1

Fundamentals

Every branch of mathematics has its subject matter, and one of the distinguishing features of logic is that so many of its fundamental objects of study are rooted in language. The subject deals with terms, expressions, formulas, theorems, and proofs. When we speak about these notions informally, we are talking about things that can be written down and communicated with symbols. One of the goals of mathematical logic is to introduce formal definitions that capture our intuitions about such objects and enable us to reason about them precisely.

At the most basic level, syntactic objects can be viewed as strings of symbols. For concreteness, we can identify symbols with particular set-theoretic objects, but, for most purposes, it does not matter what they are; all that is needed is that they are distinct from one another. A set of symbols is called an *alphabet*, and a *string* of symbols from the alphabet A is just a finite sequence of elements of A. Notions like the length of a string s and the concatenation $s^\frown t$ of two strings s and t are carried over from sequences. If $a_0, \ldots, a_{k-1}$ are symbols in some alphabet, '$a_0 \ldots a_{k-1}$' should be interpreted as the sequence $(a_0, \ldots, a_{k-1})$. These representations give much of logic a finitary, combinatorial, and computational flavor.

Nonetheless, abstraction can be helpful. What is essential about expressions like $((x + 7) \cdot (y + 9))$ is that they are built up from simple expressions – in this case, variables and numerical constants – using fixed operations in a systematic way. We ought to be able to prove things about such expressions inductively, and define operations on such expressions recursively, without descending to the level of symbols and strings. Functional programming languages often support recursive definitions on such inductively defined types.

The goal of this chapter is to develop a foundation for reasoning about syntax. While the definitions and theorems here underwrite many of the fundamental patterns of reasoning and inference in this book, most of those patterns are intuitively clear and natural when taken at face value. As a result, it would be reasonable to skim this chapter and refer back to it as necessary.

In logic, we state things about formal statements and prove things about formal proofs. This apparent circularity is sometimes confusing to novices. Philosophers and logicians often distinguish a *language* under study from the *metalanguage* used to study it, and a formal axiomatic *theory* from the *metatheory* that embodies the methods that are used to reason about it. Here our metatheory is simply everyday mathematics, as it is found in ordinary textbooks in algebra, analysis, or number theory. It is the subject matter, not the principles of reasoning, that sets mathematical logic apart.

1.1 Languages and Structures

Mathematics deals with structures. A *group* consists of a set of elements G together with a distinguished element 1 of G, a binary operation $\cdot$ on G, and an inverse function $x \mapsto x^{-1}$ from G to G, such that these data satisfy the group axioms. A *field* consists of a larger set of data, satisfying a different set of axioms. A *partial order* on a set A consists of a binary relation $\leq$ on A that is reflexive, antisymmetric, and transitive. An *equivalence relation* on a set A is a binary relation $\equiv$ on A that is reflexive, symmetric, and transitive.

Each of these can be viewed as a *structure* satisfying some *axioms*. We will later determine what sort of thing an axiom is and what it means to satisfy one. But first, we need to say what a structure is. In the examples above, each particular structure provides an interpretation of a certain set of symbols, such as $\{1, \cdot, \cdot^{-1}\}$ or $\{\equiv\}$, that are intended to denote functions or relations. Such a specification is known as a *language*.

Definition 1.1.1. A *language* is a triple (Γ, Δ, a), where Γ and Δ are disjoint sets of symbols and a is a function from $\Gamma \cup \Delta$ to $\mathbb{N}$. Γ is said to be the set of *function symbols* of the language, Δ is the set of *relation symbols*, and a assigns to each function and relation symbol its *arity*. If f is an element of Γ and $a(f) = k$, then f is said to be a *k-ary function symbol*. If R is an element of Δ and $a(R) = k$, then R is said to be a *k-ary relation symbol*.

Intuitively, a function is something that returns a value, whereas a relation is something that may or may not hold of its arguments. We can think of a 0-ary function as a constant value, that is, a function that returns a value without taking any arguments. Similarly, we can think of a 0-ary relation as a constant truth value. In the examples above, the language of groups has a 0-ary function symbol, 1, a binary function symbol, $\cdot$, and a unary function symbol, $\cdot^{-1}$. The language of equivalence relations has a single binary relation symbol, $\equiv$.

