Formal Foundations of Linguistic Theory

Carl Pollard

Contents

Introduction

Even though linguistics departments are, by tradition, usually located in colleges of humanities, linguistics itself aspires to be—and at its best, manages to be—a science. That is, linguists aim to do much the same thing that scientists in general (such as physicists, geologists, biologists, and chemists) do: to make observations of certain kinds of natural phenomena, and then state **empirical hypotheses** about them. The only difference is that the phenomena linguists study have to do not with swinging pendulums, tectonic plates, zebra mussels, or hydrocarbons, but with human language: how it sounds, what it means, how it varies across space and time, how it is learned, used, and understood.

Roughly speaking, an empirical hypothesis is just a well-informed and careful guess about what certain kinds of events will be like, based on past observations of events of that kind. To put it a bit more precisely, an empirical hypothesis is a general statement about a class of phenomena that has the following properties: (1) It is *clear and unambiguous*, that is, there is no question what it asserts (how things would have to be in order for it to be true). (2) It is *general*, in the sense that even though it is based only on a finite number of observations, it makes predictions about how other phenomena of the same kind will unfold. (3) There is a way to tell whether or not a given observation of the kind of phenomenon in question is consistent with it, so that if the hypothesis is wrong, there is some hope of finding out that it is wrong. This third property of empirical hypotheses is called **falsifiability**. Especially valued are empirical hypotheses with the additional property of being *illuminating*, in the sense of being sufficiently simple and comprehensible to help us grasp some of the hidden orderliness or systematicity in seemingly random or chaotic phenomena.

A **linguistic theory** is just a set of empirical hypotheses about a class of natural-language phenomena. Linguists often refer to the process of formulating empirical hypotheses about human languages as 'capturing linguistic generalizations'. This is just a fancy name for linguistic theorizing. The purpose of this book is to introduce some techniques for doing just that.

The techniques introduced in this book are drawn from areas of mathematics (such as set theory, logic, algebra, and formal language theory) that are usually described as **discrete**, as opposed to **continuous** (such as calculus, differential equations, Fourier analysis, or probability). The natural numbers are discrete; the real numbers are continuous. The subdisciplines of linguistics that most readily lend themselves to analysis by discrete methods include (but are not limited to) the following: (1) **morphology** (how words are built up from their meaningful subparts); (2) **syntax** (how words combine into successively larger phrases, including sentences); and (3) **semantics** (how linguistic expressions manage to refer to things in the world and express propositions about them, and how it is that some propositions follow from (or are **entailed** by) other propositions). There are also parts of **phonology** (how human languages structure spoken sounds) and **computational linguistics** (the analysis and manipulation of human language using computational concepts or computer programs) that yield to such methods. But other linguistic disciplines, such as phonetics, psycholinguistics, sociolinguistics, and historical linguistics in general call for continuous methods. Interestingly, many of the discrete mathematical techniques that come into play in the analysis of human language are the same ones used in analyzing the artifical languages employed in logic and computer science.

This is an applied mathematics book, not a linguistics book, and so the emphasis is primarily on the mathematical concepts and techniques themselves, not on the phenomena to which they are applied. In fact, most of these are of inherent interest independent of the linguistic applications, and it is entirely possible to master them without knowing or caring about linguistics at all! But the book is written primarily with the needs of linguistics students in mind.

Chapter 1

Sets

1.1 Introduction

Scientific theories usually do not directly describe the natural phenomena under investigation, but rather a mathematical idealization of them that abstracts away from various complicating factors. For example, a theory about how the earth, the sun, and the moon move under mutual gravitation might ignore such complications as the sizes of the three bodies, friction arising from the presence of interstellar dust, the gravitational force exerted by other planets and stars, or relativistic effects that become significant only as the velocities of the bodies in question approach the speed of light. In the mathematical idealization, the time might be represented by a real number; the mass of each of the three bodies by a positive real number; its location in space (or more precisely the location of its center of gravity) at a particular time by three real numbers (the x, y, and z coordinates relative to a coordinate system); its velocity at a particular time by three more real numbers; the state of the three-body system at a given time (real number) t by the 18 real numbers that specify the locations and velocities of the three bodies at time t; and the evolution of the system over time by 18 functions that give the value of each of these 18 parameters at each time t. And the theory itself is a mathematical specification of which evolutions ('paths') through 18-dimensional Euclidean space) are possible. Armed with such a theory, we can predict, given the state of the system at a given time t_0, what state it will be in at any future time t_1.

Linguistic theories make predictions not about celestial bodies, but rather about natural languages, for example: how their words can sound; how their words can be combined into phrases; what meanings they can express;

which natural-language arguments are judged valid; or how the meanings of sentences can be related to the meanings of the words they contain. As it turns out, the kinds of mathematical entities that have proven to be useful for representing such things (words, phrases, sentences, their meanings, valid arguments, etc.) are not real numbers or real-valued functions, but rather discrete (roughly, non-continuous) things such as natural numbers, strings, trees, algebras, formal languages, and proof systems. In linguistics these mathematical idealizations are often called **representations** or **models** of the phenomena in question. In this book the first of these terms will be preferred, to avoid confusion with a different, technical, use of the term "model" (in the sense of an interpretation of a logical theory) to be introduced in later chapters. For example, phrases (roughly speaking, multi-word expressions, including sentences) are often represented as (mathematical) *trees*; phonemes (roughly, minimal units of linguistic sound) as (mathematical) *graphs* of a certain kind (*feature structures*); the sequences of sounds that make up (the phonology of) words as (mathematical) *strings* of (representations of) phonemes; and linguistic meanings as (mathematical) *functions* of various kinds. (Note that it is typical for technical mathematical terms, such as *tree*, *string*, and *function*, to have other, nonmathematical meanings!)

In order to have a clear understanding of what these different kinds of mathematical entities are and why they are able to serve as linguistic representations, we will start out with an overview of **set theory**. Sets are basic mathematical entities whose existence is taken for granted by most mathematicians, and set theory begins with certain assumptions about them. Set theory is the workspace that most mathematicians work in; but more importantly for us, it is where the idealized representation of natural phenomena by linguists and other scientists is carried out. That is, sets are used to construct the represesentations of natural-language phenomena that linguistic theories talk about. In fact, all the kinds of linguistic representations mentioned above (trees, graphs, strings, and functions) are themselves sets.

1.2 Sets and Membership

We assume that there exist things which we call **sets**, and that there is a relationship, called **membership**, which either does or does not hold of any two sets. That is, if A is a set and B is a set, then either A **is a member of** B (written $A \in B$) or A is not a member of B (written $A \notin B$). There are many ways to say this. The members of a set are also called its **elements**, and instead of saying A is a member of B, we often say it **belongs to** B, or is **in** B, or is **contained in** B. Intuitively, sets can be thought of as something like collections, where the members are the things collected, or as invisible baskets, with the members being the things in the baskets. But set theory will never tell us what sets are; they are basic and cannot be reduced to, or explained in terms of, more basic things that are not sets. That is, they are the **unanalyzed primitives** of set theory.

We will make certain assumptions about how membership works based on these intuitions, and then try to ascertain what follows from them. These assumptions themselves, together with the facts that follow from them, constitute **set theory**. To be slightly more precise, they are *a* set theory, since some assumptions about how sets should work are controversial. In this chapter, we will make some of the most generally accepted of these assumptions explicit and consider some of their consequences. (In due course we will also consider some of the more controversial assumptions about how sets work.)

For the time being, we will state our assumptions about sets in English, and conduct our reasoning about what follows from these assumptions using intuitively valid English arguments called **informal proofs**. Later on we will see that it is possible to **formalize** the assumptions of set theory with the help of specialized symbolic systems (formal logics, such as predicate logic). In that case the formalized counterparts of the assumptions are called **axioms**; the additional formulas that follow from them are called **theorems**; and the formalized counterparts of the English arguments we make to justify these theorems are called **formal proofs**.

In fact, informal (but precise) natural-language reasoning is the norm amomg mathematicians and natural scientists. Usually they don't bother to formalize proofs unless they are studying proofs as mathematical objects in their own right. Later we will have occasion to do just that, for the (perhaps surprising) reason that linguistic expressions and their meanings can themselves be thought of as proofs in certain kinds of logical systems.

In ascertaining what follows from the assumptions we will make about sets and membership, the reasoning we use will be pretty much the same kind of reasoning we use when we draw conclusions from assumptions about ordinary things, e.g. kitchen appliances, furniture, people, etc. (There are, however, some ways of arguing and ways of expressing arguments that are typical of mathematical discourse, which we will look at more closely in the following chapter.) In practice, mathematics consists of more or less ordinary reasoning about not-so-ordinary things. The upshot, seemingly paradoxical, is that so-called formal linguistics is mostly done within *informal* set theory.

The resolution of the apparent paradox is that even informal set theory is more precise and explicit than linguistics that uses no set theory at all.

Now we're ready to start introducing our basic assumptions about sets, and considering some of their consequences.

1.3 Basic Assumptions about Sets

We are already assuming that there *are* sets, and that if A and B are sets, then either $A \in B$ or $A \notin B$. But to be able to *do* anything with sets, we need to make some assumptions about how they work. The assumptions we make in this chapter are the ones that are generally considered the most basic, intuitively plausible, and uncontroversial. Later we will add a few more (but not many more), including some that not all mathematicians are entirely comfortable with. We give each assumption a name, to make it easy to refer to.

Assumption 1 (Extensionality)**.** If A and B have the same members, then they are the same set (written $A = B$).

Note that in stating this assumption, we did not bother to mention that A and B are sets. That is because we've already established that we are now doing (informal) set theory, and in set theory, the only things being talked about are sets. Note also that we do not have to explicitly assume (though it is true) that if A and B do *not* have the same members, then they are *not* the same set (written $A \neq B$). That's because, if they were the same set, then everything about them, including what members they have, would be the same. This reasoning is no different than the kind of reasoning we would use to conclude (given that A and B are people), that if A and B do not have the same blood type, then they cannot be the same person: if they were the same person, everything about them—including their blood types—would be the same.

If every member of A is a member of B, we say that A is a **subset** of B, or, alternatively, that A is **included in** B), written $A \subseteq B$. Note that if $A \subseteq B$, B might have members that are not in A. On the other hand, if both $A \subseteq B$ and $B \subseteq A$, then it follows from Extensionality that $A = B$. If $A \subseteq B$ but $A \neq B$ then we say A is a **proper** subset of B, written $A \subsetneq B$.

Assumption 2 (Empty set)**.** There is a set with no members.

Note that from this assumption together with Extensionality we can conclude that the there is *only* one set with no members. We call this set the

empty set. The empty set is usually denoted by the symbol '$\emptyset$'. But later, we'll sometimes write it as '0' (the symbol for the number zero), because according to the most usual way of doing arithmetic within set theory (which we'll get to in Chapter 4), the number zero and the empty set are the same thing (in spite of what you may have been taught in other math classes!).

Now so far, we have no basis for concluding that there are any sets other than the empty set, not even sets with only one member. For example, we are not even able to make a valid argument that there is a set with $\emptyset$ as its only member. We remedy this situation by adding a few more assumptions, beginning with the following:

Assumption 3 (Pairing)**.** For any sets A and B, there is a set whose only members are A and B.

Note that, because of Extensionality again, there is *only* one set whose only members are A and B, which we write as $\{A, B\}$, Of course we could just as well have called this set $\{B, A\}$. More generally, we will notate any nonempty finite set by listing its members, separated by commas, between curly brackets, in any order. (In Chapter 5, we'll get clear about what we mean when we say a set is 'finite', but for now we'll just rely on intuition). Notice that nothing rules out the possibility that A and B are the same set, so it follows from pairing that for any set A there is a set whose only member is A, namely $\{A, A\}$. Of course, once we realize this, then we might as well just call it $\{A\}$ rather than $\{A, A\}$: repetitions inside the curly brackets don't make any difference because for any given set, either A is a member of it or it isn't; it doesn't make any sense to talk about *how many times* one set is a member of another.

A set with only one member is called a **singleton**. A special case of singleton sets is the set $\{0\}$ whose only member is 0. This set is also called 1, because according to the usual way of doing arithmetic within set theory, it is the same as the number one. Going one step further, we can use Pairing again to form the set $\{0, 1\}$, also known as 2. There is a general pattern here, which we will explain in Chapter 4.

Assumption 4 (Union)**.** For any set A, there is a set whose members are those sets which are members of (at least) one of the members of A.

Once again, Extensionality ensures the uniqueness of such a set, which is called the **union** of A, written $\bigcup A$. As a special case, if $A = \{B, C\}$, then $\bigcup A$ is the set each of whose members is in either B or C (or both). This set is usually written $B \cup C$. Note that in general this is not the same thing as $\{B, C\}$!

For any set A, the **successor** of A, written $s(A)$, is the set $A \cup \{A\}$. That is, $s(A)$ is the set with the same members as A, except that A itself is also a member of $s(A)$.[1] For example, 1 is the successor of 0, and 2 is the successor of 1.

Assumption 5 (Powerset)**.** For any set A, there is a set whose members are the subsets of A.

Yet again, Extensionality guarantees the uniqueness of such a set. We call it the **powerset** of A, written $\wp(A)$. It's important to realize that $\wp(A)$ is usually not the same set as A. That's because usually the subsets of a set are not the same as the members of the set. For example, 0 is a subset of 0 (in fact, every set is a subset of itself), but obviously 0 is not a member of 0 (since 0 is the empty set).

1.4 Russell's Paradox and Separation

Why do we need the powerset assumption? Why don't we just *define* $\wp(A)$ to be the set of all subsets of A? The answer is that the other assumptions we have made so far do not seem to enable us to conclude that there actually *is* such a set. More generally, whenever one says "the set of all sets such that blah-blah-blah", there is no guarantee that the assumptions one has made about sets enable one to conclude that there actually is a set meeting that description. That may seem counterintuitive, but, perhaps surprisingly, there is a knockdown argument that there is no such guarantee, which was discovered by the philosopher and mathematician Bertrand Russell.[2]

The argument runs as follows. Consider the description "the set of all sets which are not members of themselves." Suppose for a moment there were such a set, called R. Then would R be member of R? Well, either it is or it isn't. In the first case, we see right away that R cannot be a member of R. And in the second case, we see right away that R must be a member of R. Either way, we arrive at a contradiction, and so our temporary assumption that there is a set whose members are the sets which

[1] Nothing we have said rules out the possibility that $A \in A$, in which case $A = s(A)$. However, the most widely used set theory (called **Zermelo-Fraenkel** set theory) includes an assumption (called **Foundation**) which does rule out this possibility. We will not assume Foundation in this book.

[2] Russell made this argument in a famous letter written in 1902 to Gottlob Frege, another philosopher and mathematician, whose accomplishments include the invention of predicate calculus and of modern linguistic semantics.

are not members of themselves must have been false. This argument is called **Russell's Paradox**.

Russell's Paradox shows that, in general, we cannot assume that, for any set description, we can take for granted the existence of a set meeting that description. However, there is a more cautious assumption that proves to be extremely useful and which so far has not been shown to result in paradox.

Assumption 6 (Separation)**.** If A is a set and $P[x]$ is a condition on x (where x is a variable that ranges over sets), then there is a set, written $\{x \in A \mid P[x]\}$, whose members are all the x in A that satisfy $P[x]$.

Separation is so-called because, intuitively, we are separating out from A some members that are special in some way, and collecting them together into a set. We call Separation an assumption, but to be more precise it is an assumption **schema**: for each condition $P[x]$, we get a different separation assumption. For the moment we remain deliberately vague about what we mean by "a condition on x". (We'll clear this up in due course when we formalize set theory using predicate logic.) For the moment, the easiest way to get an idea of what we mean by a condition on x is to look at some examples.

First, suppose we have two sets A and B. Then by taking $P[x]$ to be the condition $x \in B$, Separation guarantees the existence of the set consisting of those members of A which are also in B. This set is called the **intersection** of A and B, written $A \cap B$. A and B are said to **intersect** if $A \cap B$ is non-empty; otherwise they are said to be **disjoint**. A set is called **pairwise disjoint** if no two distinct members of it intersect.

Second, by taking $P[x]$ to be the condition $x \notin B$, Separation guarantees the existence of the set consisting of those members of A which are *not* in B. This set, called the **complement of** B **relative to** A, or the **set difference of** A **and** B, is written $A \setminus B$.

