One standard list of logical axioms and rules of inference for first-order logic is provided in Appendix A. One appealing feature of this version is that there is only one rule of inference, modus ponens.

Definition. Let T be a set of first-order formulas. A **proof** from (proper axioms) T is a finite sequence of formulas ("steps") such that every step is either a logical axiom, a member of T, or the result of applying a rule of inference to previous steps in the proof. A proof of a formula P from T is a proof from T whose last step is P.

More precisely, this is the definition of a **formal proof**. Of course, mathematicians never actually write formal proofs; their proofs are informal in many ways. For one thing, proofs normally are written at least partly in a natural language such as English, rather than a first-order language. Furthermore, they include all sorts of shortcuts, such as definitions, and citing previously proved statements rather than reproving them. Still, most logicians (though perhaps not most mathematicians) are convinced that all correct proofs in mathematics could, with enough effort, be translated into formal proofs of first-order logic.

A **first-order theory** technically consists of two parts: a first-order language, and a set of formulas (usually, sentences) of that language to be used as proper axioms. There is no need to specify the logical axioms or the rules of inference since they are standard, except for inconsequential variations. For the most part, we will use the word **theory** a bit more loosely to mean any set of formulas T in a first-order language.

Notation. We write $T \vdash \mathrm{P}$ ("T **proves** P") to mean that there is a proof of P from T. There are several other ways to read this notation: P is **provable** from (or in) T, P is **deducible** from T, P is **derivable** from T, P is a **theorem** of T, or P is a **logical consequence** of T.

There are also numerous variants of this notation:

$T_1 \vdash T_2$ means that $T_1 \vdash \mathrm{P}$ for every $\mathrm{P} \in T_2$.

$\mathrm{Q} \vdash P$ means $\{\mathrm{Q}\} \vdash \mathrm{P}$.

$\vdash \mathrm{P}$ means $\emptyset \vdash \mathrm{P}$, that is, P is derivable from logical axioms and rules of inference alone. A formula with this property is called a **law of logic**.

Finally, P and Q are called **logically equivalent** if $\vdash (\mathrm{P} \leftrightarrow \mathrm{Q})$.

Notation. The set of theorems of T will be denoted $Thm(T)$.

Definition. If $Thm(T_1) \subseteq Thm(T_2)$ (which is the same as saying $T_2 \vdash T_1$), we say that T_1 is a **subtheory** of T_2, and T_2 is an **extension** of T_1. If $Thm(T_1) = Thm(T_2)$, we say that T_1 and T_2 are **equivalent** theories.

The notions of "logical consequence," "law of logic," and "logical equivalence" are analogous to the corresponding propositional notions defined in Section 1.2. The important qualitative difference is that the propositional versions are all based on truth tables and therefore are **computable** or **decidable**—a finite amount of straightforward computation always suffices to determine whether or not they hold, provided that the sets of formulas involved are finite. This is not the case for the concepts that we just defined. There is no "effective procedure" (essentially, a computer program) that can even determine whether or not any given first-order sentence is a law of logic. We should perhaps be thankful for this, since if there were such a computer program, humans would hardly ever (if ever) be needed for proving theorems! (Computability and decidability will be thoroughly discussed in Chapter 3.)

Even though the *notions* of "logical consequence," "law of logic," and "logical equivalence" are analogous to notions defined in Section 1.2, the *definitions* themselves are not analogous. The definitions in Section 1.2 are **semantic**, meaning that they are based on some concept of truth—in this case, truth tables. A law of propositional logic is a statement that is true under all possible interpretations of certain substatements. It is very natural to use semantic definitions for propositional logic because truth tables are so simple to understand and use.

By contrast, the definitions we have just given for first-order logic are **syntactic**—they are based on provability in some formal system. We can obtain syntactic definitions of the corresponding propositional notions simply by removing those parts of Appendix A that mention quantifiers. It is also fruitful to give semantic definitions of these notions for first-order logic; again, these are based on the idea that a law of logic is a statement that is true under all possible interpretations. However, it requires some "machinery" to make this precise so we will defer these definitions until Chapter 5.

Are the semantic and syntactic definitions equivalent? Yes they are, which is reassuring. This tells us that a formula P is provable from a theory T if and only if P must be true whenever all the statements in T are true. A detailed discussion of this equivalence in the case of first-order logic will also be given in Chapter 5. This equivalence provides a compelling argument that the formalists succeeded in codifying, precisely and compactly, the 2300-year-old notion of a correct mathematical deduction.

We will occasionally refer to reasoning "informally." Usually, this will mean using the fact that provability corresponds to truth, in order to avoid a tedious formal proof. For instance, it can certainly be proved formally that $\forall x\, \forall y \mathrm{P}(x, y)$ is logically equivalent to $\forall y\, \forall x \mathrm{P}(x, y)$. But the obvious truth of this equivalence may be considered a nonrigorous proof, acceptable in most circumstances.