The word "language" is misleading since a language is really a specification of a basic vocabulary from which complex expressions can be built. Later on, we will also consider other kinds of specifications. The present notion is also called a *signature*, and sometimes a *first-order language* to distinguish it from other kinds of languages. First-order languages can be used to reason about algebraic structures like groups and fields; to reason about particular structures like the natural numbers and the real numbers; or to give foundational accounts of the entire universe of mathematical objects.

Definition 1.1.2. If $L = (\Gamma, \Delta, a)$ is a language, a *structure* for L (or an *L-structure*) consists of a set U and a function I that assigns to each k-ary function symbol in Γ a k-ary function from U to U and to each k-ary relation symbol in Δ a k-ary relation on U.

An L-structure is also often called a *model for L*, or simply a *model* when the language is understood. If $\mathfrak{A} = (U, I)$ is an L-structure, we typically write $|\mathfrak{A}|$ for the set U, called the *universe of* $\mathfrak{A}$, $f^{\mathfrak{A}}$ instead of $I(f)$ for the interpretation of the function symbol f in $\mathfrak{A}$, and $R^{\mathfrak{A}}$ instead of $I(R)$ for the interpretation of the relation symbol R. For example, if L is a language with one 0-ary function symbol c (i.e. a constant symbol), two binary function symbols f and g, and one binary relation symbol R, then we can interpret L in the structure with universe $\mathbb{N}$, the constant 0, functions $+$ and $\cdot$, and relation $\leq$. For convenience, we will typically refer to this as the structure $(\mathbb{N}, 0, +, \cdot, \leq)$, leaving the correspondence with the symbols c, f, g, and R to be inferred from context.

If G and H are groups, a *homomorphism* $\varphi\colon G \to H$ is a function that maps 1^G to 1^H and respects multiplication and inverses. Saying that φ respects multiplication means that $\varphi(g_1 \cdot^G g_2) = \varphi(g_1) \cdot^H \varphi(g_2)$ holds for every g_1 and g_2 in G, and saying that it respects inverses means that $\varphi(g^{-1}) = \varphi(g)^{-1}$ holds for every g in G, where the first inverse is computed in G and the second one is computed in H. (By the usual abuse of notation, we have written G for both the group structure and the underlying carrier set, $|G|$.) If $\equiv$ is an equivalence relation on a set A and $\sim$ is an equivalence relation on a set B, then a homomorphism from $(A, \equiv)$ to $(B, \sim)$ is a function $\varphi\colon A \to B$ with the property that whenever $a_1 \equiv a_2$, we have $\varphi(a_1) \sim \varphi(a_2)$. Both of these are instances of a general notion.

Definition 1.1.3. Let L be a language and let $\mathfrak{A}$ and $\mathfrak{B}$ be L-structures. Then a *homomorphism* φ from $\mathfrak{A}$ to $\mathfrak{B}$ is a function from $|\mathfrak{A}|$ to $|\mathfrak{B}|$ that satisfies the following two requirements:

- For every k-ary function symbol f of L and every tuple $a_0, \ldots, a_{k-1}$ of elements of $|\mathfrak{A}|$, $\varphi(f^{\mathfrak{A}}(a_0, \ldots, a_{k-1})) = f^{\mathfrak{B}}(\varphi(a_0), \ldots, \varphi(a_{k-1}))$.
- For every k-ary relation symbol R of L and every tuple $a_0, \ldots, a_{k-1}$ of elements of $|\mathfrak{A}|$, if $R^{\mathfrak{A}}(a_0, \ldots, a_{k-1})$, then $R^{\mathfrak{B}}(\varphi(a_0), \ldots, \varphi(a_{k-1}))$.

Notice that the implication in the second clause of the definition of a homomorphism does not necessarily reverse: a homomorphism is required to *preserve* each of the relations in the source structure but not *reflect* them. A homomorphism is called an *embedding* if it is injective and satisfies the following strengthening of the second clause:

- For every k-ary relation symbol R of L and every tuple $a_0, \ldots, a_{k-1}$ of elements of $|\mathfrak{A}|$, $R^{\mathfrak{A}}(a_0, \ldots, a_{k-1})$ if and only if $R^{\mathfrak{B}}(\varphi(a_0), \ldots, \varphi(a_{k-1}))$.

In other words, an embedding both preserves and reflects the relations.

