# Chapter 2

# Mathematical Preliminaries

## 2.1 Elements of Set Theory

Set theory is one of the fundamental tools that we shall use in formalizing machines, languages, and computation. The notion of a *tuple*, *relation*, of *function* shall appear in virtually every discussion. The concept of *cardinality* is also central, as we shall frequently wish to characterize the "size" of infinite sets. As we shall see, not all infinite sets are equivalent, which has some rather profound implications for computer science. For example, we can use this fact alone to show that there exist well-defined mathematical functions that cannot be computed by any conceivable computer or program.

Several sets are so fundamental that we adopt standard notation for them. In particular, we define the *natural numbers*, the *integers*, and the *rationals* by

$$\begin{aligned} \mathbb{N} &= \{0, 1, 2, \ldots\} \\ \mathbb{Z} &= \{\cdots -2, -1, 0, 1, 2, \ldots\} \\ \mathbb{Q} &= \{k/m \mid k \in \mathbb{Z} \ \wedge \ m \in \mathbb{N} - \{0\}\}, \end{aligned}$$

respectively. Similarly, we denote the set of all real numbers by $\mathbb{R}$. The empty set is denoted by $\emptyset$.

### 2.1.1 Tuples and Relations

An *ordered pair* consisting of $x$ and $y$ is denoted by $(x, y)$, where $x$ is referred to as the *first coordinate* and $y$ is the *second coordinate*. The ordered pair $(x, y)$ differs from the set $\{x, y\}$ in two crucial ways: 1) order is significant, so $(x, y)$ and $(y, x)$ are distinct whenever $x$ and $y$ are, and 2) the coordinates need not be distinct, so $(x, x)$ is a valid ordered pair, whereas $\{x, x\}$ has the same meaning as $\{x\}$. Two ordered pairs, $(a, b)$ and $(x, y)$, are equal if and only if $a = x$ and $b = y$. In purely set-theoretic terms, one could define the ordered pair $(x, y)$ to be the set $\{x, \{x, y\}\}$, which has the required properties. That is, it encodes ordering because $\{x, \{x, y\}\} \neq \{y, \{x, y\}\}$ when $x$ and $y$ are distinct,

and it also handles the case where the elements are identical, since $(x, x)$ corresponds to the set $\{x, \{x\}\}$, which is distinct from the *singleton set* $\{x\}$. We will not require this level of set-theoretic formalism, however, and will treat ordered pairs as legitimate mathematical objects in themselves.

An *n-tuple*, denoted by $(x_1, x_2, \ldots, x_n)$, is the obvious generalization of an ordered pair; here, the coordinates $x_1$, $x_2$, ..., $x_n$ may or may not be distinct. The word *tuple* is sometimes used instead of $n$-tuple.

If $A$ and $B$ are sets, the *Cartesian product* of $A$ and $B$, denoted by $A \times B$, is the set of all ordered pairs in which the first coordinate is from $A$ and the second coordinate is from $B$:

$$A \times B \stackrel{\text{def}}{=} \{(x, y) \mid x \in A, y \in B\}.$$

The Cartesian product generalizes to any number of sets. Thus, the $n$-fold Cartesian product

$$A_1 \times A_2 \times \cdots \times A_n \stackrel{\text{def}}{=} \{(a_1, \ldots, a_n) \mid a_i \in A_i\}$$

is a set of $n$-tuples, where the sets $A_1, A_2, \ldots, A_n$ may or may not be distinct. Here, the operators $\times \times \cdots \times$ are taken together to define a single $n$-ary operator rather than each forming a set of ordered pairs. In the special case where each $A_i$ is the same set $A$, we denote the $n$-fold Cartesian product by $A^n$. Since

$$(A \times B) \times C \stackrel{\text{def}}{=} \{\, ((a, b), c) \mid a \in A,\, b \in B,\, c \in C\},$$

which consists of ordered pairs whose first coordinate is another ordered pair, this is clearly distinct from both $A \times (B \times C)$ and $A \times B \times C$. However, we shall occasionally treat all three sets as consisting of three-tuples. That is, we shall ignore the extra level of parentheses when it is convenient to do so, which in effect defines a *natural isomorphism* among the different sets; that is, an obvious one-to-one correspondence between the elements of one and the elements of the other.

A *binary relation*, $R$, on sets $A$ and $B$ is simply a subset of the Cartesian product: That is, $R \subseteq A \times B$. An *n-ary relation* is a subset of the Cartesian product of $n$ sets. The *arity* of a relation is the number of sets in the Cartesian product, or equivalently, the number of coordinates in the resulting $n$-tuples. For example, consider the following two relations, which are subsets of $\mathbb{N}^2$ and $\mathbb{N}^3$, respectively:

$$\begin{aligned} L &\stackrel{\text{def}}{=} \{(x, y) \mid\ x + k = y \text{ for some } k > 0\,\}, \\ P &\stackrel{\text{def}}{=} \{(x, y, z) \mid x^2 + y^2 = z^2\}, \end{aligned}$$

where $x$, $y$, $z$, and $k$ are all natural numbers. Then $L$ is the binary "less than" relation, and $P$ is the ternary "Pythagorean" relation. In the case of binary relations, such as $L$, it is common practice to use *infix* notation, "$a\,L\,b$", to denote the fact that $(a, b) \in L$. Moreover, we often adopt a special symbol for a relation, such as "$<$" to denote the less than relation; so we write "$a < b$" instead of $(a, b) \in L$.

