Cambridge University Press & Assessment
978-1-108-83314-1 — Modern Mathematical Logic
Joseph Mileti
Excerpt
More Information

1 Introduction

1.1 The Nature of Mathematical Logic

Mathematical logic originated as an attempt to codify and formalize the following:

- The language of mathematics.
- The basic assumptions of mathematics.
- The permissible rules of proof.

One of the successful results of such a program is the ability to study mathematical language and reasoning using mathematics itself. For example, we will eventually give a precise mathematical definition of a formal proof, and to avoid confusion with our current intuitive understanding of what a proof is, we will call these objects *deductions*. One can think of our eventual definition of a deduction as analogous to the precise mathematical definition of continuity, which replaces the fuzzy "a graph that can be drawn without lifting your pencil." Once we have codified the notion in this way, we will have turned deductions into precise mathematical objects, allowing us to prove mathematical theorems about deductions using normal mathematical reasoning. For example, we will open up the possibility of proving that there is no deduction of certain mathematical statements.

Some newcomers to mathematical logic find the whole enterprise perplexing. For instance, if you come to the subject with the belief that the role of mathematical logic is to serve as a foundation to make mathematics more precise and secure, then the preceding description probably sounds rather circular, and this will almost certainly lead to a great deal of confusion. You may ask yourself:

> Okay, we have just given a decent definition of a deduction. However, instead of proving things about deductions following this formal definition, we're proving things about deductions using the usual informal proof style that I've grown accustomed to in other math courses. Why should I trust these informal proofs about deductions? How can we formally prove things (using deductions) about deductions? Isn't that circular? Is that why we are only giving informal proofs? I thought that I would come away from this subject feeling better about the philosophical foundations of mathematics, but we have just added a new layer to mathematics, so we now have both informal proofs and deductions, which makes the whole thing even more dubious.

Other newcomers do not see a problem. After all, mathematics is the most reliable method we have to establish truth, and there was never any serious doubt about its validity. Such a person might react to these thoughts as follows:

> We gave a mathematical definition of a deduction, so what's wrong with using mathematics to prove things about deductions? There's obviously a "real world" of true mathematics, and we are just working in that world to build a certain model of mathematical reasoning that is susceptible to mathematical analysis. It's quite cool, really, that we can subject mathematical proofs to a mathematical study by building this internal model. All of this philosophical speculation and worry about secure foundations is tiresome, and probably meaningless. Let's get on with the subject!

Should we be so dismissive of the first, philosophically inclined, student? The answer, of course, depends on your own philosophical views, but I will give my views as a mathematician specializing in logic with a definite interest in foundational questions. It is my firm belief that you should put all philosophical questions out of your mind during a first reading of the material (and perhaps forever, if you are so inclined), and come to the subject with a point of view that accepts an independent mathematical reality susceptible to the usual mathematical methods. In your mind, you should keep a careful distinction between normal "real" mathematical reasoning and the formal precise model of mathematical reasoning we are developing. It is useful to separate these two realms by giving one a name, so one often calls the normal mathematical realm (that we will be working within) the *metatheory*. See Section 8.2 for a much more robust discussion of these ideas.

We will eventually give examples of formal theories, such as first-order set theory, which are able to support the entire enterprise of mathematics, including mathematical logic itself. Once we have developed set theory in this way, we will be able to give reasonable answers to the first student and provide other respectable philosophical accounts of the nature of mathematics. Again, see Section 8.2.

The ideas and techniques that were developed with philosophical goals in mind have now found application in other branches of mathematics and in computer science. The subject, like all mature areas of mathematics, has also developed its own very interesting internal questions, which are often (for better or worse) divorced from its roots. Most of the subject developed after the 1930s is concerned with these internal and tangential questions, along with applications to other areas, and now foundational work is just one small (but still important) part of mathematical logic. Thus, even if you have no interest in the more philosophical aspects of the subject, there remains an impressive, beautiful, and mathematically applicable theory that is worthy of your attention.