A homomorphism $\varphi\colon \mathfrak{A} \to \mathfrak{B}$ is an *isomorphism* if there is a homomorphism $\psi\colon \mathfrak{B} \to \mathfrak{A}$ such that $\psi \circ \varphi$ is the identity on $\mathfrak{A}$ and $\varphi \circ \psi$ is the identity on $\mathfrak{B}$. Exercise 1.1.1 asks you to show that $\varphi\colon \mathfrak{A} \to \mathfrak{B}$ is an isomorphism if and only if it is a surjective embedding. Two structures for a language L are said to be *isomorphic* if there is an isomorphism between them. A homomorphism from a structure $\mathfrak{A}$ to itself is called an *endomorphism*, and an isomorphism from a structure $\mathfrak{A}$ to itself is called an *automorphism*.

Think of a homomorphism φ from $\mathfrak{A}$ to $\mathfrak{B}$ as a translation between the two structures. The first clause says that applying a function in $\mathfrak{A}$ to some elements in the universe of $\mathfrak{A}$ and then translating to $\mathfrak{B}$ gives the same result as translating the elements to $\mathfrak{B}$ and applying the corresponding function in $\mathfrak{B}$. The second clause says that relations in $\mathfrak{A}$ are preserved by the translation. For example, $\varphi(x) = 2x$ is an embedding of $(\mathbb{Z}, 0, +, \leq)$ in $(\mathbb{Z}, 0, +, \leq)$, and $\psi(x) = 0$ is a homomorphism between those two structures. The identity function on $\mathbb{Z}$ is an embedding of $(\mathbb{Z}, 0, 1, +, \cdot, \leq)$ in $(\mathbb{R}, 0, 1, +, \cdot, \leq)$, and the function $h(x) = e^x$ is an isomorphism of $(\mathbb{R}, 0, +, \leq)$ and $(\mathbb{R}^{>0}, 1, \cdot, \leq)$, where $\mathbb{R}^{>0}$ denotes the positive real numbers.

With the definitions above, saying that $\mathfrak{A}$ is an L-structure means that we have chosen specific symbols to represent the relevant data. Mathematically, groups are sometimes written with multiplicative notation $1, \cdot, \cdot^{-1}$, sometimes with additive notation $0, +$, and $-$, and sometimes with neutral symbols, such as $e, \cdot$, and i. In the terminology introduced here, each language gives rise to a different kind of a structure, say, multiplicative group structures,

additive group structures, and general group structures. Whether or not this is a good thing is subject to debate, but, in any case, any structure of one kind can be mapped to a structure of one of another kind in such a way that the identity function on the underlying set will be an isomorphism between them. There are other approaches to talking about languages and structures; one is to take a signature to be a specification of arities without a choice of function symbols, and a structure for that signature to be an ordered sequence of interpretations. The various approaches are generally intertranslatable.

In Chapter 5, we will make use of the notion of a *quotient* construction for structures. Let $\mathfrak{A}$ be a structure for a language L and $\sim$ be an equivalence relation on $|\mathfrak{A}|$. Suppose furthermore that all the functions and relations in $\mathfrak{A}$ respect the congruence, as described in Appendix A.2. We define $\mathfrak{A}/\sim$ to be the structure with universe $|\mathfrak{A}|/\sim$, where for every k-ary function symbol f of L, $f^{\mathfrak{A}/\sim}([a_0], \ldots, [a_{k-1}])$ is defined to be $[f^{\mathfrak{A}}(a_0, \ldots, a_{k-1})]$, and for every k-ary relation symbol R of L, $R^{\mathfrak{A}/\sim}([a_0], \ldots, [a_{k-1}])$ holds if and only if $R^{\mathfrak{A}}(a_0, \ldots, a_{k-1})$. The function $\varphi(a) = [a]$ is then a surjective homomorphism from $\mathfrak{A}$ to $\mathfrak{A}/\sim$, and $a \sim b$ holds of elements of $|\mathfrak{A}|$ if and only if $\varphi(a) = \varphi(b)$. Thus, the quotient construction turns the equivalence relation $\sim$ on $\mathfrak{A}$ into equality on $\mathfrak{A}/\sim$.

Exercises

1.1.1. Show that a function $\varphi\colon \mathfrak{A} \to \mathfrak{B}$ between two L-structures is an isomorphism if and only if it is a surjective embedding.