A rather different application of Separation shows that there can be no set of all sets. For suppose there were; then applying Separation to it using the condition $x \notin x$, we would have the set of all sets which are not members of themselves. But as we already saw (Russell's Paradox), there can be no such set.

1.5 Ordered Pairs and Cartesian (Co-)Products

Sets do not embody any notion of order: $\{A, B\} = \{B, A\}$. But for linguistic applications, clearly we cannot escape from dealing with order! For example, we cannot describe the phonology of a word without specifying the order of the phonemes in it, not can we fully describe a sentence without specifying the order of its words. One way we might imagine responding to this need is simply to *assume* that for any A, B, there is an *ordered pair* $\langle A, B\rangle$. But what properties should we assume that ordered pairs have? Perhaps surprisingly, it turns out that once we have gotten clear about how ordered pairs should work, the assumptions we have already made about sets enable us to conclude that sets with the desired properties already exist. So we do not need to make any further assumptions in order to have ordered pairs.

In fact, the crucial property of ordered pairs, from which their usefulness derives, that they are *uniquely determined by their components*, in the sense that $\langle A, B\rangle = \langle C, D\rangle$ if and only if $A = C$ and $B = D$. Any way of defining the notion of ordered pair that results in their demonstrably having this property will suffice. The approach we will adopt here is the standard one, which is to define the **ordered pair** of A and B, written $\langle A, B\rangle$, to be the set $\{\{A\}, \{A, B\}\}$. A and B are called, respectively, the **first** and **second component** of $\langle A, B\rangle$. Notice that an ordered pair has either one or two members. In the first case, which arises when $A = B$, the ordered pair is just $\{\{A\}\}$, and both components are A. In the second case, the ordered pair has two members, one with one member and one with two members. In that case, the first component of the pair is the one that belongs to the set with one member, and the second component is the member of the two-member set which is not the member of the one-member set.

Given two sets A and B, it is also useful to have the notion of the **cartesian product** of A and B, written $A \times B$, which is supposed to be the set of all ordered pairs $\langle C, D\rangle$ such that $C \in A$ and $D \in B$. As it turns out, we do not have to *assume* that cartesian products exist, because their existence follows from Separation. (Showing this is left as an exercise.) A and B are called the **factors** of $A \times B$.

Having defined ordered pairs, we can now proceed to define an **ordered triple** to be an ordered pair whose first component is an ordered pair:

$$\langle A, B, C\rangle =_{\text{def}} \langle\langle A, B\rangle, C\rangle$$

and correspondingly the threefold cartesian product:

$$A \times B \times C =_{\text{def}} (A \times B) \times C$$

The definitions can be extended to quadruples, quintuples, etc. in the obvious way. Special cases of cartesian products, called cartesian **powers**, are ones where the factors are all the same set A. These are notated with parenthesized "exponents" (superscripts), e.g. $A^{(2)} = A \times A$, $A^{(3)} = A \times A \times A$, etc. Additionally, we define $A^{(1)}$ to be A, and we define $A^{(0)}$ to be 1. This last definition is less mysterious than it appears to be, but we will be in a better position to explain the motivation for it a little later. (It is actually closely related to the reason that $n^0 = 1$ in arithmetic, but for some readers, that may seem equally mysterious.)

Less well known than cartesian product, but also important in some of our applications, is the notion of the **cartesian coproduct**, also called the **disjoint union**) of A and B, written $A + B$. This is defined as $(\{0\} \times A) \cup (\{1\} \times B)$, the set of all ordered pairs $\langle C, D \rangle$ such that either $C = 0$ and $D \in A$ or $C = 1$ and $D \in B$. A and B are called the **cofactors** of $A + B$.

Intuitively, $A + B$ is the union of two sets, "copies" of A and B respectively, and these copies are disjoint, even if A and B are not. As with cartesian products, there is a straightforward extension to more than two cofactors. For the case of identical cofactors (called cartesian **copowers**), there does not seem to be a standard notation; here we write $A_{(n)}$, which, intuitively, is the union of n pairwise disjoint copies of A. So it should not come as much of a surprise that $A_{(1)}$ is defined to be A and $A_{(0)}$ is defined to be 0.

Chapter 2

Mathese

2.1 Introduction

Mathematicians (well, English-speaking ones, anyway) talk and write about things logical and mathematical (including set theory and anything they construct inside it) in a mixture of ordinary colloquial English and a special purpose dialect of English, which we will refer to as **Mathese**. Mathese is intended to avoid the ambiguity, vagueness, and imprecision of much ordinary colloquial English. It is a good idea to get into the habit of judiciously using Mathese when writing about formally rigorous linguistic theory for an audience with a reasonable degree of mathematical sophistication; e.g. when writing up problem sets for this course. (Alert: it is every bit as important *not* to write this way for a general linguistic audience!) Of course, unless you have an unusually strong mathematical background, it takes some time to get the hang of Mathese, so we will not require immediate mastery; and of course it's also okay to use ordinary English as long as the meaning is completely clear.

In its most basic form, all Mathese has is a few "logicky" expressions and some basic predicates for talking about set membership and equality. Fortunately, it's permissible to add new predicates and names to the language as needed, as long as you take care to *define* them in terms of expressions that are already in the language, as will be explained below. (Without such abbreviations, Mathese quickly becomes opaque to the point of sheer incomprehensibility.) There are also symbols for abbreviating expressions, which are mostly used in displayed calculations and inside of set descriptions; the abbreviations (especially the logicky ones) are usually *not* used in writing

Mathese prose (which is what you will usually be writing proofs in).[1]

2.2 "Logicky" expressions

2.2.1 Variables

These are upper- or lower-case roman letters (usually italicized in typing), with or without numerical subscripts, used roughly as pronouns or as names of arbitrary sets, e.g. x, y, x_0, x_1, X, Y, etc.

2.2.2 And

Mathese 'and' is abbreviated using the **conjunction** symbol $\wedge$. It is used mainly for combining sentences, as in:

S$_1$ and S$_2$. (Abbreviated form: $S_1 \wedge S_2$)

A sentence formed this way is called a **conjunctive** sentence. Here S_1 is called the **first conjunct** and S_2 is called the **second conjunct**. A conjunctive sentence is considered to be true if both conjuncts are true; otherwise it is false.

2.2.3 Or

Mathese 'or' is abbreviated using the **disjunction** symbol $\vee$. Like 'and', it is used mainly for combining sentences, as in:

S_1 or S_2. (Abbreviated form: $S_1 \vee S_2$)

A sentence formed this way is called a **disjunctive** sentence. Here S_1 is called the **first disjunct** and S_2 is called the **second disjunct**. Mathese *or* is **inclusive** disjunction, so that a disjunctive sentence is true if *either or both* of the disjuncts are true, and it is false otherwise.

2.2.4 Implies

Mathese 'implies' is abbreviated using one of the two **implication** symbols $\rightarrow$ or $\supset$. A synonym for 'implies' is 'if ... then ...'. It too is used for combining sentences, as in:

S_1 implies S_2. (Abbreviated forms: $S_1 \rightarrow S_2$ or $S_1 \supset S_2$)

[1] Later on, we'll introduce some formal languages, called *first-order languages*, which consist *entirely* of such symbols. By then, you'll have a good intutitive feeling for what such symbols mean. If you've taken a basic course in predicate logic, you'll already be familar with these.

A sentence formed this way is called a **conditional** or **implicative** sentence. Here S_1 is called the **antecedent** and S_2 is called the **consequent**. Caution: this does not mean quite exactly the same thing as *if* S_1 *then* S_2 in ordinary English. One difference is that a conditional Mathese sentence is considered to be true if the consequent is true, no matter whether the antecedent is true or false and even if the antecedent and the consequent seem to have nothing to do with each other, e.g.

If there does not exist a set with no members, then $0 = 0$.

is true. Another difference is that a conditional Mathese sentence is considered to be true if the antecedent is false, no matter whether the consequent is true or false, e.g.

If $0 \neq 0$ then $1 \neq 1$.

is true!

2.2.5 If and only if

Mathese 'if and only if', usually written simply as 'iff', is abbreviated using the **biimplication** symbol $\leftrightarrow$. It is used to combine sentences as in:

S_1 iff S_2. (Abbreviated form: $S_1 \leftrightarrow S_2$)

A sentence of this form is called a **biconditional**. S_1 iff S_2 can be thought of as shorthand for:

S_1 implies S_2, and S_2 implies S_1.

Consequently, a sentence of this form is considered to be true if either (1) both S_1 and S_2 are true, or (2) both S_1 and S_2 are false. Otherwise, it is false.

2.2.6 It is not the case that

Mathese 'it is not the case that' is abbreviated using one of the two **negation** symbols $\neg$ or $\sim$. It is placed before a sentence in order to negate it, as in:

It is not the case that S. (Abbreviated forms: $\neg$S or $\sim$ S)

A sentence of this form is called a **negative** sentence. Here S is called the **scope** of the negation. Unsurprisingly, a negative sentence is considered to be true if the scope is false, and false if the scope is true. For any sentence S, the sentence *it is not the case that* S is called the **negation** of S, or, equivalently, the **denial** of S.

Note that often, the effect of negation with *it is not the case that* can be achieved by ordinary English **verb negation**, which (simplifying slightly) involves replacing the finite verb (the one that agrees with the subject) V with 'does not V' if V is not an auxiliary verb (such as *has* or *is*), or negating V with a following *not* or *-n't* if it *is* an auxiliary. Thus, for example, these pairs of sentences are equivalent (express the same thing):

It is not the case that 2 belongs to 1.
2 does not belong to 1.

It is not the case that 1 is empty.
1 isn't empty.

But negation by *it is not the case that* and verb negation cannot be counted on to produce equivalent effects if the verb is in the scope of a quantifier (see following two sections). For example, these are not equivalent:

It is not the case that for every x, x belongs to x.
For every x, x doesn't belong to x.

For the first is clearly true (for example, 0 doesn't belong to 0), but the truth of the second cannot be determined on the basis of the assumptions in Chapter 1, and in fact different ways of adding further set-theoretic assumptions resolve the issue in different ways.

Note that for predicates with an abbreviatory symbol, such as *equals* ($=$) and *belongs to* ($\in$), the effect of verb negation is accomplished by a diagonal slash, e.g. $\neq$ 'is not equal to', $\notin$ 'is not a member of'.

2.2.7 For all

Mathese 'for all', abbreviated by the **universal quantifier** symbol $\forall$, forms a sentence by combining first with a variable and then with a sentence, as in:

For all x, S (abbreviated form: $\forall x$S).

The variable x is said to be **bound** by the quantifier, and the sentence S is called the **scope** of the quantifier. Synonyms of 'for all' include 'for each', 'for every', and 'for any'. Usually the bound variable also occurs in the scope; if it doesn't, then the quantification is said to be **vacuous**.

A sentence formed in this way is said to be **universally quantified**, or simply **universal**.

As long as we are using Mathese only to talk about set theory, we can assume that the bound variable in a universal sentence ranges over all sets, that is, 'for all x' is implicitly understood as 'for all sets x'.

However, often we want to universally quantify not over *every* set, but just over the sets that satisfy some condition on x, $\mathrm{S}_{1}[x]$. Then we say:

For every x with $S_1[x]$, $S_2[x]$.

This is understood to be shorthand for

For every x, $S_1[x]$ implies $S_2[x]$. (Abbreviated form: $\forall x(S_1[x] \to S_2[x])$)

If such a sentence is true, then we say that $S_1[x]$ is a **sufficient condition** for $S_2[x]$, or, equivalently, that $S_2[x]$ is a **necessary condition** for $S_1[x]$. A special case of this is that a sentence of this format is true if, no matter what x is, $S_1[x]$ is false. Such a sentence is said to be **vacuously true**. For example, the sentence

For every x with $x \neq x$, $x = 2$.

is (vacuously) true.

If a universal sentence of the form

For every x, $S_1[x]$ iff $S_2[x]$

(i.e. whose scope is a biconditional) is true, then we say $S_1[x]$ is a **necessary and sufficient condition** for $S_2[x]$.

2.2.8 There exists ... such that

Mathese 'there exists ... such that', abbreviated by the **existential quantifier** symbol $\exists$, forms a sentence by combining first with a variable and then with a sentence, as in:

There exists x such that S (abbreviated form: $\exists x$S).

The variable x is said to be **bound** by the quantifier, and the sentence S is called the **scope** of the quantifier. Synonyms of 'there exists ... such that' include 'for some' and 'there is a(n) ... such that'. Usually the bound variable also occurs in the scope; if it doesn't, then the quantification is said to be **vacuous**.

A sentence formed in this way is said to be **existentially quantified**, or simply **existential**.

As long as we are using Mathese only to talk about set theory, we can assume that the bound variable in an existential sentence ranges over all sets, that is, 'there exists x' is implicitly understood as 'there exists a set x'.

However, often we want to existentially quantify not over *every* set, but just over the sets that satisfy some condition $S_1[x]$. Then we say:

There exists x with $S_1[x]$, such that $S_2[x]$.

This is understood to be shorthand for

There exists x such that $S_1[x]$ and $S_2[x]$.

(Abbreviated form: $\exists x(S_1[x] \land S_2[x])$)

Note here the use of parentheses for disambiguation. Without the parentheses, it would be hard to be sure whether the scope of the quantifier was the conjunctive sentence or just its first conjunct. This is a common device in Mathese. Both round and square parentheses can be used, and multiple sets of parentheses can be used in the same sentence.

If a sentence contains variables which are not bound by any quantifier, those variables are called **free**. A sentence is called **closed** if it has no free variables, and **open** otherwise. A sentence whose free variables are $x_0, \ldots, x_n$ is often called a **condition** on $x_0, \ldots, x_n$. The number of free variables in a condition is called its **arity**. Thus conditions might be **nullary** (no free variables, i.e. a closed sentence), **unary** (one free variable), **binary** (two free variables), **ternary** (three free variables), etc.

2.2.9 There exists unique . . . such that

In Mathese, 'there exists unique . . . such that' (abbreviated form: $\exists! x$) combines first with a variable, then with a sentence, as in:

There exists unique x such that S. (Abbreviated form: $\exists! x$ S)

This is understood to be shorthand for:

$\exists x(S[x] \land \forall y(S[y] \to (y = x)))$

2.3 Defining Predicates

At the outset, the only predicates in Mathese are *equals* (abbreviated =) or synonyms such as *is the same as* or *is identical to*, and *is a member of* (abbreviated $\in$) or synonyms such as *belongs to* or *is an element of.* But we can *define* new predicates in terms of these and other predicates which have already been defined. The **arity** of a defined predicate is the arity of the condition that is used to define it. For example, we define "x is **empty**" to mean $\forall y(y \notin x)$, and "x is a **singleton**" to mean $\exists! y(y \in x)$; these are unary predicates. In "x **is a subset** of y" (abbreviation: $x \subseteq y$), $\subseteq$ is a binary predicate defined by the condition $\forall z(z \in x \to z \in y)$.

2.4 Defining Names

If we can prove (i.e. provide a persuasive valid argument based only on our assumptions about set theory and other things that have already been proved) that there exists a unique set x such that $\mathrm{S}[x]$, where $\mathrm{S}[x]$ is some condition on x, then we permit ourselves to bestow a name on that set. For example, it is easy to show that there is a unique set x such that x is empty. (The existence part of the proof is by the Empty Set assumption, and the uniqueness part of the proof is an application of Extensionality.) In this case, as we saw in Chapter 1, the set in question is named $\emptyset$ (read "the empty set").

2.5 Defining Functional Names

Often we can show that for any set y, there exists a unique set x satisfying some condition $\mathrm{S}[x, y]$. In such cases, we permit ourselves to introduce a **functional name**, which is basically a scheme which, for each y, provides a name for the unique set x such that $\mathrm{S}[x, y]$. To make an analogy with real life: obviously everybody has a mother, so we can use the functional name **y's mom** to refer to the unique individual x such that x is a mother of y, no matter who y is. Returning to sets, it is easy to prove that for any set y, there is a unique set x such that y is the only member of x. This justifies introducing the functional name $\mathsf{singleton}(y)$, abbreviated $\{y\}$. Likewise, we introduce the functional name $\mathsf{successor}(y)$, abbreviated $s(y)$ which, for each set y, names the unique set x that satisfies the binary condition $x = y \cup \{y\}$.