1.2 The Language of Mathematics

The first, and probably most essential, issue that we must address in order to provide a formal model of mathematics is how to deal with the language of mathematics. In this section, we sketch the basic ideas and motivation for the development of a language, but we will leave precise detailed definitions until later.

We should certainly not use English (or any other natural language) because it is constantly changing, often ambiguous, and allows the construction of statements that are not mathematical, or that express very subjective sentiments. Once we have thrown out natural language, our only choice is to invent our own formal language. At first, the idea of developing a universal language seems quite daunting. How could we possibly write down one formal language that can simultaneously express the ideas in geometry, algebra, analysis, and every other field of mathematics, not to mention those we haven't developed yet? Our approach to this problem will be to avoid (consciously) doing it all at once.

Instead of starting from the bottom and trying to define primitive mathematical statements that cannot be broken down further, let's first think about how to build new mathematical statements from old ones. The simplest way to do this is to take already established mathematical statements and put them together using *and*, *or*, *not*, and *implies*. To keep a careful distinction between English and our language (as well as between our metatheory and our formal model), we will introduce symbols for each of these, and we will call these symbols *connectives*.

- $\wedge$, which is intended to mean *and*.
- $\vee$, which is intended to mean *or*.
- $\neg$, which is intended to mean *not*.
- $\rightarrow$, which is intended to mean *implies*.

In order to ignore the nagging question of what constitutes a primitive statement, our first attempt will simply be to take an arbitrary set whose elements we think of as the primitive statements, and put them together in all possible ways using the connectives.

For example, suppose we start with the set $P = \{\mathtt{A}, \mathtt{B}, \mathtt{C}\}$. We think of $\mathtt{A}$, $\mathtt{B}$, and $\mathtt{C}$ as our primitive statements, and we may or may not care what they might express. We now want to put together the elements of P using the connectives, perhaps repeatedly. However, this naive idea quickly leads to a problem. Should the *meaning* of $\mathtt{A} \wedge \mathtt{B} \vee \mathtt{C}$ be "$\mathtt{A}$ holds, and either $\mathtt{B}$ holds or $\mathtt{C}$ holds," corresponding to $\mathtt{A} \wedge (\mathtt{B} \vee \mathtt{C})$, or should it be "Either both $\mathtt{A}$ and $\mathtt{B}$ hold, or $\mathtt{C}$ holds," corresponding to $(\mathtt{A} \wedge \mathtt{B}) \vee \mathtt{C}$? We need some way to avoid this ambiguity. Probably the most natural way to achieve this is to insert parentheses to make it clear how to group terms (we will eventually see other natural ways to overcome this issue). We now describe the collection of *formulas* of our language. We start by saying that every element of P is a formula, and then we generate other formulas using the following rules:

- If φ and ψ are formulas, then $(\varphi \wedge \psi)$ is a formula.
- If φ and ψ are formulas, then $(\varphi \vee \psi)$ is a formula.
- If φ is a formula, then $(\neg\varphi)$ is a formula.
- If φ and ψ are formulas, then $(\varphi \rightarrow \psi)$ is a formula.

For example, $((\neg(\mathtt{B} \vee ((\neg\mathtt{A}) \rightarrow \mathtt{C}))) \vee \mathtt{A})$ is a formula. This simple setup, called *propositional logic*, is a drastic simplification of the language of mathematics, but there are already many interesting questions and theorems that arise from a careful study. We will spend some time on it in Chapter 3.

Of course, mathematical language is much more rich and varied than what we can obtain using propositional logic. One important way to make more complicated and interesting mathematical statements is to make use of the quantifiers *for all* and *there exists*. We introduce the symbols $\forall$ and $\exists$, which are intended to mean "for all" and "there exists," respectively. In order to use these symbols, we will need *variables* to act as something to quantify over. We will denote variables by letters like $\mathtt{x}$, $\mathtt{y}$, $\mathtt{z}$, and so on. Once we have come this far, however, we will have to refine our naive notion of primitive statements, because it is unclear how to interpret a statement like $\forall \mathtt{x} \mathtt{B}$ without knowledge of the role of $\mathtt{x}$ "inside" $\mathtt{B}$.