1.1.2. Show that the composition of two homomorphisms is a homomorphism, and similarly for embeddings and isomorphisms.

1.1.3. Show that isomorphism is an equivalence relation.

1.1.4. For each of the following pairs, show that the two structures are isomorphic:
   a. $((a, b), <)$ and $((c, d), <)$, where $a, b, c, d \in \mathbb{R}$, $a < b$, $c < d$, and (a, b) denotes the open interval $\{x \mid a < x < b\}$
   b. $((0, 2), 1, <)$ and $(\mathbb{R}, 0, <)$
   c. $(\mathbb{R}, 0, +, <)$ and $(\mathbb{R}^{>0}, 1, \cdot, <)$, where $\mathbb{R}^{>0}$ denotes the positive real numbers.

1.1.5. For each of the following pairs, show that the two structures are *not* isomorphic:
   a. $(\mathbb{N}, <)$ and $(\mathbb{N}, >)$
   b. $((0, 1), <)$ and $((0, 1], <)$, where $(0, 1]$ denotes the half-closed interval $\{x \mid 0 < x \leq 1\}$
   c. $((0, 1) \cup (1, 2), <)$ and $((0, 2), <)$.

1.1.6. Determine all the endomorphisms and automorphisms of each of the following structures: $(\mathbb{N}, <)$, $(\mathbb{N}, +)$, $(\mathbb{R}, +)$, $(\mathbb{R}, +, <)$, and $(\mathbb{R}, \cdot)$.

1.1.7. Verify the last claim in this section: if $\mathfrak{A}$ is any structure and $\sim$ is an equivalence relation on $|\mathfrak{A}|$, then $\varphi(a) = [a]$ is a surjective homomorphism from $\mathfrak{A}$ to $\mathfrak{A}/\sim$ that preserves the relations in $\mathfrak{A}$, and for every a and b in $|\mathfrak{A}|$, $a \sim b$ if and only if $\varphi(a) = \varphi(b)$.

1.2 Inductively Defined Sets

The natural numbers can be characterized inductively as a set that contains a distinguished element, 0, and is closed under an injective operation, $\mathrm{succ}(n)$, that returns the *successor* of n. The inductive character amounts to the fact that the sequence $0, \mathrm{succ}(0), \mathrm{succ}(\mathrm{succ}(0)), \ldots$ exhausts the set of natural numbers, in the following sense: if $A \subseteq \mathbb{N}$ contains 0 and is closed

under succ, then $A = \mathbb{N}$. If we associate any property P of natural numbers with the set of numbers satisfying P, the preceding statement amounts to the principle of induction on $\mathbb{N}$.

It is often useful to characterize sets of expressions in a similar fashion. We can view arithmetic expressions like $((x + 7) \cdot (y + 9))$ as being built up from variables and numeric constants by the syntactic operations of forming sums and products. The utility of such a perspective is not limited to syntax, and we often come across sets and structures in mathematics that are generated in such a way. For example, if G is a group and S is a subset of G, the subgroup $\langle S \rangle$ generated by S is the smallest subset of G containing S and closed under the group operations. Similarly, the collection of Borel subsets of $\mathbb{R}$ is the smallest collection of subsets of $\mathbb{R}$ containing the open sets and closed under the operations of forming complements and countable unions.

In this section, we will develop a very general, abstract framework for describing sets of elements that are defined inductively, from the bottom up. Our high-level approach is somewhat heavy-handed, and we could certainly develop a theory of syntax in more concrete terms. But as the examples below indicate, the approach has applications beyond defining terms and expressions, and abstracting the common features provides a clear understanding of the essential features of the constructions.

We start with a set U of objects, a *universe*, within which the construction takes place. A *rule on* U is just a pair (S, a), where S is a subset of U and a is an element of U. We will think of a set of rules as a recipe for constructing a set of objects, where the rule (S, a) says "if the elements of S are in the set, then a must be in the set as well." Our goal is to construct a set that consists of only those elements that are *required* to be there by the rules. Notice that if S is the empty set, then the rule (S, a) asserts outright that a must be in the set we construct. Such rules are the starting point for the construction. If there are no such rules, the empty set itself satisfies all the requirements.

Let $\mathfrak{R}$ be a set of rules on U. Say that a set B is *inductive with respect to* $\mathfrak{R}$ or *closed under* $\mathfrak{R}$ if it meets the specification above, that is, for each rule $(S, a) \in \mathfrak{R}$, if $S \subseteq B$, then $a \in B$. Let

$$A = \bigcap \{B \subseteq U \mid B \text{ is inductive}\}.$$

In words, A is the intersection of all inductive subsets of U, so an element a of U is in A if and only if it is in *every* inductive subset of U. The next proposition shows that A is the set we are after.