This naming convention extends to names that depend on more than one variable. Again, to take a real-life example, we might introduce the functional name **x's seniority over** y: for any two individuals x and y this is defined to be the number of days (rounded off) from x's birthdate to y's birthdate (this is a negative integer if y's birthdate precedes x's). The general principle is that if, for some positive natural number n and some $(n + 1)$-ary condition $\mathrm{S}[x_0, \ldots x_n]$ we can prove

$$\forall x_1 \ldots \forall x_n \exists! x_0 \mathrm{S}[x_0, \ldots x_n]$$

then we are allowed to make up a functional name $\mathsf{name}(x_1, \ldots, x_n)$ which for each choice of values for the n variables $x_1, \ldots, x_n$ provides a name for the unique set which satisfies the condition for that choice of values.

Chapter 3

Relations and Functions

3.1 Relations

Intuitively, a relation is the sort of thing that either does or does not hold between certain things, e.g. the love relation holds between Kim and Sandy just in case Kim loves Sandy, and the less-than relation holds between two natural numbers A and B just in case $A < B$. How should we represent relations mathematically if sets are all we have to work with? A simple-minded first pass might be to represent the love relation as the set of all pairs $\{A, B\}$ such that A and B are two people such that A loves B. (Actually, A and B would not be people at all, but rather certain sets that we have chosen as theoretical standins for (representations of) people: remember that the only things in our mathematical workspace are sets!) Unfortunately, this is too simple, since, for example, we are left with no way to represent unrequited love: what if Kim loves Sandy but Sandy does not love Kim?

A more promising approach is to represent love as the set of *ordered pairs* $\langle A, B\rangle$ such that A loves B. Of course nobody is under the illusion that a set of ordered pairs is the answer Cole Porter had in mind when he wrote *What is this Thing Called Love?* It is what a formal semanticist would call the *extension* of the love relation. (The appropriate way to mathematically represent the actual love relation, as opposed to its extension, is a question we will turn to later when we consider how to represent linguistic meaning.) To take a less vexing example, we can consider the relation $\subseteq_U$ of set inclusion restricted to the subsets of a given set U to be the following set of ordered pairs:[1]

[1]Please note that on the right-hand side of the following definition, we are making use of a commonplace notational convention whereby $\{\langle x, y\rangle \in A \times B \mid \phi\}$ abbreviates

$$\subseteq_U =_{\text{def}} \{\langle A, B\rangle \in \wp(U) \times \wp(U) \mid A \subseteq B\}$$

More generally, we now *define* the notion of relation as follows: a **relation** between A and B is a subset of $A \times B$. Equivalently, it is a set of ordered pairs whose first and second components are in A and B respectively. Equivalently, it is a member of $\wp(A \times B)$. In the special case where $A = B$, we speak of a relation **on** A. For example, $\subseteq_A$ is a relation on $\wp(A)$. But note: according to the way we have defined the notion of a relation, there is no $\subseteq$ relation! (Explaining why is left as an exercise.) As a matter of notation, we usually write $a\ R\ b$ to mean that the ordered pair $\langle a, b\rangle$ is in the relation R; that is, $a\ R\ b$ is just another way to say $\langle a, b\rangle \in R$.

An important special case arises when $A = B$ and the relation is

$$\mathsf{id}_A =_{\text{def}} \{\langle x, y\rangle \in A \times A \mid x = y\}$$

This relation is called the **identity** relation on A.

If R is a relation from A to B, the **inverse** of R is the relation from B to A defined as follows:

$$R^{-1} =_{\text{def}} \{\langle x, y\rangle \in B \times A \mid y\ R\ x\}$$

For example, suppose $\leq$ is the standard order on the natural numbers (to be defined precisely later); its inverse is the relation $\geq$. And the inverse of the (extension) of the love relation is the is-loved-by relation. It is easy to see that for any set A,

$$\mathsf{id}_A{}^{-1} = \mathsf{id}_A$$

and that for any relation R,

$$(R^{-1})^{-1} = R$$

.

As we have seen, a relation is defined as a subset of a cartesian product $A \times B$. More precisely, this should have been called a **binary** relation. Likewise, we can define a **ternary relation** among the sets A, B, and C to be a subset of the threefold cartesian product $A \times B \times C$; thus a ternary relation is a set of ordered triples. For $n > 3$, n-fold cartesian products and n-ary relations are defined in the obvious way.

$\{z \in A \times B \mid \exists x \exists y(\phi \wedge z = \langle x, y\rangle)\}$.

Recall that a **cartesian power** is a cartesian product all of whose factors are the same, e.g. $A^{(3)} = A \times A \times A$; that $A^{(1)} = A$; and that $A^{(0)} = 1$. Correspondingly, a **unary relation** on A is just a subset of A, and a **nullary relation** on A is a subset of 1, i.e. either 1 or 0.

Suppose R is a relation from A to B and S is a relation from B to C. Then the **composition** of S and R is the relation from A to C defined by

$$S \circ R =_{\text{def}} \{\langle x, z\rangle \in A \times C \mid \exists y \in B(x\ R\ y \wedge y\ S\ z)\}$$

It is easy to see that if R is a relation from A to B, then

$$\mathsf{id}_B \circ R = R = R \circ \mathsf{id}_A$$

Suppose R is a relation from A to B. Then the **domain** and **range** of R are defined as follows:

$$\mathsf{dom}(R) =_{\text{def}} \{x \in A \mid \exists y \in B(x\ R\ y)\}$$

and

$$\mathsf{ran}(R) =_{\text{def}} \{y \in B \mid \exists x \in A(x\ R\ y)\}$$

respectively.

3.2 Functions

A relation F between A and B is called a **(total) function** from A to B provided for every $x \in A$, there exists a unique $y \in B$ such that $x\ F\ y$).
In that case we write $F\colon A \to B$. This is often expressed by saying that F **takes** members of A as **arguments** and **returns** members of B as **values** (or, alternatively, **takes its values** in B). Obviously,

$$\mathsf{dom}(F) = A$$

For each $a \in \mathsf{dom}(F)$, the unique b such that $a\ F\ b$ is called the **value** of F **at** a, written $F(a)$. Equivalently, we say F **maps** a to b, written $F\colon a \mapsto b$.

In formal semantics, linguistic meanings are often represented as functions of certain kinds. For example, it is fairly standard to represent declarative sentence meanings as functions from a set W of "possible worlds" (which themselves are taken to be representations of different possible ways the world might be) to the set 2 (i.e. $\{0, 1\}$); here 1 and 0 are identified, respectively, with the intutitive notions of truth and falsity. Not quite so straightforward is the use of function terminology by syntacticians, for example referring to the subjects and complements of a verb as its "grammatical arguments". If a verb were really a function, then what would its domain and codomain be? In due course we'll look into the motivation for talking about verbs and other linguistic expressions as if they were functions.

Note that for any set A, the identity relation id_A is the function from A to A such that

$$\mathsf{id}_A(a) = a$$

for every $a \in A$. In some linguistic theories, identity functions serve as the meanings of "referentially dependent" expressions such as pronouns and gaps.

It is not hard to see (after some reflection) that a relation R from A to B is a function from A to B iff

$$R \circ R^{-1} \subseteq \mathsf{id}_B$$

and

$$\mathsf{id}_A \subseteq R^{-1} \circ R$$

We note here a confusing though standard bit of terminology. Given a function $F : A \to B$, we often call B the **codomain** of F. What is confusing is that if B is a proper subset of some other set B', then clearly also $F: A \to B'$; but then B' must be the codomain of F! Evidently the notion of codomain of a function is not well-defined. Technically, we can clear up this confusion by defining a **(set theoretic) arrow** from A to B to be an ordered triple $f = \langle A, B, F\rangle$, where $F: A \to B$. Now we can unambiguously refer to A and B as the domain and codomain of f, respectively; F is called the **graph** of f. The point is that two distinct arrows can have the same domain and the same graph but different codomains. Thus when we speak (loosely) of a function $F: A \to B$ having B as its codomain, we are really talking about the arrow $\langle A, B, F\rangle$. Having called attention to this abuse of language, we will persist in it without further comment.

For any sets A and B, the **exponential** from A to B is the set of arrows from A to B. This is written B^A, read "B to the A". An alternative notation is $A \Rightarrow B$, read "A into B". Note for any set A there is a unique function $\Diamond_A: \emptyset \to A$ (what is it?) and a unique function $\Box_A: A \to 1$ (what is it?).

A relation F between A and B is called a **partial function** from A to B provided there is a subset $A' \subseteq A$ such that F is a (total) function from A' to B.

For $n \geq 0$, an n-ary **(total) operation** on a set A is a function from $A^{(n)}$ to A. So a unary operation on A is just a function from A to itself, and a nullary operation on A is a function from 1 (i.e. $\{0\}$) to A. It is easy to see that there is a one-to-one correspondence between A and A^{1}, with each $a \in A$ corresponding to the function from 1 to A that maps 0 to a.

Suppose $F\colon A \to B$. Then F is called:

injective, or **one-to-one**, or an **injection**, if it maps distinct members of A to distinct members of B;

surjective, or **onto**, or a **surjection**, if $\mathsf{ran}(F) = B$; and

bijective, or **one-to-one and onto**, or a **bijection**, or a **one-to-one correspondence**, if it is both injective and surjective.

An important special case of injective functions are defined as follows: if $A \subseteq B$, then the function $\mu_{A,B}\colon A \to B$ that maps each member of A to itself is called the **embedding** of A into B. Note that $\mu_{A,B}$ has the same graph as id_A, but possibly a larger codomain.

Also injective are the functions ι_1 and ι_2, called **canonical injections**, from the cofactors A and B of a coproduct $A+B$ into the coproduct, defined by $\iota_1(a) = \langle 0, a\rangle$ and $\iota_2(b) = \langle 1, b\rangle$ for all $a \in A$ and $b \in B$. Standard examples of surjections are the **projections** π_1 and π_2 of a product $A \times B$ onto its factors A and B respectively, defined by $\pi_1(\langle a, b\rangle) = a$ and $\pi_2(\langle a, b\rangle) = b$ for all $a \in A$ and $b \in B$.

Suppose $A \subseteq B$. Then the function $\chi_A\colon B \to 2$ such that, for each $b \in B$, $\chi_A(b) = 1$ iff $b \in A$ is called the **characteristic** function of A (relative to B). It is easy to see that there is a bijection from $\wp(B)$ to $B \Rightarrow 2$ that maps each subset of B to its characteristic function.

Since functions are relations, the definition of composition for relations makes sense when the two relations being composed are functions. Thus if $F\colon A \to B$ and $G\colon B \to C$, then $G \circ F\colon A \to C$, and for every $x \in A$,

$$G \circ F(x) = G(F(x))$$

It is not hard to see that[2]

$$G \circ F = \{\langle x, z\rangle \in A \times C \mid \exists y \in B(y = F(x) \wedge z = G(y))\}$$

.

[2]Please note that in the set description on the right-hand side of the following equation, we make use of a commonplace notational convention whereby $\exists y \in B\phi$ abbreviates $\exists y(\phi \wedge y \in B)$.

For example, taking it one faith for the moment that there is a set ω whose members are precisely the natural numbers, and that the familiar (binary) arithemetic operations (addition, multiplication, and exponentiation) have been given satisfactory set-theoretic definitions (we will make this precise in due course), let F and G be the functions from ω to ω such that

$$F(x) = x^2$$
$$G(x) = x + 2$$

for all $x \in \omega$. Then $G \circ F$ is given by

$$G \circ F(x) = x^2 + 2$$

Suppose once again that $F : A \to B$ and $G : B \to C$, and suppose moreover that $H : C \to D$. Then it is not hard to see that

$$H \circ (G \circ F) = (H \circ G) \circ F$$

Since functions are relations, the following hold for any function $F : A \to B$:

$$\mathsf{id}_B \circ F = F = F \circ \mathsf{id}_A$$
$$F \circ F^{-1} \subseteq \mathsf{id}_B$$
$$\mathsf{id}_A \subseteq F^{-1} \circ F$$

Additionally, it is easy to see that F is surjective iff

$$F \circ F^{-1} = \mathsf{id}_B$$

and F is injective iff

$$\mathsf{id}_A = F^{-1} \circ F$$

As a special case, if F is a unary operation on A, then

$$\mathsf{id}_A \circ F = F \circ \mathsf{id}_A = F$$

If in addition F is bijective, then the relation F^{-1} is also a unary operation on A, and

$$F \circ F^{-1} = F^{-1} \circ F = \mathsf{id}_A$$

Suppose $F : A \to B$, $A' \subseteq A$, and $B' \subseteq B$. Then the **restriction** of F to A' is the function from A' to B given by

$$F \restriction A' = \{\langle u, v\rangle \in F \mid u \in A'\}$$

Note that this is the same function as $F \circ \mu_{A',A}$. The **image** of A' by F is the set

$$F[A'] =_{\text{def}} \{y \in B \mid \exists x \in A'(y = F(x))\}$$

The **preimage** (or **inverse image**) of B' by F is the set

$$F^{-1}[B'] =_{\text{def}} \{x \in A \mid \exists y \in B'(y = F(x))\}$$

This is more simply described as

$$\{x \in A \mid F(x) \in B'\}$$

3.3 Special Kinds of Binary Relations

3.3.1 Properties of Relations

Here we collect some definitions for future reference. Throughout we assume R is a binary relation on A.

Distinct $a, b \in A$, are (R-)**comparable** if either $a \ R \ b$ or $b \ R \ a$; otherwise, they are **incomparable**. R is **connex** iff a and b are comparable for all distinct $a, b \in A$.

R is **reflexive** if $a \ R \ a$ for all $a \in A$ (i.e. $\text{id}_A \subseteq R$). R is **irreflexive** if $a \not R \ a$ for all $a \in A$ (i.e. $\text{id}_A \cap R = \emptyset$).

R is **symmetric** if $a \ R \ b$ implies $b \ R \ a$ for all $a, b \in A$ (i.e. $R = R^{-1}$). R is **asymmetric** if $a \ R \ b$ implies $b \not R \ a$ for all $a, b \in A$ (i.e. $R \cap R^{-1} = \emptyset$). R is **antisymmetric** if $a \ R \ b$ and $b \ R \ a$ imply $a = b$ for all $a, b \in A$ (i.e. $R \cap R^{-1} \subseteq \text{id}_A$). Thus asymmetry is a special case of antisymmetry; more specifically, a relation is asymmetric iff it is both antisymmetric and irreflexive.

A relation R is **transitive** if $a \ R \ b$ and $b \ R \ c$ imply $a \ R \ c$ for all $a, b, c \in A$ (i.e. $R \circ R \subseteq R$). R is **intransitive** if $a \ R \ b$ and $b \ R \ c$ imply $a \not R \ c$ for all $a, b, c \in A$ (i.e. $(R \circ R) \cap R = \emptyset$).

3.3.2 Orders and Preorders

A **preorder** is a reflexive transitive relation; and an **order** is an antisymmetric preorder. On of the most useful orders overall is the subset relation $\subseteq_A$ on $\wp(A)$. In linguistic applications, as we will see later on, one of the most widely used orders is the *dominance* order on the nodes of a *tree*, used in many syntactic theories to represent the (putative) constituent structure of a linguistic expression. (But not in all syntactic theories; for example, in the family of syntactic theories known as *categorial grammar*, the notion of constituent plays little or no role.)

In many approaches to formal semantics of natural languages, the representations of declarative sentence meanings (usually called *propositions*) are preordered by a relation called *entailment*. Without getting technical at this point, if p and p' are the propositions expressed by two natural-language sentence utterances S and S', p entails p' just in case, no matter what the world is like, if S is true with the world that way, then so is S'. In order to have a formal theory of this, we will have to have a way of set-theoretically representing sentence utterances, propositions, and possible ways the world might be. Considerable care is needed here, since one and the same sentence can express different propositions depending on the context of utterance, and utterances of different sentences can express the same proposition. A controversial issue here is whether or not the entailment relation is antisymmetric. In other words: if two sentences always agree in truth value no matter what the world is like, then must they express the same proposition? We will take up these and related issues in due course.