Let's think a little about our primitive statements. As we have mentioned, it seems daunting to come up with primitive statements for all areas of mathematics at once, so let's think of the various areas in isolation. For instance, take group theory. A group is a set G equipped with a binary operation $\cdot$ (that is, $\cdot$ takes in two elements $x, y \in G$ and produces a new element of G denoted by $x \cdot y$) and an element e satisfying the following:

- Associativity: For all $x, y, z \in G$, we have $(x \cdot y) \cdot z = x \cdot (y \cdot z)$.
- Identity: For all $x \in G$, we have $x \cdot e = x = e \cdot x$.
- Inverses: For all $x \in G$, there exists $y \in G$ such that $x \cdot y = e = y \cdot x$.

Although it is customary, and certainly easier on the eyes, to put $\cdot$ between two elements of the group, let's instead use the standard function notation in order to make the mathematical notation uniform across different areas. In this setting, a group is a set G equipped with a function $f \colon G \times G \to G$ and an element e satisfying the following:

- For all $x, y, z \in G$, we have $f(f(x, y), z) = f(x, f(y, z))$.
- For all $x \in G$, we have $f(x, e) = x = f(e, x)$.
- For all $x \in G$, there exists $y \in G$ such that $f(x, y) = e = f(y, x)$.

In order to allow our language to make statements about groups, we introduce a *function symbol* $\mathtt{f}$ to represent the group operation, and a *constant symbol* $\mathtt{e}$ to represent the group identity. Now the group operation is supposed to take in two elements of the group, so if $\mathtt{x}$ and $\mathtt{y}$ are variables, then we should allow the formation of $\mathtt{f}(\mathtt{x}, \mathtt{y})$, which should denote an element of the group (once we have assigned elements of the group to $\mathtt{x}$ and $\mathtt{y}$). Also, we should permit the constant symbol to be used in this way, allowing us to form things like $\mathtt{f}(\mathtt{x}, \mathtt{e})$. Once we have formed such expressions, we should be allowed to use them like variables in more complicated expressions, such as $\mathtt{f}(\mathtt{f}(\mathtt{x}, \mathtt{e}), \mathtt{y})$. Each of these expressions formed by putting together, perhaps repeatedly, variables and the constant symbol $\mathtt{e}$ using the function symbol $\mathtt{f}$ is called a *term*. Intuitively, a term will name a certain element of the group once we have assigned elements to the variables.

With a way to name group elements in hand, we are now in position to describe our primitive statements. The most basic thing that we can say about two group elements is whether or not they are equal, so we introduce a new *equality symbol*, which we will denote by the customary $=$. Given two terms t_1 and t_2, we call the expression $(t_1 = t_2)$ an *atomic formula*. These are our primitive statements.

With atomic formulas in hand, we can use the old connectives and the new quantifiers to make (potentially complex) statements, which we again call *formulas*. First, all atomic formulas are formulas. Given formulas we already know, we can put them together using the connectives above. Also, if φ is a formula and $\mathtt{x}$ is a variable, then each of the following is a formula:

- $\forall \mathtt{x} \varphi$.
- $\exists \mathtt{x} \varphi$.

Perhaps without realizing it, we have described a reasonably powerful language capable of making many nontrivial statements. For instance, we can write formulas in this language that express the axioms for a group:

- $\forall \mathtt{x} \forall \mathtt{y} \forall \mathtt{z}(\mathtt{f}(\mathtt{f}(\mathtt{x}, \mathtt{y}), \mathtt{z}) = \mathtt{f}(\mathtt{x}, \mathtt{f}(\mathtt{y}, \mathtt{z})))$.
- $\forall \mathtt{x}((\mathtt{f}(\mathtt{x}, \mathtt{e}) = \mathtt{x}) \wedge (\mathtt{f}(\mathtt{e}, \mathtt{x}) = \mathtt{x}))$.
- $\forall \mathtt{x} \exists \mathtt{y}((\mathtt{f}(\mathtt{x}, \mathtt{y}) = \mathtt{e}) \wedge (\mathtt{f}(\mathtt{y}, \mathtt{x}) = \mathtt{e}))$.