Proposition 1.2.1. *The following hold:*

1. *A is inductive.*
2. *If $B \subseteq A$ is inductive, then $B = A$.*

Proof For the first claim, suppose $(S, a) \in \mathfrak{R}$ and $S \subseteq A$. By the definition of A, $S \subseteq B$ for every inductive subset B of U. But if B is inductive and $S \subseteq B$, then a is in B. So a is in every inductive set as well, and so $a \in A$.

For the second claim, if $B \subseteq U$ is inductive, then $A \subseteq B$, since every element of A is in every inductive set. Since we are assuming $B \subseteq A$, we have $B = A$. □

The first part of Proposition 1.2.1 says that A is closed under the rules, while the second part of the proposition says that A is the smallest such set. In practice, the relevant set of rules is often described with a list of conditions, as in the following example.

Example. Let G be any group and S be any nonempty subset of G. Then the *subgroup of G generated by S* is the smallest subset H of the carrier of G that satisfies the following:

- If g is any element of S, then g is in H.
- If g_1 and g_2 are in H, then so is $g_1 g_2$.
- If g is in H, then so is g^{-1}.

To express this in formal terms, take the universe U to be the carrier of G and take $\mathfrak{R}$ to be the union of the following three sets:

- $\{(\emptyset, g) \mid g \in S\}$
- $\{(\{g_1, g_2\}, g_1 g_2) \mid g_1, g_2 \in G\}$
- $\{(\{g\}, g^{-1}) \mid g \in G\}$.

Then H is the subset of G defined inductively by $\mathfrak{R}$.

Notice that there is a theorem implicit in our use of the phrase "the subgroup of G generated by S," namely, that the inductively defined set is in fact a subgroup. This follows easily from the closure under the rules. The inductive character implies that if K is any other subgroup of G containing S, then $H \subseteq K$.

I will leave it to you to carry out similar translations for the next three examples.

Example. The collection of *Borel subsets of* $\mathbb{R}$ is the smallest subset B of $\mathcal{P}(\mathbb{R})$ that satisfies the following:

- If a, b are in $\mathbb{R}$, then the open interval (a, b) is in B.
- If S is in B, so is $\overline{S}$, the complement of S.
- If $(S_i)_{i \in \mathbb{N}}$ is a countable sequence of subsets of $\mathbb{R}$ and each S_i is in B, then so is $\bigcup_i S_i$.

Example. Let A be any set and R be any binary relation on A. Then the *transitive closure* of R is the relation R' defined inductively as follows:

- For any $a, b \in A$, if $R(a, b)$, then $R'(a, b)$.
- For any $a, b, c \in A$, if $R'(a, b)$ and $R'(b, c)$, then $R'(a, c)$.

The closure and inductive properties imply that R' is the smallest transitive relation on A containing R.

Example 1.2.2. Let U be a collection of sets containing $\emptyset$ and closed under the function $\mathrm{succ}(a) = a \cup \{a\}$. (The set-theoretic *axiom of infinity* states precisely that there exists such a set U.) Then, in set-theoretic terms, the set of natural numbers, $\mathbb{N}$, can be defined as the smallest set containing $\emptyset$ and closed under succ. Similarly, let U be a collection of sets containing $\emptyset$ and closed under succ and countable unions. Then the smallest subset of U with these properties is exactly the set of countable ordinals.

Example 1.2.3. More generally, suppose U is any set of objects and $\mathfrak{F}$ is a set of functions from U to U of various arities, including constants. Then we can define the set A to be the smallest subset of U closed under all the elements of $\mathfrak{F}$. More precisely, A is the smallest subset of U containing all the constants in $\mathfrak{F}$, and closed under all the functions. This set has the following two properties:

- It contains all the constants in $\mathfrak{F}$ and is closed under each of the functions.
- If $B \subseteq A$ has these properties, then $B = A$.

Since this example will figure prominently in this book, I will describe the set of rules explicitly. For each k-ary function $f \in \mathfrak{F}$ and sequence $a_0, \ldots, a_{k-1}$, we add the rule $(\{a_0, \ldots, a_{k-1}\}, f(a_0, \ldots, a_{k-1}))$. Intuitively, this says that as soon as $a_0, \ldots, a_{k-1}$ are included in the set, we should include $f(a_0, \ldots, a_{k-1})$ as well. Each constant c in $\mathfrak{F}$ corresponds to the rule $(\emptyset, c)$.