Definitions. A theory T is called **consistent** if no contradiction can be derived from it. A formula P is said to be **independent** of T if neither P nor $\sim$ P can be proved from T. T is called **complete** if it is consistent and no *sentence* of its language is independent of it. In other words, T is complete if $T \vdash \mathrm{P}$ or $T \vdash\sim \mathrm{P}$, but not both, for every sentence P of the language of T. (The language of T is the smallest first-order language that contains T.)

The subject that deals with first-order languages and theories is called **first-order predicate logic**, or simply **first-order logic**. We conclude this section by stating two of the most important **metatheorems** of first-order logic, that is, theorems *about* first-order logic. Their proofs can be found in most logic texts such as [End] and [Sho]:

Theorem 1.3 (Deduction Theorem).

$$\textit{If } T \cup \{\mathrm{P}\} \vdash \mathrm{Q}, \ \textit{then } T \vdash (\mathrm{P} \to \mathrm{Q}).$$

The converse of this theorem also holds; essentially, it is the rule of inference modus ponens. The deduction theorem is the formal justification of the method of conditional proof (direct proof of implications). Similarly, the next result is the formal justification of the method of universal generalization:

Theorem 1.4 (Generalization Theorem). *If $T \vdash \mathrm{P}(x)$ and the variable x does not occur free in any formula in T, then $T \vdash \forall x \mathrm{P}(x)$.*

Exercise 7. Let T be a theory and P a sentence. Prove:

(a) If T is inconsistent, then every formula is derivable from T.

(b) $T \vdash \mathrm{P}$ if and only if $T \cup \{\sim \mathrm{P}\}$ is inconsistent.

(c) P is independent of T if and only if $T \cup \{\text{P}\}$ and $T \cup \{\sim \text{P}\}$ are both consistent.

1.5 Examples of first-order theories

In the previous section it was claimed that first-order languages form an adequate framework for the translation of mathematical statements into a purely symbolic form. We will now illustrate, by means of several examples, precisely what we mean by this claim. As we will see, the claim is not completely unproblematic.

Example 16 (Peano Arithmetic). One of the first successful formalizations of a part of mathematics was the axiomatization of arithmetic, first carried out by Richard Dedekind and then refined by Peano in the 1890s. The intended domain of this theory is the set $\mathbb{N}$, but one can use the theory to define and study $\mathbb{Z}$ and $\mathbb{Q}$ (the set of rational numbers) as well. The most common first-order language $\mathcal{L}$ used for this theory has two binary function symbols, $+$ and $\cdot$, a unary function symbol S ("successor"), and a constant symbol $\overline{0}$. (We write $\overline{0}$ rather than 0 to emphasize that this is a formal symbol, not the number 0. However, most logic books—including this one—are not consistently careful to make this type of distinction.) As usual, the operators $+$ and $\cdot$ are written between their arguments. The proper axioms of Peano arithmetic, PA for short, include the following straightforward ones:

1. $S(x) \neq \overline{0}$. ($\overline{0}$ is not any number's successor.)
2. $S(x) = S(y) \rightarrow x = y$. ($S$ is one-to-one.)
3. $x + \overline{0} = x$.
4. $x + S(y) = S(x + y)$.
5. $x \cdot \overline{0} = \overline{0}$.
6. $x \cdot S(y) = (x \cdot y) + x$.

In addition to the above, PA also needs some sort of principle of mathematical induction. The most straightforward statement of induction is

7. $[\bar{0} \in A \ \wedge \ \forall n(n \in A \rightarrow \ S(n) \in A)] \rightarrow \ \forall n(n \in A)$.

The intention here is that the variable n ranges over natural numbers, while the variable A ranges over *sets* of natural numbers. But $\mathcal{L}$ does not have variables for sets. One obvious solution to this difficulty is to expand $\mathcal{L}$ to a two-sorted language with natural number variables and set variables, and add the binary relation symbol $\in$ to $\mathcal{L}$.

But this type of two-sorted first-order language, with variables for elements and for subsets of an intended domain, does not provide the intended meaning unless the domain of the set variables consists of all subsets of the domain of the element variables. As we will see in Chapter 5, the rules for interpreting first-order theories do not require this. If we want this version of induction to mean what it ought to mean, we need to go beyond first-order logic and instead use a **second-order** version of formal arithmetic. Second-order logic, and the reasons for using it in this type of situation, will be discussed in the next section.

In order to complete the axiomatization of *first-order* PA, we must replace the concise form of induction given above with the so-called **predicate form**: for each $\mathcal{L}$-formula $\mathrm{P}(n)$ with the free variable n (and possibly other free variables), we include the axiom

7′. $[\mathrm{P}(\bar{0}) \wedge \forall n(\mathrm{P}(n) \rightarrow \mathrm{P}(S(n)))] \rightarrow \ \forall n \mathrm{P}(n)$.