We can also write a formula saying that the group is abelian:

$$\forall \mathtt{x} \forall \mathtt{y}(\mathtt{f}(\mathtt{x}, \mathtt{y}) = \mathtt{f}(\mathtt{y}, \mathtt{x})),$$

along with a formula expressing that the center of the group is nontrivial:

$$\exists \mathtt{x}(\neg(\mathtt{x} = \mathtt{e}) \wedge \forall \mathtt{y}(\mathtt{f}(\mathtt{x}, \mathtt{y}) = \mathtt{f}(\mathtt{y}, \mathtt{x}))).$$

Perhaps unfortunately, we can also write syntactically correct formulas that express things nobody would ever utter, such as

$$\forall \mathtt{x} \exists \mathtt{y} \exists \mathtt{x}(\neg(\mathtt{e} = \mathtt{e})).$$

However, we can certainly handle some unnatural statements, and their existence makes it easier to generate, and reason about, the statements that actually matter.

What if we want to consider an area other than group theory? Commutative ring theory doesn't pose much of a problem, so long as we are allowed to alter the number of function symbols and constant symbols. We can simply have two function symbols $\mathtt{a}$ and $\mathtt{m}$ that take two arguments (where $\mathtt{a}$ represents addition and $\mathtt{m}$ represents multiplication) and two constant symbols 0 and 1 (where 0 represents the additive identity and 1 represents the multiplicative identity). Writing the axioms for commutative rings in this language is straightforward.

To take something fairly different, what about the theory of partial orderings? Recall that a partial ordering is a set P equipped with a subset $<$ of $P \times P$, where we write $x < y$ to mean that (x, y) is an element of this subset, satisfying the following:

- Irreflexive: For all $x \in P$, we have $x \not< x$.
- Transitive: If $x, y, z \in P$ are such that $x < y$ and $y < z$, then $x < z$.

Analogous to the syntax we used when handling the group operation, we will use notation that puts the ordering in front of the two arguments. Such an approach might seem odd at this point, given that we are putting equality in the middle, but we will see that such a convention provides a unifying notation for other similar objects. We thus introduce a *relation symbol* $\mathtt{R}$ (meant to represent $<$), and we keep the equality symbol $=$, but we no longer have a need for constant symbols or function symbols.

In this setting (without constant or function symbols), the only terms that we have (i.e. the only names for elements of the partial orderings) are the variables. However, our atomic formulas are more interesting because now there are two basic things we can say about elements of the partial ordering: whether they are equal and whether they are related by the ordering. Thus, our atomic formulas are things of the form $t_1 = t_2$ and $\mathtt{R}(t_1, t_2)$ where t_1 and t_2 are terms. From these atomic formulas, we build up all our formulas as described above.

We can now write formulas expressing the axioms of partial orderings:

- $\forall \mathtt{x} \neg \mathtt{R}(\mathtt{x}, \mathtt{x})$.
- $\forall \mathtt{x} \forall \mathtt{y} \forall \mathtt{z}((\mathtt{R}(\mathtt{x}, \mathtt{y}) \wedge \mathtt{R}(\mathtt{y}, \mathtt{z})) \to \mathtt{R}(\mathtt{x}, \mathtt{z}))$.

We can also write a formula saying that the partial ordering is a linear ordering:

$$\forall \mathtt{x} \forall \mathtt{y}(\mathtt{x} = \mathtt{y} \vee (\mathtt{R}(\mathtt{x}, \mathtt{y}) \vee \mathtt{R}(\mathtt{y}, \mathtt{x}))),$$

along with a formula expressing that there exists a maximal element:

$$\exists \mathtt{x} \forall \mathtt{y}(\neg \mathtt{R}(\mathtt{x}, \mathtt{y})).$$

The general idea is that by leaving flexibility in the types and number of constant symbols, relation symbols, and function symbols, we will be able to handle many areas of mathematics. We call this setup *first-order logic*. An analysis of first-order logic will consume the vast majority of our time.