Let R be a preorder on A, $S \subseteq A$, and $a \in S$. Then a is **maximal** in S if $a\ R\ b$ implies $b\ R\ A$ for every $b \in S$; a is **minimal** in S if $b\ R\ a$ implies $a\ R\ b$ for every $b \in S$. a is **greatest** in S if $b\ R\ a$ for every $b \in S$; a is **least** in S if $a\ R\ b$ for every $b \in S$; and a is a **top** (respectively, **bottom**) if it is greatest (respectively, least) in A. If there is a unique top, it is written $\top_R$. If there is a unique bottom, it is written $\bot_R$. Clearly, if S has any greatest (least) elements, then they (and only they) are maximal (minimal) elements of S.

When the preorder in question is being used to represent the entailment relation on the propositions in a model-theoretic semantics of a natural language, then those propositions (if any) which are true independently of how things are (such propositions are called *necessary truths*) must be tops; and those propositions (if any) which are false no matter how things are (such propositions are called *necessary falsehoods*) must be bottoms. (Why?) Propositions which are neither necessary truths nor necessary falsehoods are called *contingent*; their truth or falsity depends on how things are.

Now suppose the preorder R is also antisymmetric (i.e. it is an order). Then S can have at most one greatest (or least) member; in particular, there can be at most one top (or bottom). If a is greatest (or least) in S, then it is the unique maximal (or minimal) element of a.

But it is possible (even if R is antisymmetric) for a to be the unique maximal (or minimal) element of S without being the greatest (or least) element in S. For that matter, S can have more than one maximal (or minimal) element without any of them being greatest (or least). It's an instructive exercise to try to verify the foregoing assertions by constructing suitable examples.

In a connex preorder, for a to be maximal (minimal) in S is the same thing as for a to be greatest (least) in S. A connex order is called a **chain**, a **total order**, or a **linear order**. A chain is called a **well-ordering** provided every non-empty subset of A has a least element. The standard example of a well-ordering is the standard ($\leq$) order on the natural numbers.

For linguists, the most familiar chains are the *linear precedence (LP)* orders that arise in the representation of the consituent structure (within linguistic theories that countenance such things) of a linguistic expression by an ordered tree, namely (1) the LP order on the daughters (immediate consituents) of a nonterminal node, and (2) the LP order on the preterminals. We will take a close look at the use of tree representations in syntax in Chapters 7 and 8.

3.3.3 Equivalence Relations

An **equivalence** relation is a symmetric preorder. If R is an equivalence relation, then for each $a \in A$ the (R-)**equivalence class** of a is

$$[a]_R =_{\text{def}} \{b \in A \mid a\ R\ b\}$$

Usually the subscript is dropped when it is clear from context which equivalence relation is in question. The members of an equivalence class are called its **representatives**. Note that the set of equivalence classes (written A/R and called the **quotient** of A by R) is a **partition** of A, i.e. it is pairwise disjoint and its union is A. It's easy to see that the function from A to A/R that maps each member of A to its equivalence class is a surjection. More generally, for any function $F \colon A \to B$, there is an equivalence relation $\equiv_F$, with two members of A being equivalent just in case F maps them to the same member of B.

If R is a preorder on A, the relation $\equiv_R$ defined by $a \equiv_R b$ iff both $a\ R\ b$ and $b\ R\ a$ is easily seen to be an equivalence relation. In the special case where R is the entailment relation between propositions in a semantic theory, this equivalence relation is called *truth-conditional equivalence*. Thus truth-conditionally equivalent propositions are true under exactly the same conditions. In semantic theories where entailment is taken to be antisymmetric, truth-conditionally equivalent propositions are identical.

Chapter 4

The Natural Numbers, Induction, and Recursive Definition

4.1 The Natural Numbers

In Chapter 1, we introduced 0 (aka $\emptyset$), its successor $1 = s(0) = 0 \cup \{0\} = \{0\}$, 1's successor $2 = s(1) = 1 \cup \{1\} = \{0, 1\}$, and 2's successor $3 = s(2) = 2 \cup \{2\} = \{0, 1, 2\}$. This is how the first four natural numbers are usually modelled within set theory; it's intuitively obvious that we could go on in the same way to model as many of the natural numbers as time would permit. Note that $0 \in 1 \in 2 \in 3 \in \cdots$ and $0 \subseteq 1 \subseteq 2 \subseteq 3 \cdots$. Is there a set consisting of *all* the natural numbers? The assumptions we made in Chapter 1 do not seem to enable us to draw this conclusion. It would be most useful to have such a set, but we are not yet quite in a position to add the assumption that there is a set whose members are precisely the natural numbers, since so far we haven't said what a natural number is! But we are about to.

A set s called **inductive** iff it has 0 as a member and has the successor of each of its members as a member. We then define a **natural number** to be a set which belongs to every inductive set. It is not hard to show that 0, 1, 2, and 3 are all natural numbers. But at this stage, for all we know, *every* set might be a natural number. After all, even though we defined what it means for a set to be inductive, at this point we don't know that there *are* any inductive sets! What if there weren't any? In that case, it's easy to see that indeed every set would be a natural number. And then, since (as we already know) there is no set of all sets, there could not be a set of all the natural numbers. So if we want there to be a set of all natural numbers, there better be at least one inductive set.

We now add to our assumptions about sets the following:

Assumption 7 (Natural Numbers)**.** There is a set whose members are the natural numbers.

By Extensionality, there can only be one such set. We call it ω. With the help of this assumption, it is now easy to prove the following two theorems:[1]

Theorem 4.1. *ω is inductive.*

Proof. Exercise. $\square$

Theorem 4.2. *ω is a subset of every inductive set.*

Proof. Exercise. $\square$

The relation $<$ (read **less than**) on ω is defined by $n < m$ iff $n \in m$, and the relation $\leq$ (read **less than or equal to**) by $n \leq m$ iff $n < m$ or $n = m$. (So $\leq$ is the reflexive closure of $<$.)[2] The terminology "less than or equal to" is justified, since in fact $\leq$ is an order, as we will show. In fact we will show more, namely that $\leq$ is a well-ordering (and in particular a linear order).

4.2 Induction and Recursive Definition

The following theorem is a corollary of the preceding one:

Theorem 4.3 (Principle of Mathematical Induction)**.** *The only inductive subset of ω is ω.*

Proof. Exercise. $\square$

[1] A **theorem** is just something important that we can prove. More generally, something that we can prove is usually called a **proposition**. (Note: this is a different use of the term *proposition* than in linguistic semantics, where it refers to the intepretation of a declarative sentence utterance.) So a theorem is an important proposition. A **lemma** is a proposition which is not so important in and of itself, but which is used in order to prove a theorem. And a **corollary** of a proposition is another proposition which is easily proved from it.

[2] Later we will be able to prove that, for any two natural numbers n and m, $n < m$ iff $n \subsetneq m$, and $n \leq m$ iff $n \subseteq m$.

The Principle of Mathematical Induction (PMI) is one of the mathematician's most important resources for proving theorems. It is applicable any time we want to prove that a condition $\phi[n]$ is true for every natural number n. The trick is to consider the set $\{n \in \omega \mid \phi[n]\}$ and show that it is inductive. To put it another way, we first prove $\phi[0]$ (this is called the **base case** of the proof) and then prove that, if we assume $\phi[k]$ for an arbitrary natural number k (the so-called **inductive hypothesis**), then $\phi[s(k)]$ follows (the so-called **inductive step**). By way of illustration, we prove the following:

Proposition 4.1. *Let* $\mathbf{suc}\colon \omega \to \omega$ *be the function that maps each natural number to its successor. Then* $\mathsf{ran}(\mathbf{suc}) = \omega \setminus \{0\}$.

Proof. Obviously $0 \notin \mathsf{ran}(\mathbf{suc})$. Let T be the set of all natural numbers that are either 0 or else the successor of some natural number. We must show that T is inductive, that is that (1) $0 \in T$ and (2) for each $n \in T, \mathbf{suc}(n) \in T$. But both of these are immediate consequences of the definition of T. □

Why do we persist in saying "$\mathbf{suc}(n)$" instead of "1+n"? Answer: because the operation of addition for natural numbers has not been defined yet. Yet it seems clear how addition works: for any natural number m, $m+0$ should be m; and if k is nonzero (so that it is the successor of some other natural number n), then $m+k$ should be the successor of $m+n$. That is, for each $m \in \omega$ we would like to *define* addition by the equations

$$m + 0 = m$$

and

$$m + \mathbf{suc}(n) = suc(m + n)$$

Definitions of this kind are called **recursive**. But how do we know recursive definitions make sense? The answer is provided by the Recursion Theorem, henceforth abbreviated RT:

Theorem 4.4 (RT)**.** *Let* X *be a set,* $x \in X$*, and* $F\colon X \to X$*. Then there exists a unique function* $h\colon \omega \to X$ *such that (1)* $h(0) = x$*, and (2) for every* $n \in \omega$*,* $h(\mathbf{suc}(n)) = F(h(n))$*.*

RT is not hard to prove, but the proof is a bit long. So we relegate it to an appendix, and turn straightaway to some applications.

4.3 Arithmetic

4.3.1 Addition

As our first application of RT, let's show that the informal recursive definition of addition given above actually makes sense.

To get started, suppose $m \in \omega$. We'll use RT to show there is a function A_m such that $A_m(0) = m$ and $A_m(\mathbf{suc}(n)) = \mathbf{suc}(A_m(n))$. The trick, as always when applying RT, is to find the right instantiations of X, x, and F. In the present case the happy choices are $X = \omega$, $x = m$, and $F = \mathbf{suc}$; with these choices, the function h whose unique existence is guaranteed by RT has just the properties we want for A_m. We then define the **addition** operation $+\colon \omega^{(2)} \to \omega$ such that, for all $m, n \in \omega$, $m + n =_{\text{def}} A_m(n)$. It follows from this definition that $m + 0 = m$ for all $m \in \omega$ and $m + \mathbf{suc}(n) = \mathbf{suc}(m + n)$ for all $m, n \in \omega$, as desired.

Theorem 4.5. *For every natural number n, $1 + n = \mathbf{suc}(n)$.*

Proof. Exercise. □

4.3.2 Multiplication

Turning next to multiplication, we first use RT to define multiplication by a fixed natural number m. We want a function M_m such that (1) $M_m(0) = 0$, and (2) for every $n \in \omega$, $M_m(\mathbf{suc}(n)) = m + M_m(n)$. To this end, we apply RT again, this time with $X = \omega$, $x = 0$, and $F = A_m$. We then define the **multiplication** operation $\cdot\colon \omega^{(2)} \to \omega$ such that $m \cdot n =_{\text{def}} M_m(n)$. So $m \cdot 0 = 0$ and $m \cdot (1 + n) = m + m \cdot n$, which is as it should be.

Theorem 4.6. *For every $n \in \omega$, $1 \cdot n = n$.*

Proof. Exercise. □

With more time and ambition, one can also prove the familiar Five Laws of Arithmetic (hereafter we omit the "$\cdot$" for multiplication):

1. Associativity of Addition: $m + (n + p) = (m + n) + p$
2. Commutativity of Addition: $m + n = n + m$
3. Distributivity of Multiplication Over Addition: $m(n + p) = mn + mp$
4. Associativity of Multiplication: $m(np) = (mn)p$

5. Commutavity of Multiplication: $mn = nm$

Yet another good exercise is to give a recursive definition of the **exponentiation** operation $m \star n = m^n$ and prove that the definition makes sense. Hint: define $m \star n$ to be $E_m(n)$ where $E_m(0) = 1$ and $E_m(\mathbf{suc}(n)) = m \cdot E_m(n)$. This establishes the first two of the following three general properties of exponentiation:

1. $m^{0} = 1$
2. $m^{1+n} = m(m^n)$
3. $m^{n+p} = (m^n)(m^p)$

Note that the second is a special case of the third, which is usually called the **Law of Exponents**.

4.3.3 The Infinitude of the Natural Numbers

Everyone knows that there is an infinite number of natural numbers, but what exactly does that mean? A set is called **finite** if it is in one-to-one correspondence with a natural number, and **infinite** otherwise. A set is called **Dedekind infinite** if it is in one-to-one correspondence with a proper subset of itself. On the basis of the assumptions we've made so far about sets, it's possible to prove (see Chapter 5) that any Dedekind-infinite set is infinite.[3] Since we already know that $\mathsf{ran}(\mathbf{suc}) = \omega \setminus \{0\}$, we could then show ω is infinite if we could show that $\mathbf{suc}\colon \omega \to \omega$ is injective. This is of course the case; a sketch of a proof follows.

First, we define a set A to be **transitive** iff every member of a member of A is itself a member of A. It is easy to see that all three of the following conditions on a set A are equivalent to transitivity:

1. $(\bigcup A) \subseteq A$;
2. every member of A is a subset of A; and
3. $A \subseteq \wp(A)$.

[3] To prove the converse, however, we need an additional assumption, viz. the Assumption of Choice (AC, Chapter 5). AC also enables us to prove that ω is the "smallest" infinite set, in the sense of being in one-to-one correspondence with a subset of any other infinite set.

The proof that **suc** is injective requires a couple of preliminary results:

Lemma 4.1. *If A is transitive, then $\bigcup s(A) = A$.*

Proof. We use the (easily proved) general fact about union that

$$\bigcup(x \cup y) = \left(\bigcup x\right) \cup \left(\bigcup y\right)$$

and reason as follows:

$$\begin{aligned}\bigcup s(A) &= \bigcup(A \cup \{A\}) \\ &= (\bigcup A) \cup (\bigcup\{A\}) \\ &= (\bigcup A) \cup A \\ &= A\end{aligned}$$

The last step follows from the fact that $\bigcup A \subseteq A$ for transitive A. $\square$

Lemma 4.2. *Every natural number is transitive.*

Proof. Exercise. $\square$

Theorem 4.7. **suc** *is injective.*

Proof. Suppose $\mathbf{suc}(m) = \mathbf{suc}(n)$. Then $\bigcup \mathbf{suc}(m) = \bigcup \mathbf{suc}(n)$. But m and n are transitive (by Lemma 4.2), so (by Lemma 4.1) $\bigcup \mathbf{suc}(m) = m$ and $\bigcup \mathbf{suc}(n) = n$. Therefore $m = n$. $\square$

As noted above, the infinitude of ω is a corollary of this.

4.3.4 The Well-Ordering of w

We now have the resources to establish that the relation $\leq$ on ω is an order, indeed a well-ordering (i.e. a chain such that every nonempty subset of ω has a least member). Given how obvious this seems, the argumentation required is surprisingly intricate and too long to reproduce in full detail here, so we content ourselves with an outline, including key lemmata and proof sketches.

Recall that by definition $m < n$ iff $m \in n$, and $m \leq n$ iff $m < n$ or $m = n$.

Theorem 4.8. *For all $n \in \omega$, $n = \{m \in \omega \mid m < n\}$.*

Proof. To show inclusion, suppose $m \in n$. Since ω is transitive, $m \in \omega$. Then $m < n$. To show the reverse inclusion, suppose $m < n$. Then by definition, $m \in n$. □

Lemma 4.3. *For all $m, n \in \omega$, $m < \mathbf{suc}(n)$ iff $m \leq n$.*

Proof. $m < \mathbf{suc}(n)$ iff $m \in \mathbf{suc}(n)$ iff $m \in n \cup \{n\}$ iff $m \in n$ or $m \in \{n\}$ iff ($m \in n$ or $m = n$) iff $m \leq n$. □

Lemma 4.4. *For all $m, n \in \omega$, $m \in n$ iff $\mathbf{suc}(m) \in \mathbf{suc}(n)$.*

Proof. For the only-if direction , assume $\mathbf{suc}(m) \in \mathbf{suc}(n)$. Then $\mathbf{suc}(m) < \mathbf{suc}(n)$, so by Lemma 4.3, $\mathbf{suc}(m) \leq n$, i.e. either $\mathbf{suc}(m) \in n$ or $\mathbf{suc}(m) = n$. If $\mathbf{suc}(m) \in n$, then $m \in \mathbf{suc}(m) \in n$, so $m \in n$ by transitivity. Otherwise $\mathbf{suc}(m) = n$; but $\mathbf{suc}(m) = m \cup \{m\}$, from which it follows easily that $m \in n$.