This works well for many purposes, but it does have two drawbacks. One is that the single induction axiom has been replaced by an infinite list of axioms (a so-called **axiom schema**), a situation that cannot be avoided if we stay in the language $\mathcal{L}$. That is, there is no finite set of axioms that is equivalent (in terms of the theorems obtained) to axioms (1) through (6) above plus the schema (7′). We express this limitation by saying that first-order PA is not **finitely axiomatizable**. (Because P is a propositional variable, rather than a mathematical variable within $\mathcal{L}$, it cannot be quantified in first-order PA.)

The second, more serious, drawback is that this axiom schema might have less "power" than the version involving sets. To see this, note that our axiom schema of induction may be viewed as the set version restricted to sets that are **definable** by a formula of $\mathcal{L}$, that is, sets of the form $\{n : \mathrm{P}(n)\}$. The set of formulas of $\mathcal{L}$ is countable, by Proposition 1(e) of Appendix C. Therefore, since there are uncountably many subsets of $\mathbb{N}$, it is quite plausible that the predicate form of induction could be inadequate for proving some important theorems.

In summary, we have the following situation, which turns out to be quite common: there is a sensible first-order theory that seems to provide a correct formalization of arithmetic. However, this first-order theory does not have as much theorem-proving power as one would like, especially for more advanced purposes, and this limitation cannot be remedied in the original first-order language. At the same time, those who want to maintain that all of mathematics can be carried out within first-order logic need not admit defeat: they can specify ZFC set theory (see Example 18 below) as the first-order theory to be used for all of mathematics.

From now on, unless stated otherwise, the terms "Peano arithmetic" and "PA" refer to *first-order* Peano arithmetic. The discussion above emphasizes the limitations of PA, but in fact a surprising amount of mathematics can be carried out in subtheories of PA in which the induction axiom schema is severely restricted, instead of being allowed for all $\mathcal{L}$-formulas $\mathrm{P}(n)$. The investigation of these "weak" subsystems of PA has proven to be a very fruitful area of research. Section 4.4 will provide a more detailed treatment of what can and cannot be proved in PA.

Exercise 8. To get a feel for PA, you might want to prove a few basic arithmetical facts from its axioms (not too formally, or you'll drive yourself crazy!). Reasonable choices might be the commutative laws of addition and multiplication and the distributive laws. Also, you might want to show how to define the predicate $m < n$ within PA, and then prove irreflexivity, transitivity, etc. Almost all of these proofs require induction (on just one variable, even if there is more than one variable in the statement).

Example 17 (The First-Order Theory of Rings and Fields). Let $\mathcal{L}$ be the first-order language of a ring, as described in Example 15. In the context of ring theory, it is not necessary to have the symbols 0 and $-$ in the language, because they are definable. However, for most purposes it is more convenient to include these symbols in $\mathcal{L}$.

In $\mathcal{L}$, it is simple to write down the usual axioms of a ring, a commutative ring, a ring with unity, a field, etc. (See Appendix D for basics.) In this context, the axioms are just the defining properties of these algebraic structures, rather than basic, assumed truths. Many simple theorems of ring and field theory can be stated in this language and proved from the appropriate axioms: the uniqueness of identity elements and inverses, the fact that any number times 0 equals 0, the fact that a field has no zero-divisors, etc. On the other hand, it is easy to see that $\mathcal{L}$ is not adequate for a full treatment of rings and fields. Among other things, it provides no way of discussing arbitrary subrings or subsets of a ring, or mappings between rings.

Furthermore, the language $\mathcal{L}$ lacks the means to express some rather basic facts about a single ring or field. For example, consider the idea of characteristic of a field. If we want to state in $\mathcal{L}$ that the characteristic of a field is 3, it is easy to do so: $1 + 1 + 1 = 0$. (As usual, the associative law enables us to omit parentheses on the left side of this equation.)

Exercise 9. How would we state this property of a field in $\mathcal{L}$ if there were no symbols for the identity elements?

But now suppose we want to axiomatize the theory of a field of characteristic zero. If you look in a standard abstract algebra text, the definition given for this is something like "There is no positive integer n such that $n \cdot 1 = 0$, where $n \cdot 1$ is an abbreviation for $1 + 1 + \cdots + 1$ (n times)." It's tempting to think that this can be formalized in $\mathcal{L}$ as $\forall n > 0(n \cdot 1 \neq 0)$. But the problem is that the variable n in this formula denotes a natural number, not a member of the field in question, so this formula is not within the language $\mathcal{L}$.