Now we do not claim that first-order logic allows us to express everything in mathematics, nor do we claim that each of the setups just presented allows us to express everything of importance in that particular field. For example, take the group theory setting. We can express that every nonidentity element has order two with the formula

$$\forall \mathtt{x}(\mathtt{f}(\mathtt{x}, \mathtt{x}) = \mathtt{e}),$$

but it seems difficult to say that every element of the group has finite order. The natural guess is

$$\forall \mathtt{x} \exists \mathtt{n}(\mathtt{x}^{\mathtt{n}} = \mathtt{e}),$$

but this poses a problem for two reasons. First, our variables are supposed to quantify over elements of the group in question, not the natural numbers. Second, we put no construction in our language to allow us to write something like $\mathtt{x}^{\mathtt{n}}$. For each fixed n, we can express this (for $n = 3$ we can write $\mathtt{f}(\mathtt{f}(\mathtt{x}, \mathtt{x}), \mathtt{x})$, and for $n = 4$ we can write $\mathtt{f}(\mathtt{f}(\mathtt{f}(\mathtt{x}, \mathtt{x}), \mathtt{x}), \mathtt{x})$), but it is not clear how to generalize these ideas without allowing quantification over the natural numbers.

For another example, consider trying to express that a group is simple (i.e. has no nontrivial normal subgroups). The natural instinct is to quantify over all subsets H of the group G, and say that if it so happens that H is a normal subgroup, then H is either trivial or everything. However, we have no way to quantify over subsets. It is certainly possible to allow such constructions, which leads to *second-order logic*. We can even go further and allow quantification over sets of subsets (for example, one way of expressing that a ring is Noetherian is to say that every nonempty set of ideals has a maximal element), which leads to *third-order logic*, and so on.

Newcomers to the field often find it strange that we focus primarily on first-order logic. There are many reasons to give special attention to first-order logic (which we will come to appreciate throughout our study), but for now one should think of it as providing a simple example of a language that is capable of expressing many important aspects of various branches of mathematics. In fact, we will eventually understand that the limitations of first-order logic are precisely what allow us to prove powerful theorems about it. Moreover, these powerful theorems allow us to deduce interesting mathematical consequences.

1.3 Syntax and Semantics

In the preceding discussion, we introduced certain symbols that were intended to denote certain mathematical concepts (such as using $\wedge$ in place of "and," $\forall$ in place of "for all," and a function symbol $\mathtt{f}$ in place of the group operation f). Building and maintaining a careful distinction between formal symbols and how to interpret them is a fundamental aspect of mathematical logic.

The basic structure of the formal statements that we write down using the symbols, connectives, and quantifiers is known as the *syntax* of the logic that we are developing. Syntax corresponds to the grammar of the language in question with no thought given to meaning. Imagine an English instructor who cared nothing for the content of your writing, but only cared that it was grammatically correct. That is exactly what the syntax of a logic is all about. Syntax is combinatorial in nature and is based on simple rules that provide admissible ways to manipulate symbols devoid of any knowledge of their intended meaning.

The manner in which we are permitted (or forced) to interpret the symbols, connectives, and quantifiers is known as the *semantics* of the the given logic. In a logic, there are often some symbols that we are forced to interpret in specific rigid ways. For instance, in the preceding examples, we interpret the symbol $\wedge$ to mean *and*. In the propositional logic setting, this does not settle how to interpret a formula because we have not said how to interpret the elements of P. We have some flexibility here, but once we assert that we should interpret certain elements of P as true and the others as false, our formulas express statements that are either true or false.

The first-order logic setting is more complicated. Since we have quantifiers, the first thing that must be done in order to interpret a formula is to fix a set X that will act as the set of objects over which the quantifiers will range. With such a *universe* X in place, we can interpret each function symbol $\mathtt{f}$ taking k arguments as an actual function $f\colon X^k \to X$, each relation $\mathtt{R}$ symbol taking k arguments as a subset of X^k, and each constant symbol $\mathtt{c}$ as an element of X. Once we have fixed what we are talking about by providing such interpretations, we can view our formulas as expressing something meaningful. For example, if we have fixed a group G and interpret $\mathtt{f}$ as the group operation and $\mathtt{e}$ as the identity, then the formula

$$\forall x \forall y (f(x, y) = f(y, x))$$

is either true or false, according to whether G is abelian or not.