The terminology commonly used to describe these constructions varies. In the general situation, we may say that A is the *smallest set closed under the rules in* $\mathfrak{R}$ or the set *inductively generated by* $\mathfrak{R}$. Similar language is used to describe the construction of Example 1.2.3. For example, we have already described the natural numbers as the *smallest set containing* 0 *and closed under* succ in Example 1.2.2, and we might also say that $\mathbb{N}$ is *inductively generated from* $\{0\}$ *by* succ.

The second claim of Proposition 1.2.1 is really an induction principle for A. It implies that in order to show that every element of A has some property P, it suffices to show that for each rule (S, a) in $\mathfrak{R}$, if $S \subseteq A$ and P holds of every element of S, then P holds of a. To see that this principle follows from Proposition 1.2.1, given P, let B be the set of elements of A satisfying P. The hypothesis of the principle says exactly that B is closed under the set of rules in $\mathfrak{R}$, and the second claim of Proposition 1.2.1 then implies that every element of A satisfies P.

The abstract characterization of A as the intersection of all inductive subsets of U is clever, but, from a foundational point of view, it is heavy-handed: we have defined A by reference to the collection of *all* the subsets of U, which may be very large. To make matters worse, it also contains the very object A that is being defined, making it a prototypical example of an *impredicative definition*. We can often provide a more explicit description of the set we are after. If the set of rules $\mathfrak{R}$ has the property that each rule $(S, a) \in \mathfrak{R}$ is finite, which is to say, the set S is finite, then we define a finite sequence $(a_0, \ldots, a_{n-1})$ of elements of U to be a *formation sequence* (again with respect to $\mathfrak{R}$) if, for each i, there is a rule (S, a_i) with $S \subseteq \{a_0, \ldots, a_{i-1}\}$. Intuitively, this says that every element of the formation sequence is justified by previous elements of the sequence.

Proposition 1.2.4. *With U and $\mathfrak{R}$ as above, let A be the subset of U defined inductively by $\mathfrak{R}$. Then A is equal to the set of elements a of U such that there is a formation sequence containing a.*

Proof Let B be the set of elements a of U such that there is a formation sequence containing a. To show that A is a subset of B, we use the induction principle for A. Suppose (S, a) is a rule in $\mathfrak{R}$ and there is a formation sequence for each element of S. Since S is finite, we can concatenate these and append a to obtain a formation sequence for a.

On the other hand, to show that B is a subset of A, let $(a_0, \ldots, a_{n-1})$ be any formation sequence. Then, using the definition of a formation sequence, it is easy to show by induction on the natural numbers that for each i, $a_i \in A$. □

The constructions described here can be described in other terms. Given a universe U and a set $\mathfrak{R}$ of rules, define the function $\Gamma : \mathcal{P}(U) \to \mathcal{P}(U)$ by

$$\Gamma(B) = \{a \in U \mid \text{for some } S \subseteq B,\ (S, a) \in \mathfrak{R}\}.$$

In other words, $\Gamma(B)$ consists of the elements that should be added to B in conformance with the rules. It is easy to check that Γ is *monotone*: whenever $B \subseteq C$, we have $\Gamma(B) \subseteq \Gamma(C)$. Exercise 1.2.2 asks you to show that any monotone operator from $\mathcal{P}(U)$ to $\mathcal{P}(U)$ has a

least fixed point, which is to say, there is a set A such that $\Gamma(A) = A$ and whenever $\Gamma(B) = B$, $A \subseteq B$. The construction we have described here is the special case where Γ is defined from a set of rules, as above. Exercise 1.2.4 asks you to show that, conversely, given any monotone operator Γ, the least fixed point of Γ can be characterized as the smallest set closed under a suitable set of rules.

Exercises

1.2.1. For each of the examples of inductive definitions in this section, define a corresponding set of rules explicitly.

1.2.2. Let U be a set, let Γ be a monotone function from $\mathcal{P}(U)$ to $\mathcal{P}(U)$, and let

$$A = \bigcap \{B \in \mathcal{P}(U) \mid \Gamma(B) \subseteq B\}.$$

Note that $\Gamma(U) \subseteq U$, so A is the intersection of a nonempty set.

a. Show that $\Gamma(A) \subseteq A$. (Hint: show $\Gamma(A) \subseteq B$ whenever $\Gamma(B) \subseteq B$.)
b. Show that $A \subseteq \Gamma(A)$. So $A = \Gamma(A)$ is a fixed point of Γ.
c. Show that if B is any other fixed point, $A \subseteq B$.