So how can we formalize, in $\mathcal{L}$, that a field has characteristic zero? The standard way is to use an axiom schema, as in Example 16: start with the usual field axioms and add, for each n, the formula $1 + 1 + \cdots + 1 \neq 0$, where there are n 1's on the left side of the equation. We will prove in Chapter 5 that there is no finite set of axioms of $\mathcal{L}$ that is equivalent to this infinite list of axioms. In other words, the first-order theory of a field of characteristic zero, like first-order Peano arithmetic, is not finitely axiomatizable. Furthermore, there is absolutely no way in $\mathcal{L}$, even using an infinite set of axioms, to express that a field has *finite* (that is, nonzero) characteristic! We will also see that these limitations are not just an esoteric curiosity; they lead to some questions that are of genuine interest to algebraists.

If we want a theory in which we can work with concepts such as finite characteristic, subrings, and homomorphisms, we need to go beyond the first-order theory of rings and fields. As in Example 16, we could use a second-order theory, or we could use the full power of set theory.

Example 18 (Zermelo–Fraenkel Set Theory). One of the most important mathematical achievements of the early part of the twentieth century was the development of versions of set theory that apparently avoid the paradoxes of "naive" set theory (to be discussed in Chapter 2), and yet do not significantly diminish the freedom to define abstract and infinite sets, as envisioned by the founders of set theory. The most important of these theories is called Zermelo–Fraenkel (ZF) set theory; with the addition of the axiom of choice, it is simply called ZFC set theory.

ZFC is a remarkable first-order theory. All of the results of contemporary mathematics can be expressed and proved within ZFC, with at most a handful of esoteric exceptions. Thus it provides the main support for the formalist position regarding the formalizability of mathematics. In fact, logicians tend to think of ZFC and mathematics as practically synonymous.

On the other hand, ZFC is in many ways an extremely simple theory. This is especially true of its language. The language of set theory has just one binary relation symbol $\in$. It is not even necessary to include the equality symbol, since equality of sets can be defined (two sets being equal if and only if they have exactly the same elements). It is worth noting that ZFC is a "pure" set theory: all the objects under discussion are technically sets. There are not even variables or axioms for the natural numbers; every mathematical object must be a set. We will examine ZFC in much more detail in Chapters 2 and 6.

Example 19 (A First-Order Theory of Family Relationships). We conclude this section with a nonmathematical example. But this example provides a good setting to practice translating English statements into a formal symbolic language, with careful use of quantifiers.

Let $\mathcal{L}$ be a first-order language with one unary relation symbol $\mathrm{W}(x)$ and one binary relation symbol $\mathrm{P}(x, y)$, in addition to equality. The intended interpretation is that the variables of $\mathcal{L}$ denote people (living or dead), $\mathrm{W}(x)$ means "x is female," and $\mathrm{P}(x, y)$ means "x is a parent of y." (There is no requirement to mention any interpretation when defining a first-order language or theory, but doing so is often very helpful to the reader.) All of the usual blood relationships for which we have words can be expressed in this language. For example, the statements that one person is another's mother, grandparent, uncle, brother, half-sibling, or first cousin can all be formalized in $\mathcal{L}$.

On the other hand, not everything one might want to say about family relationships can be expressed in $\mathcal{L}$. If you give it some thought, you should be able to convince yourself that there's no way to express "x is a descendant of y" or "x and y are blood relatives" in $\mathcal{L}$. This limitation is rather similar to the situation we discussed in first-order field theory, where certain statements regarding the characteristic of a field could not be formalized. As an exercise, see if you can expand $\mathcal{L}$ in a way that makes it possible to say anything about blood relationships that can be expressed in English. Would it suffice to add integer variables to the language, and a relation symbol $\mathrm{D}(x, y, n)$ that means "y is an nth level descendant of x"? (So $\mathrm{D}(x, y, 1)$ would mean y is x's child, etc.) Is it possible to express "x and y are blood relatives" in this expanded language?

So we have a couple of possibilities for a first-order language for the formalization of family relationships. What about axioms? In the original language with only "people variables," the most obvious axioms to include are that each person has a unique mother and father. From this, we can prove many simple facts, for example, that each person has exactly four grandparents (if there's no incest!), and that sib-linghood is an equivalence relation (if we agree that each person is his or her own sibling).

But this simple axiomatization is inadequate if we want to prevent circular relationships, such as someone being his or her own parent or grandparent. What kind of axioms would prevent these? Surprisingly, there is no finite set of axioms in $\mathcal{L}$ that will do so. If we go to an expanded language as above, or a language in which we can refer to a person's time of birth, and assert (as an axiom) that every person is born after his or her parents were, then we can guarantee noncircularity. Essentially, it would be desirable to be able to express the "descendant" (or "ancestor") binary relation in whatever formal language we choose, and then have axioms that state that these relations are partial orderings. You might find it instructive to carry this out in some detail.