Always keep the distinction between syntax and semantics clear in your mind. Many basic theorems of the subject involve the interplay between syntax and semantics. For example, suppose that Γ is a set of formulas and that φ is a formula. We will eventually define what it means to say that Γ *implies* the formula φ. In the logics that we discuss, we will have two fundamental but seemingly distinct approaches. One way of saying that the formulas in Γ imply φ is semantic: whenever we provide an interpretation making all of the formulas of Γ true, it happens that φ is also true. For instance, if we are working in propositional logic and we have $\Gamma = \{((A \wedge B) \vee C)\}$ and $\varphi = (A \vee C)$, then Γ implies φ in this sense because whenever we assign true/false values to A, B, and C so that the formulas in Γ are true, it happens that φ will also be true. Another approach that we will develop is syntactic. We will define deductions as certain "formal proofs" built from certain permissible syntactic manipulations, and Γ will imply φ in this sense if there is a witnessing deduction. The Soundness Theorem and the Completeness Theorem for first-order logic (and propositional logic) say that the semantic version and syntactic version coincide. Thus, we can (amazingly!) mimic mathematical reasoning using purely syntactic manipulations.

1.4 The Point of It All

One important aspect, often mistaken as the *only* aspect, of mathematical logic is that it allows us to study mathematical reasoning using the tools of mathematics. The last sentence of the previous section provides a prime example. The Completeness Theorem says that we can capture the idea of one mathematical statement following from other mathematical statements using nothing more than syntactic rules on symbols. While this result is certainly computationally, philosophically, and foundationally interesting, it is also much more than that. A simple consequence is the Compactness Theorem, which says something very deep about mathematical reasoning, and also has many interesting applications within mathematics.

Although we have developed logics with the modest goal of handling certain fields of mathematics, it is a wonderful and surprising fact that we can embed (nearly) all of mathematics in an elegant and natural first-order system: first-order set theory. Consequently, we introduce the possibility of proving that certain mathematical statements are independent of our usual set theory axioms. In other words, we might be able to show that there exist formulas φ such that there is no deduction (from the usual axioms) of φ, and also no deduction of $(\neg\varphi)$. Furthermore, the field of set theory has blossomed into an intricate field with its own deep internal questions.

Other very interesting and fundamental subjects arise when we ignore the foundational aspects and deductions altogether and simply look at what we have accomplished by establishing a precise language to describe an area of mathematics. With a language in hand, we now have a way to say that certain objects are *definable* in that language. For instance, take the language of commutative rings. If we fix a particular commutative ring, then the formula $\exists y(m(x, y) = 1)$ has a free variable x and “defines” the set of units in the ring. Since we can mathematically reason about formulas, we have opened up the possibility of proving lower bounds on the complexity of any definition of a certain object, or even of proving that no such definition exists in the given language.

Another, closely related, way to take our definitions of precise languages and run with them leads to the subject of *model theory*. In group theory, we state some axioms and work from there in order to study all possible realizations of the axioms (i.e. all possible groups). However, as we have seen, the group axioms arise in one possible language with one possible set of axioms. Instead, we can study all possible languages and all possible sets of axioms and see what we can prove generally, and how the realizations compare to each other. In this sense, model theory is a kind of abstract abstract algebra.

Finally, although it is probably far from clear how it fits in at this point, *computability theory* is intimately related to the above subjects. To see the first glimmer of a connection, notice that computer programming languages are also formal languages with a precise grammar and a clear distinction between syntax and semantics. However, the connection runs much more deeply. Computational questions will arise early in both our study of propositional logic (at the end of Section 3.2) and first-order logic (at the end of Sections 4.2 and 4.5). We will eventually see that the computable sets play an important role in understanding definability in the natural numbers, and this connection will lead to the Incompleteness Theorems in Chapter 12.

1.5 Terminology and Notation

We briefly introduce some terminology and notation that we will use throughout the book. Our definitions of the natural numbers and finite sequences are chosen with an eye toward set theory. Once we reach Chapter 8, we will introduce some new notation for a few of these concepts, but we will still adopt the same conventions.