For the if direction, we use PMI. Let

$$T = \{n \in \omega \mid \forall m \in n, \mathbf{suc}(m) \in \mathbf{suc}(n)\}$$

It is sufficient to show that T is inductive. This is left as an exercise. □

Lemma 4.5. *For all $n \in \omega$, $n \notin n$.*

Proof. This is another inductive proof. Let $T = \{n \in \omega \mid n \notin n\}$ It suffices to show T is inductive. The base case is trivial, and the inductive step is an easy consequence of Lemma 4.4. □

Theorem 4.9. *$<$ is transitive, irreflexive, and connex.*

Proof. Transitivity follows readily from Lemma 4.2 and irreflexivity from Lemma 4.5. Connexity is proved inductively, by showing that the set $T = \{n \in \omega \mid \forall m \in \omega \mid m \neq n \rightarrow (n \in m \vee m \in n)\}$ is inductive; the inductive step of the proof appeals to both Lemma 4.4 and Lemma 4.3. □

As two easy consequences of this theorem, we have the following

Corollary 4.1. *For all $m, n \in \omega$, $m \in n$ iff $m \subsetneq n$.*

Corollary 4.2. *$\leq$ is a chain.*

And finally:

Theorem 4.10. *$\leq$ is a well-ordering.*

Proof. Suppose $A \subseteq \omega$ has no least element, It suffices to show $A = \emptyset$. To this end, let B be the set of all natural numbers n such that no natural number less than n belongs to A. All that is required is to show B is inductive. This is left as an exercise. (Hint: use Lemma 4.3 in the inductive step.) $\square$

4.4 Transitive Closure and Reflexive Transitive Closure

Let R be a binary relation on A. Then informally, the **transitive closure** of R, written R^+, is usually "defined" as follows: For all $n \in \omega$, recursively define $h(n)$ by $h(0) = \mathrm{id}_A$, $h(1) = R$, and $h(n+1) = h(n) \circ R$. Then $R^+ =_{\text{def}} \bigcup_{n>0} h(n)$. We leave as an exercise the formal justification of this definition using RT. Similarly, the **reflexive transitive closure** of R, written R^*, is $\bigcup_{n \in \omega} h(n)$. Note that $R^* = R^+ \cup \mathrm{id}_A$.

Lemma 4.6. *R^+ is transitive.*

Proof. Exercise. $\square$

Theorem 4.11. *The transitive closure of R is the intersection of all the transitive relations of which R is a subset, i.e.*

$$R^+ = \bigcap \{S \subseteq A^{(2)} \mid R \subseteq S \textit{ and } S \textit{ is transitive}\}$$

Proof. Exercise. $\square$

4.5 Hasse Diagrams

A **Hasse diagram** is a kind of textual (paper or blackboard) diagrammatic representation of a preorder $\sqsubseteq$ on a set A, made up of dots and straight line segments directly connecting two dots (here "directly" means there are no dots on the line segment other than the two being connected). The line segments are of two kinds: (1) nonhorizontal (i.e. either slanting or vertical) single line segments, and (2) horizontal double line segments. The interpretation is as follows: the dots represent the members of A; if (the dots representing) b and a are connected by a single nonhorizontal line segment and b is higher (on the page or board) than a, then $a \prec b$; and if a and b are connected by a horizontal double line segment, then a and b are 'tied', i.e. $a \sqsubseteq b$ and $b \sqsubseteq a$. (So if $\sqsubseteq$ is an order, there will be no horizontal double line segments.)

Any finite preorder can be represented by a Hasse diagram, but not every infinite one can. (There can be infinite Hasse diagrams, but there is not enough time to draw all of one! Sometimes the gist of an infinite Hasse diagram can be conveyed with judicious use of ellipsis ("and so on") dots, though.) For antisymmetric preorders (i.e. orders) the property of being representable by a Hasse diagram is easy to express precisely in set-theoretic terms: it is the property of being the reflexive transitive closure of its own covering relation. It can be shown (though the details are a bit tedious) that any finite order has this property.

Chapter 5

Infinities

Two sets A and B are said to be **equinumerous**, written $A \approx B$, iff there is a bijection from A to B. It follows that a set is finite iff it is equinumerous with a natural number.

It is easy to show that equinumerosity is an equivalence relation on the powerset of any set. It is not hard to show that for any set A, $\wp(A) \approx 2^A$: the bijection in question maps each subset of A to its characteristic function (with respect to A).

Intuitively speaking, equinumerosity may seem to amount to "having the same number of members". As we soon will see, this intuition is essentially on the mark in the case of finite sets. But when the sets involved are infinite, intuition may fail us. For example, all the following sets can be shown to be (pairwise) equinumerous: ω, $\omega \times \omega$, the set $\mathbb{Z}$ of integers, and the set $\mathbb{Q}$ of rational numbers.[1]

Not all infinite sets are equinumerous! To put it imprecisely but suggestively, there are "different sizes of infinity". For example, as Cantor famously proved, $\omega \not\approx I$, where I is the set of real numbers from 0 to 1. It is beyond the scope of this book to consider how the real numbers are modelled set-theoretically, but for our purposes it will suffice to think of I as the set of "decimal expansions", i.e. the set of functions from ω to 10 ($= \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$) excluding the ones which for some natural number n assign 9 to every natural number greater than or equal to n.[2] The proof is surprisingly simple: suppose f is an injection from ω to I. Then f cannot

[1] Actually proving all these things would of course require us to model 'the integers' and 'the rationals' as sets. There are standard ways of doing that, but limitations of space and time prevent us from spelling them out here.

[2] We can omit these because they have alternative decimal expansions, e.g. .7999... represents the same real number as .8000....

be a surjection. To see why, let r be the member of I (i.e. the function from ω to 10) which, for each $n \in \omega$, maps n to 6 if $f(n) = 5$ and maps n to 5 otherwise. A moment's thought shows that r cannot be in the range of f!

Theorem 5.1. *For any set A, $A \not\approx \wp(A)$.*

Proof. Let g be a function from A to $\wp(A)$. We will show g cannot be surjective. To this end, let $B = \{x \in A \mid x \notin g(x)\}$. Then obviously $B \in \wp(A)$. But B cannot be in the range of g. For suppose it were. In that case there would exist a $y \in A$ such that $B = g(y)$. But then $y \in B$ iff $y \notin g(y)$ In other words, $y \in B$ iff $y \notin B$, a contradiction. $\square$

A set is said to be **Dedekind infinite** iff it is equinumerous with a proper subset of itself. (Contrast this with the definition that a set is **infinite** if it is not equinumerous with any natural number.)

Theorem 5.2. *No natural number is Dedekind infinite.*

Proof. Exercise. [Hint: show that the set whose members are the natural numbers n such that every injective function from n to n is bijective is inductive.] $\square$

Corollary 5.1. *No finite set is Dedekind infinite..*

Proof. Exercise. $\square$

Corollary 5.2. *If A is Dedekind infinite, then it is infinite.*

Proof. Exercise. $\square$

Is the converse of this corollary true? We will return to this question later in this chapter.

Corollary 5.3. *ω is infinite.*

Proof. This follows from the preceding corollary together with the fact, proved in Chapter 4, that the successor function is a bijection from ω to $\omega \setminus \{0\}$. $\square$

Corollary 5.4. *No two distinct natural numbers are equinumerous.*

Proof. Exercise. [Hint: use the fact (Chapter 4) that the $\leq$ order on ω is connex, together with the theorem above.] $\square$

Corollary 5.5. *For Any finite set A, there is a unique natural number equinumerous with A.*

Proof. Exercise. $\square$

The unique natural number equinumerous with a finite set A is called the **cardinality** of A, written $|A|$.

Lemma 5.1. *If $C \subsetneq n \in \omega$, then $C \approx m$ for some $m < n$.*

Proof. Exercise. [Hint: show that the set whose members are those natural numbers n such that any proper subset of n is equinumerous to a member of n is inductive.] $\square$

Theorem 5.3. *Any subset A of a finite set B is itself finite.*

Proof. Let $n = |B|$, so there is a bijection $f\colon B \to n$. Then $f[A] \subseteq f[B] = n$. So either $f[A] = n$ or $f[A] \subsetneq n$. If $f[A] = n$, then $A \approx B$. If $f[A] \subsetneq n$, then by the previous lemma $f[A] \approx m$ for some $m < n$. $\square$

We say a set A is **dominated** by a set B, written $A \preceq B$, iff there is an injection from A to B, or, equivalently, iff A is equinumerous with a subset of B. If $A \preceq B$ and $A \not\approx B$, A is said to be **strictly dominated** by B, written $A \precnapprox B$ or $A \prec B$.

Some simple exercises are to show that for any sets A, B, and C, (a) $A \preceq A$; (b) if $A \preceq B$ and $B \preceq C$ then $A \preceq C$; and (c) $A \preceq \wp(A)$.

Theorem 5.4 (Schröder-Bernstein)**.** *For any sets A and B, if $A \preceq B$ and $B \preceq A$, then $A \approx B$.*

We have the resources to prove this, but since the proof is rather involved, we postpone it to the appendix.

Before continuing, we need to add to our list of assumptions about sets again (remember our last new assumption was that there is a set whose members are the natural numbers). To state the new assumption, we first need a couple of definitions. First, if A is a set, then the **nonempty powerset** of A, written $\wp_{ne}(A)$, is just $\wp(A) \setminus \{\emptyset\}$, i.e. the set of nonempty subsets of A. And second, a **choice function** for A is a function $c\colon \wp_{ne}(A) \to A$ such that, for each nonempty subset B of A, $c(B) \in B$. The new assumption is this:

Assumption 8 (Choice)**.** There is a choice function for any set.

It has been proved (by Paul Cohen, in 1963) that Choice is *independent* of the other assumptions we have made, in the sense that, if in fact our other assumptions are consistent, then either one of Choice or its denial (that some set does not have a choice function) can be added without leading to inconsistency. But as a practical matter, most working mathematicians prefer to assume Choice, because there are so many useful theorems that cannot be proved without it. One such theorem is the following:

Theorem 5.5. *If A is infinite, then $\omega \preceq A$.*

This proof is also deferred to the appendix.

Theorem 5.6 (Dedekind-Pierce)**.** *A set is infinite iff it is Dedekind infinite.*

Proof. The only-if part was proven above. Now suppose A is infinite. Then $\omega \preceq A$, that is, there is an injection $f\colon \omega \to A$. Now define a bijection $g\colon A \to A \setminus \{f(0)\}$ as follows: if $a \in A$ is not in the range of f, then $g(a) = a$; and if a is in the range of f, so that $a = f(n)$ for some $n \in \omega$, then $g(a) = f(n+1)$. It is easy to see that g is injective and its range is $A \setminus \{f(0)\}$. □

A set is said to be **countable** if it is dominated by ω. An infinite countable set is called **denumerable**, **denumerably infinite**, or **countably infinite**. A set which is not countable is called **uncountable**, **nondenumerable**, or **nondenumerably infinite**.

Corollary 5.6. *Any countably infinite set is equinumerous with ω.*

Proof. Exercise. □

Corollary 5.7. *Any infinite subset of ω is equinumerous with ω.*

Proof. Exercise. □

Some standard examples of countably infinite sets are the following: ω, $\omega \times \omega$, the positive natural numbers, the even natural numbers, $\mathbb{Z}$ (the integers), and $\mathbb{Q}$ (the rationals). Some standard examples of nondenumerable sets are $\mathbb{R}$ (the reals), the subset I of $\mathbb{R}$ consisting of the real numbers between 0 and 1 (including 0 and 1), and $\wp(\omega)$.

Now consider the following statement:

Proposition 5.1 (Continuum Hypothesis)**.** *There is no set A such that $\omega \precnapprox A \precnapprox \wp(\omega)$.*

This hypothesis has the same status as the Assumption of Choice: it can be proven to be independent of our other assumptions. The same is true of the following generalized form of the Continuum Hypothesis:

Proposition 5.2 (Generalized Continuum Hypothesis)**.** *For any infinite set B, there is no set A such that $B \precnapprox A \precnapprox \wp(B)$.*

Chapter 6

Introduction to Formal Languages

It is a familiar and basic intuition that language somehow involves stringing things together. Examples include stringing phonemes together to form syllables or (phonologies of) morphemes, stringing morphemes together to form words, stringing words together to form phrases (including sentences), and stringing sentences together to form discourses. Indeed, in the early days of syntactic theory (early to mid 1950s), natural languages were modelled as sets of strings, and the notion of a grammar was identified with a mathematical device for listing the members of such sets. But what exactly is a string?

6.1 Strings

As we saw in Chapter 4, for every natural number n,

$$n = \{m \in \omega \mid m < n\}$$

Let us now consider, for some set A and some $n \in \omega$, the set A^n, i.e. the set of arrows (functions with specified codomains) from n to A. The members of this set are called the **A-strings of length n**. In a linguistic application, we would think of the members of A as linguistic entities of some kind (phonemes, morphemes, words, etc.) that we would like to be able to string together, and of a particular A-string of length $n > 0$, f, as one of the possible results of stringing n such entities together, namely the one starting with $f(0)$, then $f(1)$, then $f(2)$, etc. If $f(i) = a_i$ for all $i < n$, then we usually denote f by the string (in the informal sense) of symbols $a_0 \ldots a_{n-1}$. (But in working with strings, it is important to remember that, technically, a string is not really a bunch of symbols lined up from left to right on a page, but rather a function whose domain is a natural number.) Also, it's important to note that there is exactly one A-string of length 0, denoted by ϵ_A (or just ϵ when no confusion is possible).[1] The set of all A-strings of length greater than 0 is denoted by A^+.

For strings of length 1, a mild notational confusion arises: if $f\colon 1 \to A$ and $f(0) = a$, then the notation 'a' could refer to either a itself (a member of A), or to the length-one A-string f. It should be clear from context which is intended. Note also that an A-string of length one is the same thing as a nullary operation on A.

The "infinite counterpart" of an A-string is called an **infinite A-sequence**; technically, an infinite A-sequence is a function from ω to A.

The set of all A-strings, i.e. the union of all the sets A^n, for all $n \in \omega$, is written A^*. Thus $A^* = A^+ \cup \{\epsilon_A\}$. When the identity of the set A is clear from context, we usually speak simply of strings, rather than A-strings. It should be obvious that there is a function from A^* to ω that maps each string to its length, and that the relation on strings of having the same length is an equivalence relation. Of course the sets in the partition induced by that equivalence relation are just the sets A^n. If A is a subset of another set B, then clearly there is an injection $\eta\colon A^* \to B^*$ that maps each A-string to a B-string just like it except that its codomain is B instead of A.

For each $n \in \omega$, there is an obvious bijection from $A^{(n)}$ to A^n. For $n \geq 2$, the bijection maps each n-tuple $\langle a_0, \ldots, a_{n-1}\rangle$ to the string $a_0 \ldots, a_{n-1}$. For $n = 1$, it maps each $a \in A$ to the length-one string that maps 0 to a; and for $n = 0$, it is the (only) function from 1 to $\{\epsilon_A\}$, i.e. the function that maps 0 to ϵ_A.

The binary operation of **concatenation** on A^*, written $\frown$, can be described intuitively as follows: if f and g are strings, then $f \frown g$ is the string that "starts with f and ends with g." More precisely, for each pair of natural numbers $\langle m, n\rangle$, if f and g are strings of length m and n respectively, then $f \frown g$ is the string of length $m + n$ such that

1. $(f \frown g)(i) = f(i)$ for all $i < m$; and
2. $(f \frown g)(m + i) = g(i)$ for all $i < n$.

It can be proven inductively (though the details are quite tedious) that for

[1] In Chapter 3, we called this $\Diamond_A$, but the name ϵ_A is more usual when we are thinking of it as a string.

any strings f, g, and h, the following equalities hold:[2]

1. $(f \frown g) \frown h = f \frown (g \frown h)$
2. $f \frown \epsilon = f = \epsilon \frown f$

Usually concatenation is expressed without the "$\frown$", by mere juxtaposition; e.g. fg for $f \frown g$. And because concatenation is an associative operation, we can write simply fgh instead of $f(gh)$ or $(fg)h$.

