduction}$$

Figure 1.4: (Non-Propositional) Proof Rules for First-Order Logic

a mapping from D^n to the truth values $\{\textbf{true}, \textbf{false}\}$. A model by itself is not enough to determine the meanings of expressions since we also need to fix the interpretation of any free variables in the expression. An *assignment* ρ assigns values in D to the individual variables of the language. A formula A is true in a model if it is true with respect to any assignment ρ of values in D to the free variables in A. If ρ is an assignment, then $[d/x]\rho$ is the assignment that assigns d to x and $\rho(y)$ to any other variable y. The meaning of a formula in a model $\mathcal{M}$ under an assignment ρ is shown in Figure 1.3. The formula $(\exists x.\ A)$ is true in the model under assignment ρ if the formula A is true in the model under the assignment $[d/x]\rho$, for some value d in D. The truth value of a formula with free variables depends on the choice of the assignment, but a statement can only be either true or false in a model. A statement is semantically true if it is true in all models.

In addition to the proof rules of propositional logic (in Figure 1.2), first-order logic contains rules pertaining to equality and quantification. These rules are shown in Figure 1.4. It is