Definition 1.5.1 We let $\mathbb{N} = \{0, 1, 2, \dots\}$ and we let $\mathbb{N}^+ = \mathbb{N}\backslash\{0\}$.

Given a natural number n, it is somewhat common in combinatorics to use the notation $[n]$ for the set $\{1, 2, \dots, n\}$. We choose to start with 0 instead, to lay the groundwork for our eventual definition of the natural numbers in Section 8.4.

Definition 1.5.2 For each $n \in \mathbb{N}$, we let $[n] = \{m \in \mathbb{N} : m < n\}$, so $[n] = \{0, 1, 2, \dots, n-1\}$.

Notation 1.5.3 Let $f \colon A \to B$ be a function. Given $X \subseteq A$, we write $f \upharpoonright X$ for the function with domain X and codomain B that is obtained by restricting f to X.

We will regularly work with finite sequences. While these are often taken as primitive objects, we choose to define a sequence of length n as a function with domain $\{0, 1, \dots, n-1\}$. See Section 8.5 for the development of these ideas in set theory.

Definition 1.5.4 Let X be a set. Given $n \in \mathbb{N}$, we call a function $\sigma\colon [n] \to X$ a *finite sequence* from X of length n. We denote the set of all finite sequences from X of length n by X^n. We use λ to denote the unique sequence of length 0, so $X^0 = \{\lambda\}$. Finally, given a finite sequence σ from X, we use the notation $|\sigma|$ to denote the length of σ.

Definition 1.5.5 Given a set X, we let $X^* = \bigcup_{n \in \mathbb{N}} X^n$, that is, X^* is the set of all finite sequences from X.

We denote finite sequences by simply listing the elements in order. For instance, if $X = \{a, b\}$, then the sequence *aababbba* is an element of X^*. Sometimes, for clarity, we will insert commas and instead write a, a, b, a, b, b, b, a.

Definition 1.5.6 If $\sigma, \tau \in X^*$, we denote the concatenation of σ and τ by $\sigma\tau$.

Definition 1.5.7 If $\sigma, \tau \in X^*$, we say that σ is an *initial segment* of τ, and write $\sigma \sqsubseteq \tau$, if $\sigma = \tau \upharpoonright [n]$ for some n. We say that σ is a *proper initial segment* of τ, and write $\sigma \sqsubset \tau$ if $\sigma \sqsubseteq \tau$ and $\sigma \neq \tau$.

Since we use the notation $\overline{a}$ and $\underline{a}$ for some other concepts, we adopt the following notation for equivalence classes.

Notation 1.5.8 Given an equivalence relation $\sim$ on a set A, and an $a \in A$, we typically write $[a]$ for the equivalence class of a, that is, $[a] = \{b \in A : a \sim b\}$.

If $\sim$ is an equivalence relation on A, then $[a]$ is a subset of A. The set of all subsets of a given set will play an important role throughout our study.

Definition 1.5.9 Given a set A, we let $\mathcal{P}(A)$ be the set of all subsets of A, and we call $\mathcal{P}(A)$ the *power set* of A.

For example, we have $\mathcal{P}(\{1,2\}) = \{\emptyset, \{1\}, \{2\}, \{1,2\}\}$ and $\mathcal{P}(\emptyset) = \{\emptyset\}$. A simple combinatorial argument shows that if $|A| = n$, then $|\mathcal{P}(A)| = 2^n$.

We will also often want to change the value of a function on a single input, and it is useful to introduce some notation for that operation.

Notation 1.5.10 Let $f\colon A \to B$ be a function, and let $c \in A$ and $d \in B$. We let $f[c \rightsquigarrow d]$ denote the function with domain A and codomain B defined by

$$f[c \rightsquigarrow d](a) = \begin{cases} d & \text{if } a = c, \\ f(a) & \text{otherwise.} \end{cases}$$

That is, $f[c \rightsquigarrow d]$ agrees with f on all inputs with the exception of c, which is now sent to d.