So A is the least fixed point. The construction generalizes to arbitrary complete lattices. The statement that every monotone function on a complete lattice has a least fixed point is called the *Knaster–Tarski* theorem.

1.2.3. The following provides an alternative "bottom-up" definition of the least fixed point of Γ, using principles of transfinite recursion along suitable ordinals. (It requires some basic set theory.)

Let Γ be monotone. Define a sequence of subsets of U by transfinite recursion on the ordinals, as follows:

- $A_0 = \emptyset$
- $A_{\alpha+1} = \Gamma(A_\alpha)$
- $A_\lambda = \bigcup_{\alpha<\lambda} A_\alpha$, for limit ordinals λ.

Do the following:

a. Show that whenever $\alpha < \beta$, $A_\alpha \subseteq A_\beta$.
b. Show, using cardinality considerations, that for some α, $A_\alpha = A_{\alpha+1}$. (In fact, $\alpha < |U|^+$, the least cardinal larger than the cardinality of U.) After that, the process stabilizes, so A_α is a fixed point of Γ.
c. Show that if B is any fixed point of Γ, then for every β, $A_\beta \subseteq B$. In particular, A_α is the least fixed point.

1.2.4. Let Γ be a monotone function from $\mathcal{P}(U)$ to $\mathcal{P}(U)$ and define the set of rules $\mathfrak{R} = \{(S, a) \mid a \in \Gamma(S)\}$.
a. Show that for any $B \subseteq U$, $\Gamma(B) \subseteq B$ if and only if B is closed under $\mathfrak{R}$.
b. Show that the least fixed point of Γ is exactly the subset of U inductively generated by $\mathfrak{R}$.

1.2.5. Show that every monotone function from $\mathcal{P}(U)$ to $\mathcal{P}(U)$ also has a *greatest fixed point*.

1.3 Terms and Formulas

We will now apply the abstract machinery we have just developed to the relatively concrete tasks of defining syntactic objects like terms and formulas. Let L be a language without any relation symbols, or, if there are any, just ignore them for now. Fix a stock of variables $x_0, x_1, x_2, \ldots$ different from the symbols of L. (We will generally be interested in languages

with countably many symbols, and we will generally need only countably many variables. In principle, however, nothing prevents us from using uncountably many symbols and variables, and this is often useful in model theory.) We want the *terms* of L to be expressions built up from the variables and constant symbols using the function symbols in the language. For example, if L has a constant symbol, c, a unary function symbol, f, and two binary function symbols, h and k, then the following are examples of terms in L:

$$c,\ \ x_0,\ \ x_1,\ \ f(c),\ \ f(x_0),\ \ h(f(x_0), x_1),\ \ k(f(c), h(f(x_0), x_1)).$$

Formally, we will take a term in this language to be a string of symbols in alphabet that includes the symbols in F, the variables, symbols for the open- and close-parentheses, and a comma. We assume that all these symbols are distinct from one another.

Definition 1.3.1. Let $L = (\Gamma, \Delta, a)$ be a language. The set of *terms* of L is the smallest set of strings over the alphabet above satisfying the following:

- If x is a variable, then x is a term.
- If c is a constant symbol in Γ (a 0-ary function symbol), then c is a term.
- If $f \in \Gamma$ has arity k and $t_0, \ldots, t_{k-1}$ are all terms, then so is $f(t_0, \ldots, t_{k-1})$.

I have taken some notational liberties in this definition. The first condition says, more precisely, that if x is any variable, then the string 'x' is a term. The conclusion of the third condition says, more properly, that the string '$f($'$\frown t_0 \frown$'$,$'$\frown \ldots \frown$'$,$'$\frown t_k \frown$'$)$' is a term. I will generally rely on the more convenient manner of presentation above and leave these details implicit.

There are additional syntactic nuances. Assuming we have fixed a set of variables $\{x_0, x_1, \ldots\}$, the inscription "x" in the definition is a variable ranging over these symbols. Similarly, the inscriptions "t_0," ..., "t_{k-1}" in the third condition range over terms, which is to say, they are variables ranging over syntactic expressions. Bearing in mind the distinction between theory and metatheory, these are sometimes called *metavariables*, to distinguish them between the symbols for variables in the formal language we are constructing. Once again, I will avoid such ponderous language and trust you to be mindful of the difference.