6.2 Formal Languages

A **formal** (A-)**language** is defined to be a subset of A^*. But when it is clear that we are talking about formal languages rather than natural languages, we will usually just speak of an A-language, or simply a language if the identity of A is clear from the context. In the most straightforward application of formal languages to linguistics, we mathematically model a natural language as a set of A-strings, where A is a set each of whose members is (a representation of) one of the words of the natural language in question. Of course this is a very crude model, since it disregards any linguistic structure a sentence has other than the temporal sequence of the words themselves. Additionally, once one speaks of a sentence as a string of words, one is immediately faced with the question of what counts as a word, or, to put it another way, what criterion of identity one is using for words. Is it enough to be homophonous (i.e. to sound the same), so that *meat* and *meet* count as the same word? Or to be homographic (written the same), so that *row* 'linear array' and *row* 'fight' count as the same word? Or must two words have the same sound, meaning, and 'part of speech' (whatever we think that is), so that *murder* counts as two words (one a noun and one a verb)? We will return to these and related questions in later chapters.

For the time being, we set such issues aside and assume we know what we mean by a 'word'. Assuming that, we can begin theorizing about questions such as the following: How many sentences (qua word strings) does the language have? Is there a way to list all its members? Is there a way to decide whether a given word string is a sentence of the language? Can we construct a plausible model of the process by which people who know the language recognize that a given string is a sentence of the language? Can the

[2]As we will see later, the truth of these equations means that A^* together with the nullary operation ϵ and the binary operation $\frown$ is an instance of a kind of algebra called a *monoid*; i.e. (1) $\frown$ is an *associative* operation, and (2) ϵ is a *two-sided identity* for $\frown$.

processing model somehow be extended to a model of how language users interpret utterances in context?

In order to address such questions, we need some techniques for defining formal languages. Since natural languages uncontroversially have an infinitude of sentences (how do you know?), it will not do to just make a list of A-strings. In due course we'll consider various kinds of *formal grammars*—mathematical systems for specifying formal languages—but we already have a powerful tool for doing just that, namely the Recursion Theorem (RT). One important way RT is used to specify an A-language L is roughly as follows: we start with (1) a set L_0 of A-strings which we know to be in the A-language we wish to define, and (2) a general method for adding more strings to any arbitrary set of strings, i.e. a function F from A-languages to A-languages. We can think of L_0 as the "dictionary" of the language we are trying to define and F as its "rules". We then define L as the union of the infinite sequence of languages $L_0, \ldots, L_n, \ldots$ where for each $k \in \omega$, L_{k+1} is the result of appying F to L_k.

To make this precise, it will help to introduce a little notation. First, suppose B is a set, $n \in \omega$, and $f \colon n \to B$ (in our applications, B will usually be $\wp(A^*)$.) Suppose also that for each $i < n$, $f(i) = x_i$. Then $\bigcup \mathsf{ran}(f)$ is written $\bigcup_{i<n} x_i$. If f is an infinite sequence in B, i.e. a function from ω to B, and $f(n) = x_n$ for all $n \in \omega$, then $\bigcup \mathsf{ran}(f)$ is written $\bigcup_{n\in\omega} x_n$. For example, it's not hard to see that $\bigcup_{n\in\omega} A^n = A^*$.

We now give a simple example of a recursive definition for a language. Intuitively, a *mirror image* string in A is one whose second half is the reverse of its first half. Informally, we define the language $\mathrm{Mir}(A)$ as follows:

1. $\epsilon \in \mathrm{Mir}(A)$;
2. If $x \in \mathrm{Mir}(A)$ and $a \in A$, then $axa \in \mathrm{Mir}(A)$;
3. Nothing else is in $\mathrm{Mir}(A)$.

Formally, this definition is justified by the RT as follows (here X, x, and F are as in the statement of RT in Chapter 4) we take X to be $\wp(A^*)$, x to be $\{\epsilon\}$, and $F \colon \wp(A^*) \to \wp(A^*)$ to be the function such that for any A-language S,

$$F(S) = \{y \in A^* \mid \exists a \exists x[a \in A \wedge x \in S \wedge y = axa]\}$$

.

RT then guarantees the existence of a function $h \colon \omega \to \wp(A^*)$ such that $h(0) = \{\epsilon\}$ and for every $n \in \omega$, $h(\mathbf{suc}(n)) = F(h(n))$. Finally, we define

$\mathrm{Mir}(A)$ to be $\bigcup_{n\in\omega} h(n)$. Intuitively, $h(n)$ is the set of all mirror image strings of length $2n$.

6.3 Operations on Languages

Let A be a set, so that A^* is the set of A-strings, $\wp(A^*)$ is the set of A-languages, and $\wp(\wp(A^*))$ is the set whose members are *sets* of A-languages.

We introduce the following notations for certain particularly simple A-languages:

a. For any $a \in A$, $\underline{a}$ is the singleton A-language whose only member is the string of length one a (remember this is the function from 1 to A that maps 0 to a).

b. $\underline{\epsilon}$ is the singleton A-language whose only member is the null A-string (i.e. the unique arrow from 0 to A). An alternative notation for this language is I_A.

c. $\emptyset$ as always is just the empty set, but for any A we can also think of this as the A-language which contains no strings! An alternative notation for this language is 0_A.

Next, we define some operations on $\wp(A^*)$. In these definitions L and M range over A-languages.

a. The **concatenation** of L and M, written $L \bullet M$, is the set of all strings of the form $u \frown v$ where $u \in L$ and $v \in M$.

b. The **right residual** of L by M, written L/M, is the set of all strings u such that $u \frown v \in L$ for every $v \in M$.

c. The **left residual** of L by M, written $M\backslash L$, is the set of all strings u such that $v \frown u \in L$ for every $v \in M$.

d. The **Kleene closure** of L, written $\mathsf{kl}(L)$, has the following informal recursive definition (formalizing this definition will be an exercise):

   i. (base clause) $\epsilon \in \mathsf{kl}(L)$;

   ii. (recursion clause) if $u \in L$ and $v \in \mathsf{kl}(L)$, then $uv \in \mathsf{kl}(L)$; and

   iii. nothing else is in $\mathsf{kl}(L)$.

To put it even less formally but more intuitively: the Kleene closure of L is the language whose members are those strings that result from concatenating together zero or more strings drawn from L.

e. The **positive Kleene closure** of L, written $\mathsf{kl}^+(L)$, has the following informal recursive definition:

   i. (base clause) If $u \in L$, then $u \in \mathsf{kl}^+(L)$;

   ii. (recursion clause) if $u \in L$ and $v \in \mathsf{kl}^+(L)$, then $uv \in \mathsf{kl}^+(L)$; and

   iii. nothing else is in $\mathsf{kl}^+(L)$.

   Intuitively: the positive Kleene closure of L is the language whose members are those strings that result from concatenating together one or more strings drawn from L.

6.4 Regular Languages

Linguists are often concerned not just with languages, but with *sets* of languages, e.g. the set of finite languages, the set of *decidable* languages (languages for which an algorithm exists that tells for any given string whether it is in the language), the set of *recursively enumerable languages* (languages for which an algorithm exists for listing all its strings while not listing any strings not in the language), etc. In computational linguistics applications, one of the most important sets of languages is (for a fixed alphabet A) the set $\mathsf{Reg}(A)$ of **regular** A-languages. As with many other important sets of languages, there are several different ways to define this set, all of which give the same result. For our purposes, the simplest way is a recursive definition. The informal version runs as follows:

1. For each $a \in A$, $\underline{a} \in \mathsf{Reg}(A)$;

2. $\mathsf{0}_A \in \mathsf{Reg}(A)$;

3. $\mathsf{I}_A \in \mathsf{Reg}(A)$;

4. for each $L \in \mathsf{Reg}(A)$, $\mathsf{kl}(L) \in \mathsf{Reg}(A)$;

5. for each $L, M \in \mathsf{Reg}(A)$, $L \cup M \in \mathsf{Reg}(A)$;

6. for each $L, M \in \mathsf{Reg}(A)$, $L \bullet M \in \mathsf{Reg}(A)$; and

7. nothing else is in $\mathsf{Reg}(A)$.

Note that in this definition, the first three clauses are base clauses and the next three are recursion clauses. The formalization of this definition using RT is left as an exercise. (Hint: remember that we are defining not a language, but rather a set of languages, and therefore the choice of X (as in the statement of RT in Chapter 4) is not $\wp(A^*)$ but rather $\wp(\wp(A^*))$.

6.5 Context-Free Grammars

Context-free grammars (CFGs) are a particular way of defining languages recursively that is very widely used in syntactic theory; in one form or another, CFGs play a central role in a wide range of syntactic frameworks (here 'framework' means, roughly, a research paradigm or community), including, to name just a few, all forms of transformational grammar (TG); many kinds of categorial grammar (CG); lexical-functional grammar (LFG); generalized phrase structure grammar (GPSG); and head-driven phrase structure grammar (HPSG). In due course it will emerge that CFGs are a rather blunt instrument for modelling natural languages, but they are a good point of departure in the sense that they can be elaborated, refined, and adapted in many ways (some of which we will examine closely) that make them more suitable for this task.

The basic idea behind CFGs is to *simultaneuously* recursively define a finite set of different languages, each of which consitutes a set of strings that have the same "distribution" or "privileges of occurrence" or "combinatory potential" in the whole language being defined, which is the union of that set of languages. The languages in that family are called the **syntactic categories** of the whole language.

Getting technical, a CFG consists of four things: (1) a finite set T whose members are called **terminals**; (2) a finite set N whose members are called **nonterminals**; (3) a finite set D of ordered pairs called **lexical entries**, each of which has a nonterminal as its left component and a terminal as its right component[3]; and (4) a finite set P of ordered pairs called **phrase structure rules** (or simply PSRs), each of which has a nonterminal as its left component and a non-null string of nonterminals as its right component[4].

Intuitively, the terminals are the words (or word phonologies, or word orthographies – see above) of the language under investigation. The non-

[3]Formal language theorists usually allow any T-string as the right component of a lexical entry, but we will not need this generality for our applications.

[4]Formal language theorists usually allow any $(N \cup T)$-string containing at least one nonterminal as the right component of a PSR, but again this generality goes beyond the needs of our linguistic applications.

terminals are names of the syntactic categories. The lexical entries make up the dictionary (or lexicon) of the language. And the PSRs provide a mechanism for telling which strings (other than length-one strings of words) are in the language and what syntactic categories they belong to. Once all this is made more precise, the CFG will specify, for each nonterminal A, a T-language C_A, and the language defined by the CFG will be the union over all $A \in N$ of the C_A.

We'll make all this precise in two stages, first using an informal recursive definition (the usual kind), and then a more informal or 'official' definition employing the Recursion Theorem (RT).

First, the informal version. As with all recursive definitions, a CFG has a base part and a recursion part. The base part makes use of the lexicon D and the recursion part uses the set P of PSRs. Starting with the lexicon, remember that formally a lexical entry is an ordered pair $\langle A, t\rangle \in D \subseteq N \times T$; but formal language theorists usually write entries in the form

$$A \to t$$

to express that $\langle A, t\rangle \in D$. In the informal recursive definition, the significance of a lexical entry expressed as follows:

$$\text{If } A \to t, \text{ then } t \in C_A.$$

That is: for any terminal a which the dictionary pairs with the nonterminal A, the string a of length one will be in the category which that nonterminal names.

Note that it is conventional to abbreviate sets of lexical entries with the same left-hand side using curly brackets on the right-hand side, e.g.

$$A \to \{t_1, t_2\}$$

abbreviates

$$A \to t_1$$
$$A \to t_2$$

As mentioned above, the recursive part of the (informal) recursive definition draws on the set P of PSRs. Technically, a PSR is an ordered pair $\langle A, A_0 \ \ldots A_{n-1}\rangle \in P \subseteq N \times N^+$, but formal language theorists usually write form

$$A \to A_0 \ \ldots A_{n-1}$$

to express that $\langle A, A_0 \ \ldots A_{n-1} \rangle \in P$. In the informal recursive definition, the significance of a PSR is expressed this way:

If $A \to A_0 \ \ldots A_{n-1}$ and for each $i < n$, $s_i \in C_{A_i}$, then $s_0 \ldots s_{n-1} \in C_A$.

That is: if, for each nonterminal on the right-hand-side of some rule, we have a string belonging to the category named by that nonterminal, then the result of concatenating together all those strings (in the same order in which the corresponding nonterminals appear in the rule) is a member of the category named by the nonterminal on the left-hand side of the rule.

As with lexical entries, sets of rules with the same left-hand side can be abbreviated using curly brackets on the right-hand side.

Before going on to the formal, RT-based formulation of CFGs, we illustrate the informal version with a 'toy' (i.e. ridiculously simplified) linguistic example.

$T = \{\mathsf{Fido}, \mathsf{Felix}, \mathsf{Mary}, \mathsf{barked}, \mathsf{bit}, \mathsf{gave}, \mathsf{believed}, \mathsf{the}, \mathsf{cat}, \mathsf{dog}, \mathsf{yesterday}\}$

$N = \{\mathrm{S}, \mathrm{NP}, \mathrm{VP}, \mathrm{TV}, \mathrm{DTV}, \mathrm{SV}, \mathrm{Det}, \mathrm{N}\}$

D consist of the following lexical entries:

$\mathrm{NP} \to \{\mathsf{Fido}, \mathsf{Felix}, \mathsf{Mary}\}$

$\mathrm{VP} \to \mathsf{barked}$

$\mathrm{TV} \to \mathsf{bit}$

$\mathrm{DTV} \to \mathsf{gave}$

$\mathrm{SV} \to \mathsf{believed}$

$\mathrm{Det} \to \mathsf{the}$

$\mathrm{N} \to \{\mathsf{cat}, \mathsf{dog}\}$

$\mathrm{Adv} \to \mathsf{yesterday}$

P consists of the following PSRs:

$\mathrm{S} \to \mathrm{NP\ VP}$

$\mathrm{VP} \to \{\mathrm{TV\ NP}, \mathrm{DTV\ NP\ NP}, \mathrm{SV\ S}, \mathrm{VP\ Adv}\}$

$\mathrm{NP} \to \mathrm{Det\ N}$

In this grammar, the nonterminals are names for the syntactic categories of noun phrases, verb phrases, transitive verbs, sentential-complement verbs, ditransitive verbs, determiners, and common noun phrases.[5] The lexical entries tell us, for example, that **Felix** (the length-one word string, not the word itself) is a member of the syntactic category C_{NP}, and the PSRs tell us, for example, that the string that results from concatenating two strings, one belonging to the syntactic category C_{NP} (e.g. **Felix**) and the other belonging to the syntactic category C_{VP} (e.g. **barked**), in that order (in this case, the length-two string **Felix barked**), belongs to the syntactic category C_{S}.

Finally, we show how to formalize the simultaneous recursive definition of the syntactic categories associated with a CFG, using the RT. As always when applying the RT, the key is making the right choice for the three pieces of data X, x, and F. Since we are defining not a language but rather a function from nonterminals to languages, the right choice for X is not $\wp(T^*)$ but rather $\wp(T^*)^N$; x will be a member of this set, and F will be a function from this set to itself.

So what is x? Intuitively, it should tell us, for each nonterminal A, which strings are in the syntactic category C_A by virtue of the lexicon alone, i.e. without appealing to the recursive part of the defnition (the PSRs). That is, x is the function that maps each nonterminal A to the set of strings t (all of which will have length one) such that $A \to t$ is one of the lexical entries.

What about F? What should be the result of applying F to an arbitrary function $L : N \to \wp(T^*)$? Well, for each $A \in N$, we will want $F(L)(A)$ to contain all the strings that were in $L(A)$, together with any strings that can be obtained by applying a rule of the form $A \to A_0 \ldots A_{n-1}$ to strings $s_0, \ldots, s_{n-1}$, where, for each $i < n$, s_i belongs to the language that L assigned to A_i. Another way to say this is that F maps each L to the function that maps each nonterminal A to the language which is the union of the following two languages: (1) $L(A)$, and (2) the union, over all rules of the form $A \to A_0 \ldots A_{n-1}$, of the languages $L(A_0) \bullet \ldots \bullet L(A_{n-1})$.

Given these values of X, x, and F, the RT guarantees us a unique function h from ω to functions from N to $\wp(T^*)$. Finally, for each nonterminal A, we define the corresponding syntactic category to be

$$C_A =_{\mathrm{def}} \bigcup\nolimits_{n \in \omega} h(n)(A)$$

A suggested exercise here is to calculate, for as many values of n as you have

[5] The category names are a bit confusing, since the categories of noun phrases, verb phrases, and common noun phrases are allowed to contain length-one strings (intuitively, words).

patience for, and for each nonterminal A, the value of $h(n+1)(A) \setminus h(n)(A)$ (that is, the set of strings that are added to C_A at the nth recursive step).