Notice that this definition is an instance of Example 1.2.3, since the set of terms is generated by the following collection of functions on strings:

- for each variable x, the string 'x'
- for each 0-ary symbol c in Γ, the string 'c'
- for each k-ary symbol f in F with $k > 0$, the k-ary function $\bar{f}$ which takes as input arbitrary strings $t_0, \ldots, t_{k-1}$ and assembles the string $f(t_0, \ldots, t_{k-1})$.

We can use induction to prove some basic facts about terms.

Proposition 1.3.2. *Every term in a language L has the same number of left and right parentheses.*

Proof The claim is true of the base cases, namely, the variables and constants. And, assuming it is true for $t_0, \ldots, t_{k-1}$, it is also true of the string of symbols $f(t_0, \ldots, t_{k-1})$. □

The following is even easier to prove by induction:

Proposition 1.3.3. *Let t be any term of L. Then either t is a constant symbol, or t is a variable, or there are a k-ary function symbol f of L and a sequence of terms $s_0, \ldots, s_{k-1}$ such that t is the string $f(s_0, \ldots, s_{k-1})$.*

What is not nearly as obvious is that a given term falls into one of these three categories in a unique way. Establishing this stronger fact has to rely on the specific details of the definition. If we had allowed infix notation like $t_0 + t_1$ and $t_0 \cdot t_1$ and failed to include parentheses, then a term like $x_0 + x_1 \cdot x_2$ would be ambiguous; it could arise from applying the $+$ construction to x_0 and $x_1 \cdot x_2$, or the $\cdot$ construction to $x_0 + x_1$ and x_2. We therefore need to prove that our current definition avoids such ambiguity.

Theorem 1.3.4. *On the set of terms, the generating functions are injective and their images are disjoint from one another.*

Proof It is easy to see that the images of all the operations are disjoint, because each string begins with the corresponding symbol, and we are assuming that the variables, constant symbols, and function symbols are all distinct. So all that remains is to show that each term-forming operation is injective.

So suppose f is a k-ary function symbol with $k > 0$, $t_0, \ldots, t_{k-1}$, $t'_0, \ldots, t'_{k-1}$ are all terms, and $f(t_0, \ldots, t_{k-1})$ and $f(t'_0, \ldots, t'_{k-1})$ are equal, which is to say, they are the same string of symbols. Dropping the first two characters and the last character, we have that the two strings '$t_0, \ldots, t_{k-1}$' and '$t'_0, \ldots, t'_{k-1}$' are the same. We need to show that each t_j is equal to t'_j, which is to say, the two strings are the same. In other words, we need to show that a string of symbols cannot be parsed as the concatenation of k-terms, separated by commas, in two distinct ways.

To prove this, we first establish an auxiliary claim. If s and t are strings of symbols, say that s is a *proper initial segment of* t if s is strictly shorter than t, and the two sequences of characters agree up to the length of s. I claim that if t is any term and s is a proper initial segment of t, then s is *not* a term. We prove this by induction on terms. The only proper initial segment of a constant or variable is the empty string, which is not a term (prove this by induction on terms as well). In the induction step, suppose f is a k-ary function symbol with $k > 0$, and s is a proper initial segment of $f(t_0, \ldots, t_{k-1})$. Then it has one of the following forms:

- the empty string
- f
- $f($
- $f(t_0, \ldots, t_j$
- $f(t_0, \ldots, t_j, t'$

where, in the last case, t' is a proper initial segment of t_{j+1}. It is not hard to verify that the first three are not terms (again, using induction on terms). We have already established that every term has the same number of left and right parentheses, so the fourth case has more left parentheses than right parentheses, and hence is not a term. To handle the last case, we need to establish the slightly stronger assertion that any initial segment of a term has no more right parentheses than left parentheses, which is again easy to do, by induction on terms.

Returning to the main proof, since '$t_0, \ldots, t_{k-1}$' and '$t'_0, \ldots, t'_{k-1}$' are the same string, either t_0 is equal to t'_0, or one is a proper initial segment of the other. By the previous claim, the latter is impossible, so t_0 and t'_0 are equal. Dropping these and the subsequent comma from each string, we have that '$t_1, \ldots, t_{k-1}$' and '$t'_1, \ldots, t'_{k-1}$' are the same string. Proceeding iteratively in this way, we obtain that each t_j is equal to t'_j, as required. □