Chapter 7

(Pre)Semilattices and Trees

7.1 Informal Motivation

As we will illustrate presently, given a CFG $\langle T, N, D, P \rangle$, a nonterminal $A \in N$, and a T-string $s \in C_A$, we can use the CFG to guide us in constructing a proof that $s \in C_A$. In fact, as anyone who has taken a course in formal language theory or computational linguistics will already realize, there are general procedures for deciding, given *any* CFG $\langle T, N, D, P \rangle$, *any* T-string s and *any* nonterminal A, whether or not $s \in C_A$. Such a procedure is called a **recognizer** because it tells, in effect, whether the CFG recognizes a given string as a member of a given syntactic category. In order to decide correctly that $s \in C_A$, the recognizer essentially must construct a *proof* that this is the case. What about making the correct decision when $s \notin C_a$? For that to be possible, the recognizer must in some sense 'know' when it has gotten to the point where, had there been a proof that $s \in C_A$, it would have found one; at that point it would render a negative decision. A **parser**, roughly speaking, is just a recognizer which renders not merely a decision but also (symbolic representations of) the proofs (if any) upon which the decision was based.

The construction of recognizers and parsers for CFGs and other kinds of formal grammars, one of the central concerns of both formal language theorists and of computational linguists, is a very highly evolved and subtle discipline, which unfortunately is beyond the scope of this book. However, the fundamental distinction between a parser and a (mere) recognizer has an analog that is relevant even for empirical/theoretical linguists (as opposed to formal language theorists and computational linguists), namely the intuition that a sentence is not just a string of words that belongs to C_{S} but rather a *way* that the string in question belongs to C_{S}. To take a very simple example, let's consider a slightly expanded version of the toy English grammar in Chapter 6, as follows:

$T = \{$Fido, Felix, Mary, barked, bit, gave, believed, heard, the, cat, dog, yesterday$\}$

$N = \{\mathrm{S},\ \mathrm{NP},\ \mathrm{VP},\ \mathrm{TV},\ \mathrm{DTV},\ \mathrm{SV},\ \mathrm{Det},\ \mathrm{N},\ \mathrm{Adv}\}$

D consist of the following lexical entries:

$\mathrm{NP} \to \{\mathsf{Fido},\ \mathsf{Felix},\ \mathsf{Mary}\}$

$\mathrm{VP} \to \mathsf{barked}$

$\mathrm{TV} \to \mathsf{bit}$

$\mathrm{DTV} \to \mathsf{gave}$

$\mathrm{SV} \to \{\mathsf{believed},\ \mathsf{heard}\}$

$\mathrm{Det} \to \mathsf{the}$

$\mathrm{N} \to \{\mathsf{cat},\ \mathsf{dog}\}$

$\mathrm{Adv} \to \mathsf{yesterday}$

P consists of the following PSRs:

$\mathrm{S} \to \mathrm{NP\ VP}$

$\mathrm{VP} \to \{\mathrm{TV\ NP},\ \mathrm{DTV\ NP\ NP},\ \mathrm{SV\ S},\ \mathrm{VP\ Adv}\}$

$\mathrm{NP} \to \mathrm{Det\ N}$

The only additions are (1) the nonterminal Adv (adverb); (2) the terminals heard and yesterday; (3) the lexical entries for yesterday as an adverb and for heard as a sentential-complement verb; and (4) the PSR VP $\to$ VP Adv.

Now consider the string $s =$ **Mary heard Fido bit Felix yesterday**. According to our grammar, $s \in C_{\mathrm{S}}$ (the syntactic category of sentences), but few (if any) syntacticians would say that s is an English sentence! Rather, they would say that the word string s corresponds to *two different sentences*, one roughly paraphrasable as *Mary heard yesterday that Fido bit Felix* and another roughly paraphrasable as *Mary heard that yesterday, Fido bit Felix.* Of course, these two sentences mean different things; but more relevant for our present purposes is that we can also characterize the difference between the two sentences purely in terms of two distinct ways of *proving* that $s \in C_{\mathrm{S}}$.

To understand this point, remember from Chapter 6 that the set of syntactic categories is (informally) defined by simultaneous recursive definition as follows:

1. (Base Clause) If $A \to t$, then $t \in C_A$.
2. (Recursion Clause) If $A \to A_0 \ \ldots A_{n-1}$ and for each $i < n$, $s_i \in C_{A_i}$, then $s_0 \ldots s_{n-1} \in C_A$.

Then the two proofs run as follows:

Proof 1. From the lexicon and the base clause, we know that **Mary, Fido, Felix** $\in C_{\mathrm{NP}}$, **heard** $\in C_{\mathrm{SV}}$, **bit** $\in C_{\mathrm{TV}}$, and **yesterday** $\in C_{\mathrm{Adv}}$. Then, by repeated applications of the recursion clause, it follows that:

a. since **bit** $\in C_{\mathrm{TV}}$ and **Felix** $\in C_{\mathrm{NP}}$, **bit Felix** $\in C_{\mathrm{VP}}$;

b. since **bit Felix** $\in C_{\mathrm{VP}}$ and **yesterday** $\in C_{\mathrm{Adv}}$, **bit Felix yesterday** $\in C_{\mathrm{VP}}$;

c. since **Fido** $\in C_{\mathrm{NP}}$ and **bit Felix yesterday** $\in C_{\mathrm{VP}}$, **Fido bit Felix yesterday** $\in C_{\mathrm{S}}$;

d. since **heard** $\in C_{\mathrm{SV}}$ and **Fido bit Felix yesterday** $\in C_{\mathrm{S}}$, **heard Fido bit Felix yesterday** $\in CP_{\mathrm{VP}}$; and finally,

e. since **Mary** $\in C_{\mathrm{NP}}$ and **heard Fido bit Felix yesterday** $\in C_{\mathrm{VP}}$, **Mary heard Fido bit Felix yesterday** $\in C_{\mathrm{S}}$.

$\square$

Proof 2. The same as Proof 1, up through step a. From there, we proceed as follows:

a. since **Fido** $\in C_{\mathrm{NP}}$ and **bit Felix** $\in C_{\mathrm{VP}}$, **Fido bit Felix** $\in C_{\mathrm{S}}$;

b. since **heard** $\in C_{\mathrm{SV}}$ and **Fido bit Felix** $\in C_{\mathrm{S}}$, **heard Fido bit Felix** $\in C_{\mathrm{VP}}$;

c. since **heard Fido bit Felix** $\in C_{\mathrm{VP}}$ and **yesterday** $\in C_{\mathrm{Adv}}$, **heard Fido bit Felix yesterday** $\in C_{\mathrm{VP}}$; and finally,

d. since **Mary** $\in C_{\mathrm{NP}}$ and **heard Fido bit Felix yesterday** $\in C_{\mathrm{VP}}$, **Mary heard Fido bit Felix yesterday** $\in C_{\mathrm{S}}$.

□

There is nothing complicated about any of this, but this is not how a syntactician would usually describe the difference between the two homophonous sentences. Instead, s/he would draw two different *tree diagrams* as in Figures 7.1 and 7.2.

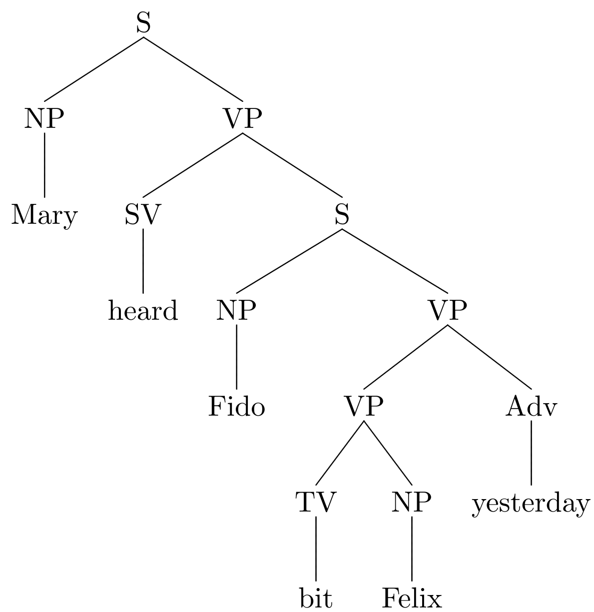

Figure 7.1: Tree diagram with 'low' adverb attachment.

But what exactly *are* tree diagrams, and what is supposed to be their relationship to the linguistic phenomena being theorized about? Well, roughly speaking (we will get more precise in the following two sections), the diagrams are elaborated Hasse diagrams of mathematical objects called *labelled trees*. And what are labelled trees? Well, *trees* are partially ordered sets of a certain kind, and a labelled tree is a tree together with a function that assigns things called *labels* to the members (called *nodes*) of the partially ordered set. When syntacticians use labelled trees, the labels assigned to the minimal nodes are drawn from the set T of terminals and the labels assigned to the other nodes are drawn from the set N of nonterminals.

Intuitively, it is pretty clear that these two tree diagrams are closely related to, or in some sense correspond to, the two proofs given earlier (though the precise relationship remains quite obscure at this stage). But

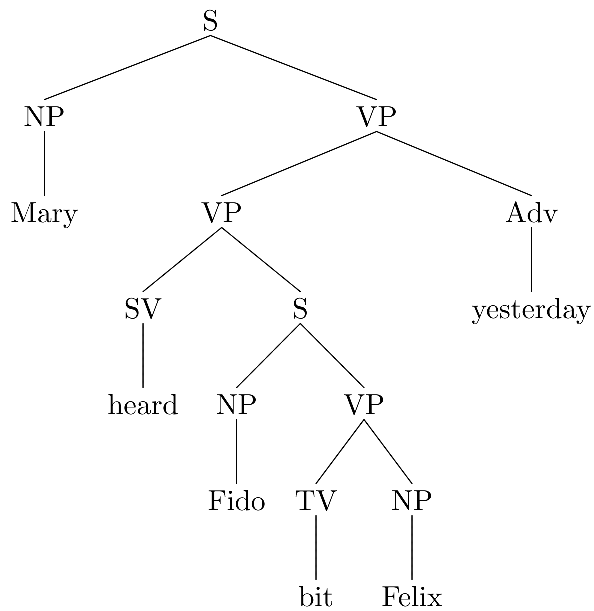

Figure 7.2: Tree diagram with 'high' adverb attachment.

when we begin to construct a linguistic theory, which should we use? Should we use labelled trees or elaborations of them as set-theoretic idealizations of sentences, as is done in syntactic frameworks such as HPSG (head-driven phrase structure grammar) and LFG (lexical-functional grammar)? Or is it better to think of a sentence as a proof that a certain string belongs to a certain syntactic category, as is done in CG (categorial grammar)? Or is it perhaps best to use a hybrid approach with both set-theoretic and proof-theoretic aspects, such as most forms of transformational grammar (TG), which include MP (the Minimalist Program) and its predecessor GB (Government-Binding)? We will not be able to answer these questions until we start to formalize logic and look at how formal logic is applied to linguistic theory. But because tree representations are so widely used by syntacticians (not to mention semanticists, computational linguists, and logicians), it is important for us to get clear early on precisely what trees are and how syntacticans use them. To that end, it will be convenient to first consider some somewhat more general notions, *(pre-)semilattiices*, of which trees are a special case. These will turn out to have a wide range of other special cases, such as residuated lattices, heyting algbras, and boolean algebras, with importnat linguistic applications of their own.

7.2 LUBs, GLBs, and (Pre-)Semilattices

7.2.1 Least Upper Bounds and Greatest Lower Bounds

Throughout this section, $\sqsubseteq$ is a preorder on a set A, and $\equiv$ ($=\equiv_{\sqsubseteq}$) is the equivalence relation induced by the preorder.

Suppose $a \in A$ and $S \subseteq A$. Recall (Chapter 3, subsection 3.3.2) that a is a **greatest** (respectively, **least**) member of S provided, for every $b \in S$, $b \sqsubseteq a$ (respectively, $a \sqsubseteq b$). Greatest (least) members of A are called **tops** (**bottoms**). All greatest (least) members of S are equivalent. So if $\sqsubseteq$ is an order, than S can have at most one greatest (least) member (because the equivalence relation induced by an order is the identity relation). In particular, in an order, there can be at most one top and at most one bottom.

Continuing to suppose $a \in A$ and $S \subseteq A$, recall also that a is a **maximal** (respectively, **minimal**) member of S iff, for every $b \in S$, $a \sqsubseteq b$ (respectively, $b \sqsubseteq a$) implies that $a \equiv b$. We observed that if $\sqsubseteq$ is an order, then if a is the greatest (least) member of S, then it is also the unique maximal (minimal) member of S.

If $S \subseteq A$ and $a \in A$, we call a an **upper** (**lower**) **bound** of S provided, for every $b \in S$, $b \sqsubseteq a$ (respectively, $a \sqsubseteq b$). Thus by definition a greatest (least) member of S is an upper (lower) bound of S; but the upper (lower) bounds of S need not be members of S. In case $S = A$, the notions of upper bound and greatest member (top) (lower bound and least member (bottom)) coincide.

The set of upper (lower) bounds of S is denoted by $\mathrm{UB}(S)$ ($\mathrm{LB}(S)$). In case S is a singleton $\{a\}$, $\mathrm{UB}(S)$ ($\mathrm{LB}(S)$) is written $\uparrow a$ ($\downarrow a$), read **up** of a (**down** of a).

A least member of $\mathrm{UB}(S)$ is called a **least upper bound (lub)** of S, and a greatest member of $\mathrm{LB}(S)$ is called a **greatest lower bound (glb)** of S. In case $S = A$, the notions of lub and top (glb and bottom) coincide. (A good exercise here is to find an example where a lub of S does not belong to S.) If S has a *unique* glb (lub), it is written $\bigsqcap S$ ($\bigsqcup S$). If A has a unique top (bottom), it is written $\top$ ($\bot$). (A good exercises here is to show that if there are any bottoms, they are the least upper bounds of $\emptyset$, and if there are any tops, they are the greatest lower bounds of $\emptyset$.)

If $\sqsubseteq$ is an order on A and $S \subseteq A$, then S can have at most one glb (lub). (Why?) For the case $S = A$, this means there can be at most one top and at most one bottom. In the special case $S = \{a, b\}$, if S has a glb (lub), it is written $a \sqcap b$ ($a \sqcup b$). It is an easy but tedious exercise to show the following, for all $a, b, c \in A$:

Facts about $\sqcap$ and $\sqcup$ when $\sqsubseteq$ is an order:

1. (Idempotence) $a \sqcap a$ exists and equals a;
2. (Commutativity) if $a \sqcap b$ exists, so does $b \sqcap a$, and they are equal;
3. (Associativity) if $(a \sqcap b) \sqcap c$ and $a \sqcap (b \sqcap c)$ exist, they are equal; and
4. (Interdefinability) $a \sqsubseteq b$ iff $a \sqcap b$ exists and equals a.

Given these facts, one can immediately establish corresponding facts with all instances of $\sqcap$ replaced by $\sqcup$, by considering A with the **opposite** (or **dual**) order $\sqsubseteq^{-1}$, usually called $\sqsubseteq^{\mathrm{op}}$. This is an example of a widely used proof technique called **duality** whereby a result about a preorder is reinterpreted as a result about the dual (pre-)order.

Now suppose we have two sets A and B preordered by $\sqsubseteq$ and $\leq$ respectively. A function $f\colon A \to B$ is called **monotonic** or **order-preserving** with respect to the given preorders provided, for all $a, a' \in A$, if $a \sqsubseteq a'$, then $f(a) \leq f(a')$; and f is called **antitonic** or **order-reversing** with respect to the given preorders provided for all $a, a' \in A$, if $a \sqsubseteq a'$ then $f(a') \leq f(a)$. A monotonic (respectively, antitonic) bijection is called a **preorder isomorphism** (respectively, **preorder anti-isomorphism**) provided its inverse is also monotonic (respectively, antitonic). Two preordered sets are said to be **preorder-isomorphic** provided there is a preorder isomorphism from one to the other. Intuitively speaking, preorder-isomorphic preorders are "copies of each other", differing only in which members they contain.

It is possible to consider to different preorders on the same set. For example, besides the usual order $\leq$ on the nonzero natural numbers, we could also the consider the order $\sqsubseteq$ that holds between a pair of nonzero natural numbers if the first is a factor of (i.e. divides evenly into) the second.

If $\sqsubseteq$ and $\leq$ are two preorders on the same set A, we can ask whether the identity function on A is monotonic from the first to the second. Interestingly, many (all?) languages have a special grammatical construction, called the **correlative comparative** construction, to describe situations of this kind. A typical English example is a sentence such as *The more expensive an SUV is, the more cupholders it has*, which asserts that the identity function on the set of SUVs is monotonic from $\sqsubseteq$ to $\leq$, where, for two SUVs a and b, $a \sqsubseteq b$ means b costs at least as much as a, and $a \leq b$ means that b has at least as many cupholders as a.

7.2.2 (Pre-)semilattices

Now let A be a set with a preorder $\sqsubseteq$ and a binary operation $\sqcap$ ($\sqcup$) such that, for all $a, b \in A$, $a \sqcap b$ ($a \sqcup b$) is a glb (lub) of $\{a, b\}$ (not necessarily the only one). Such an operation is called a **meet** (**join**) operation, and a preorder equipped with such an operation is called a **lower** (**upper**) **presemilattice**; one with both is called a **prelattice**. If the preorder is an order, the *pre-* prefix is dropped: thus, an order with a meet (join) is a **lower** (**upper**) semilattice, and an order with both is simply a **lattice**. Then we have, for all $a, b, c, d \in A$:

Facts about $\sqcap$ in a lower presemilattice:

1. (Idempotence up to Equivalence) $a \sqcap a \equiv a$;
2. (Commutativity up to Equivalence) $a \sqcap b \equiv b \sqcap a$;
3. (Associativity up to Equivalence) $(a \sqcap b) \sqcap c \equiv a \sqcap (b \sqcap c)$;
4. (Interdefinability) $a \sqsubseteq b$ iff $a \sqcap b \equiv a$;
5. (Monotonicity on Both Sides) For each $a \in A$, the function that maps each $b \in A$ to $a \sqcap b$ ($b \sqcap a$) is monotonic.
6. (Substititivity up to Equivalence) If $a \equiv c$ and $b \equiv d$ then $a \sqcap b \equiv c \sqcap d$.

Duality gives corresponding facts for join in an upper presemilattice. We also have the following:

Facts about $\sqcap$ and $\sqcup$ in a prelattice:

1. (Absorption up to Equivalence) $(a \sqcup b) \sqcap b \equiv b \equiv (a \sqcap b) \sqcup b$;
2. (Semidistributivity) $(a \sqcap b) \sqcup (a \sqcap c) \sqsubseteq a \sqcap (b \sqcup c)$.

A prelattice is called **distributive** if the inequality reverse to Semidistributivity holds:

$$a \sqcap (b \sqcup c) \sqsubseteq (a \sqcap b) \sqcup (a \sqcap c)$$

holds. Thus in a distributive prelattice, we have the following (Distributivity up to Equivalence):

$$a \sqcap (b \sqcup c) \equiv (a \sqcap b) \sqcup (a \sqcap c)$$

It can be shown (though it is a fair amount of work) that this equivalence holds in a prelattice just in case the dual one (formed by interchanging meets and joins) does.

Of course all the equivalences stated in this section become equalities if the preorder in question is an order.

7.3 Trees

7.3.1 Technical Preminaries

Here we gather together some facts that will simplify the discussion of trees.

Theorem 7.1. *Any nonempty finite order has a minimal (and so, by duality, a maximal) member.*

Proof sketch. Let T be the set of natural numbers n such that every ordered set of cardinality $n+1$ has a minimal member, and show that T is inductive. The main idea of the proof is to show that T is an inductive set. $\square$

Corollary 7.1. *Any nonempty finite chain has a least (and so, by duality, a greatest) member.*

Proof. This follows from the fact (itself a simple consequence of connexity) that in a chain, a member is least (greatest) iff it is minimal (maximal). $\square$

Theorem 7.2. *For any natural number n, any chain of cardinality n is order-isomorphic to the usual order on n (i.e. the restriction to n of the usual $\leq$ order on ω).*

Proof sketch. By induction on n. The case $n = 0$ is trivial. By inductive hypothesis, assume the statement of the theorem holds for the case $n = k$ and let A of cardinality $k + 1$ be a chain with order $\sqsubseteq$. By the Corollary, A has a greatest member a, so there is an order isomorphism f from k to $A \setminus \{a\}$. The rest of the proof consists of showing that $f \cup \{\langle k, a\rangle\}$ is an order isomorphism. $\square$

Theorem 7.3. *Suppose $\sqsubseteq$ is an order on a finite set A. Then $\sqsubseteq = \prec^*$. That is: a finite order is the reflexive transitive closure of its own covering relation.*

Proof. That $\prec^* \subseteq \sqsubseteq$ follows easily from the definition of reflexive transitive closure and the the transitivity of $\sqsubseteq$. To prove the reverse inclusion, suppose $a \sqsubseteq b$ and let X be the (nonempty, finite) set of all subsets of A which, when ordered by $\sqsubseteq$, are chains with b as greatest member and a as least member. (X is nonempty since one of its members is $\{a, b\}$.) Then X itself is ordered by $\subseteq_X$, and so by Theorem 7.1 has a maximal member C. Let $n + 1$ be $|C|$; by Theorem 7.2, there is an order-isomorphism $f \colon n + 1 \to C$. Clearly $n > 0$, $f(0) = a$, and f(n) = b. Also, for each $m < n$, $f(m) \prec f(m + 1)$, because otherwise, there would be a c properly between $f(m)$ and $f(m+1)$, contradicfting the maximality of C. □

7.3.2 Trees

We now define a **tree** is to be a finite set A with an order $\sqsubseteq$ and a top $\top$, such that the covering relation $\prec$ is a function with domain $A \setminus \{\top\}$. In the linguistic community, the following terminology for trees is standard:

1. The members of A are called the **nodes** of the tree.
2. $\top$ is called the **root**.
3. If $x \sqsubseteq y$, y is said to **dominate** x; and if additionally $x \neq y$, then y is said to **properly dominate** x.
4. If $x \prec y$, then y is said to **immediately dominate** x. In that case $y = \prec(x)$ is called the **mother** of x, and x is said to be a **daughter** of y.
5. Distinct nodes with the same mother are called **sisters**.
6. A minimal node (i.e. one with no daughters) is called a a **terminal** node.
7. A node which is the mother of a terminal node is called a **preterminal** node.

We state here some important facts about trees, sketching some of the proofs and leaving others as exercises.

Theorem 7.4. *No node can dominate one of its sisters.*

Proof. Exercise. □

Theorem 7.5. *For any node a, $\uparrow a$ is a chain.*

Proof sketch. Use the RT to define a function $h\colon \omega \to A$, with $X = A$, $x = a$, and F the function which maps non-root nodes to their mothers and the root to itself. Now let $Y = \mathsf{ran}(h)$; it is easy to see that Y is a chain, and that $Y \subseteq {\uparrow}a$. It remains to show that ${\uparrow}a \subseteq Y$. So assume $b \in {\uparrow}a$; we have to show $b \in Y$.

By definition of ${\uparrow}a$, $a \sqsubseteq b$, and so by Theorem 7.3 of section 7.3, $a \prec^* b$. From this and the definition (Chapter 4, section 4.4) of reflexive transitive closure, it follows that there is a natural number n such that $a \prec_n b$, where $\prec_n$ is the n-fold composition of $\prec$ with itself. In other words, there is an A-string $a_0 \ldots a_n$ such that $a_0 = a$, $a_n = b$, and for each $k < n$, $a_k \prec a_{k+1}$. But then $b = h(n)$, so $b \in Y$. $\square$

Corollary 7.2. *Two distinct nodes have a meet iff they are comparable.*

Proof. Exercise. $\square$

Theorem 7.6. *Any two nodes have a lub (and so a tree is a join semilattice).*

Proof. Exercise. $\square$

7.3.3 Ordered Trees

An **ordered tree** is a set A with *two* orders $\sqsubseteq$ and $\leq$, such that the following three conditions are satisfied:

1. A is a tree with respect to $\sqsubseteq$.
2. Two distinct nodes are $\leq$-comparable iff they are not $\sqsubseteq$ comparable.
3. (No-tangling condition) If a, b, c, d are nodes such that $a < b$, $c \prec a$, and $d \prec b$, then $c < d$.

In an ordered tree, if $a < b$, then a is said to **linearly precede** b.

Theorem 7.7. *If a is a node in an ordered tree, then the set of daughters of a ordered by $\leq$ is a chain.*

Proof. Exercise. $\square$

Theorem 7.8. *In an ordered tree, the set of terminal nodes ordered by $\leq$ is a chain.*

Proof. Exercise. $\square$

7.4 Trees in Syntax

In many syntactic frameworks, ordered trees (often elaborated with some additional structure) are employed in the modelling of linguistic expressions (words and phrases). In such contexts, the trees are variously called *phrase structures*, *phrase structure trees*, or *phrase markers*. Typically (though the details differ from framework to framework, as discussed below), the grammar (in the sense of the total theory of the particular natural language being analyzed) will include a CFG $\langle T, N, D, P\rangle$, and the phrase structure trees will be ordered trees equipped with a **labelling** function that assigns terminal symbols to terminal nodes and nonterminals to nonterminal nodes. We then speak of the set of phrase structure trees **generated by**, or **licensed by**, or **admitted by** the CFG, which is defined as follows:

The Set of Phrase Structure Trees Admitted by a CFG

A phrase structure tree is **generated** by the CFG $\langle T, N, D, P\rangle$ iff

1. for each preterminal node with label A and (terminal) daughter with label t, $A \to t \in D$; and
2. for each nonterminal nonpreterminal node with label A and linearly ordered (as per Theorem 7.7) daughters with labels $A_0, \ldots, A_{n-1}$ respectively, $(n > 0)$, $A \to A_0 \ \ldots A_{n-1} \in P$.

Additionally, for a phrase structure tree with linearly ordered (as per Theorem 7.8) set of terminal nodes $a_0, \ldots, a_{n-1}$ with labels $t_0, \ldots, t_{n-1}$ respectively, the string $t_0 \ldots t_{n-1}$ is called the **terminal yield** of the phrase structure tree.

Appendix A

Deferred Proofs

From Chapter 4

Theorem 4.4 (Recursion Theorem (RT))**.** *Let X be a set, $x \in X$, and $F\colon X \to X$. Then there exists a unique function $h\colon \omega \to X$ such that*

$$h(0) = x, \text{ and} \tag{A.1}$$
$$\text{for every } n \in \omega,\ h(\mathbf{suc}(n)) = F(h(n)) \tag{A.2}$$

In the following proof sketch, the key idea is to define

$$\begin{aligned}\mathcal{A} = \{ & v : \omega \rightharpoonup X \mid \\ & v(0) = x \leftrightarrow 0 \in \mathsf{dom}(v) & \text{(a)} \\ & \wedge \forall n \in \omega(\mathbf{suc}(n) \in \mathsf{dom}(v) \to (n \in \mathsf{dom}(v) \wedge v(\mathbf{suc}(n)) = F(v(n)))) & \text{(b)} \\ \} \end{aligned}$$

and then define $h = \bigcup \mathcal{A}$.

The proof can be broken into four parts:

1. Show that h is (at least) a partial function.
2. Show that the two clauses in the definition of $\mathcal{A}$ hold when $h = v$.
3. Show that $\mathsf{dom}(h)$ is inductive.
4. Show that h is the only function from ω to X that satisfies the RT conditions.

Sketch of part 1. Let $S = \{n \in \omega \mid \text{for at most one } y \in X, \langle n, y\rangle \in h\}$. The trick is to show S is inductive and that therefor $S = \omega$ by PMI. $\square$

Sketch of part 2. We already know that $h : \omega \rightharpoonup X$. For (a), suppose $0 \in \mathsf{dom}(h)$. Then by definition of h, there is a $v : \omega \rightharpoonup X$ such that $v(0) = h(0)$. But $v(0) = x$. For (b), suppose $\mathbf{suc}(n) \in \mathsf{dom}(h)$. By definition of h again, there is a $v : \omega \rightharpoonup X$ such that $v(\mathbf{suc}(n)) = h(\mathbf{suc}(n))$. But v satisfies (b), so $n \in \mathsf{dom}(v)$ and $v(\mathbf{suc}(n)) = F(v(n))$. But by definition of h again, $h(n) = v(n)$. So $h(\mathbf{suc}(n)) = v(\mathbf{suc}(n)) = F(v(n)) = F(h(n))$. $\square$

Sketch of part 3. We need to show that $\mathsf{dom}(h)$ is inductive. Then $\mathsf{dom}(h) = \omega$ by PMI. $\square$

Sketch of part 4. Suppose two functions h and h' from ω to X satisfy (A.1) and (A.2). Then we need to show that

$$T = \big\{n \in \omega \mid h(n) = h'(n)\big\}$$

is inductive, and therefore $T = \omega$ by PMI. $\square$

From Chapter 5

Theorem 5.4 (Schröder-Bernstein)**.** *For any sets A and B, if $A \preceq B$ and $B \preceq A$, then $A \approx B$.*

Proof. By definition of $\preceq$, there are injections $f \colon A \to B$ and $g \colon B \to A$. Let C be the unique function from ω to $\wp(A)$ such that $C(0) = A \setminus \mathsf{ran}(g)$ and $C(n+1) = g[f[C(n)]]$ for all $n \in \omega$; henceforth we write C_n for $C(n)$. Now we define $h \colon A \to B$ such that $h(x) = f(x)$ if $x \in \bigcup_{n\in\omega} C_n$ and $h(x) = g^{-1}(x)$ otherwise; this makes sense since $\mathsf{ran}(g) = A \setminus C_0$. We will show h is bijective.

To show h is injective, suppose x and x' are distinct members of A; it suffices to show that $h(x) \neq h(x')$. Since f and g^{-1} are one-to-one, we need only consider the case where $x \in C_m$ and $x' \notin \bigcup_{n\in\omega} C_n$. Now we define $D_n =_{\text{def}} f[C_n]$ for all $n \in \omega$, so that $C_{n+1} = g[D_n]$. Then $h(x) = f(x)$, which is in D_m; but $h(x') = g^{-1}(x')$, which is *not* in D_m (since otherwise we would have $x' \in C_{m+1}$). So $h(x) \neq h(x')$, as desired.

To show h is surjective, let $y \in B$; we will show that $y \in \mathsf{ran}(h)$. Clearly, for each n, $D_n \subseteq \mathsf{ran}(()h)$. So we can assume $y \in B \setminus \bigcup_{n\in\omega} D_n$. Next, we note that, for all n, $g(y) \notin C_n$ (the proof, which is inductive, is left as an exercise). Therefore $g(y) \notin \bigcup_{n\in\omega} C_n$. So $h(g(y)) = g^{-1}(g(y)) = y$. So $y \in \mathsf{ran}(h)$. $\square$

Theorem 5.5. *If A is infinite, then $\omega \preceq A$.*

Proof. Let c be a choice function for A, and let h be the unique function from ω to $\wp(A)$ such that $h(0) = \emptyset$ and $h(n+1) = h(n) \cup \{c(A \setminus h(n))\}$ for all $n \in \omega$. Note for future reference that for any $m, n \in \omega$ with $m < n$, $h(m+1) \subseteq h(n)$. Also define $g\colon \omega \to A$ by $g(n) =_{\text{def}} c(A \setminus h(n))$, so that, for each $n \in \omega$, $h(n+1) = h(n) \cup \{g(n)\}$, and consequently also $g(n) \in h(n+1)$. Clearly, for all $n \in \omega$, $g(n) \notin h(n)$, since $g(n) = c(A \setminus h(n)) \in A \setminus h(n)$.

To complete the proof, we will show g is injective. So let m and n be distinct natural numbers; without loss of generality we can assume that $m < n$. Then $g(m) \in h(m+1)$, and so $g(m) \in h(n)$. But we already showed that $g(n) \notin h(n)$, so $g(m) \neq g(n)$; this shows g is injective as required. $\square$

Index