### 2.1.2 Functions and Partial Functions

Let $X$ and $Y$ be sets and let $f$ be a binary relation on $X$ and $Y$; that is, $f \subseteq X \times Y$. Then $f$ is called a *function* if for every $x \in X$, there is exactly one ordered pair $(a, b) \in f$ such that $a = x$.

When $f$ has this property, we often write $f : X \rightarrow Y$, and refer to $X$ as the *domain* of $f$ and $Y$ as the *co-domain* of $f$. We typically write $f(a) = b$ instead of $(a, b) \in f$. The notation $f(a)$ is convenient for functions because each element of the domain is associated with one unique element of the co-domain. By a *partial function* we mean a generalization of a function in which we allow zero or more elements of the domain to have *no* associated element in the co-domain; that is, we allow $f(a)$ to be *undefined* for zero or more $a \in A$. Note that every function therefore qualifies as a partial function as well; that is, it is a partial function that happens to be defined on all the elements of the domain. Thus, to make it clear when we are talking about functions in the traditional sense, we will often refer to them as *total functions*[1].

It is frequently useful to consider the action of a function $f : X \rightarrow Y$ on *subsets* of the domain $X$. For any subset $Z \subseteq X$, we define

$$f(Z) \stackrel{\text{def}}{=} \{f(z) \mid z \in Z\},$$

which is known as the *image of $Z$ under $f$*. A related concept is the set-valued *inverse image* with respect to the function $f$, which is defined by

$$f^{-1}(y) \stackrel{\text{def}}{=} \{x \in X \mid f(x) = y\}.$$

Note that $f^{-1}$ is a relation but in general *not* a function, since it may be one-to-many. That is, $f^{-1}(y)$ may consist of many elements, or may even be the empty set. Combining the above notations, we define

$$f^{-1}(Z) \stackrel{\text{def}}{=} \{x \in X \mid f(x) \in Z\},$$

for any subset $Z \subseteq Y$, which is called the *inverse image of $Z$ under $f$*.

A function $f : X \rightarrow Y$ is said to be *surjective* (or *onto*) if $f(X) = Y$. It is said to be *injective* (or *one-to-one*) if each $x \in X$ is mapped to a unique $y \in Y$. That is, $x \neq y$ implies that $f(x) \neq f(y)$, or equivalently, $f(x) = f(y)$ implies that $x = y$. Finally, $f$ is said to be *bijective* if it is both injective and surjective (one-to-one and onto). The inverse of $f : X \rightarrow Y$ is a function if and only if $f$ is a bijection. The sets $X$ and $Y$ are said to be in *one-to-one correspondence* if and only if a bijection exists from one to the other.

Two functions are equal if and only if they produce the same values for each element of the domain. Thus, we say that two functions on the same domain are *distinct* (i.e. not equal) if and only if there is at least one element at which they differ. That is, if $f : X \rightarrow Y$ and $g : X \rightarrow Z$, then $f \neq g$ if and only if $f(x) \neq g(x)$ for some $x \in X$.

### 2.1.3 Cardinality

If a set $S$ can be put into one-to-one correspondence with the set $\{1, 2, 3, \ldots, n\}$, then $n$ is said to be the *cardinal number* of $S$, and we write $|S| = n$. In other words, the *cardinality* of a set $S$ is its size. This notion also applies to infinite sets, but we need additional notation and terminology to talk about such sets. For example, if the set $A$ can be put into one-to-one correspondence with

[1] Technically, such a function would be a *total partial function*, as it is a special case of a partial function. However, that absurd-sounding terminology is usually abandoned in favor of the much more logical-sounding *total function*.

the *natural numbers*, $\mathbb{N} = \{0, 1, 2, \ldots\}$, then we say that $A$ is *countably infinite*. This is expressed in symbols as $|A| = \aleph_0$, where the symbol $\aleph$ (*aleph*) is from the Hebrew alphabet[2].

If two sets can be put into one-to-one correspondence, we say that they are are of the same cardinality, or *equinumerable*, and this is our criterion for declaring two sets to be of the "same size", whether finite or infinite. A set that is equinumerable with a subset of $\mathbb{N}$ (and possibly all of $\mathbb{N}$) is said to be *countable* (or *denumerable*); thus, all finite and countably infinite sets are countable. A surprising result of set theory, discovered by the mathematician Georg Cantor in the 19th century, is that there are *uncountable* (or *nondenumerable*) sets. For example, Cantor succeeded in showing that the continuum of real numbers, $\mathbb{R}$, is uncountable using a technique known as *diagonalization*. We state this as a theorem.

**Theorem 4** *The set of real numbers, $\mathbb{R}$, is uncountable. That is, $\mathbb{R}$ cannot be put into one-to-one correspondence with the natural numbers, $\mathbb{N}$.*

**Proof:** We will show that no function $f : \mathbb{N} \to \mathbb{R}$ can be bijective. We need only establish that every such function leaves out at least one element of $\mathbb{R}$, and thus fails to be onto. For a given $f$, we can construct an element of $\mathbb{R}$ that is not in the set $f(\mathbb{N}) = \{f(0), f(1), f(2), \ldots\}$ by *diagonalization*. Let $\delta(k, x)$ denote the $k$th binary digit of $x$ (after the binary point). Now let $y \in \mathbb{R}$ be any number such that its binary digits after the binary point satisfy

$$\delta(k, y) = \begin{cases} 1 & \text{if } \delta(k, f(k)) = 0 \\ 0 & \text{otherwise,} \end{cases}$$

for $k = 1, 2, 3, \ldots$. Then clearly $y \neq f(k)$ for any $k$, since the numbers differ in the $k$th digit after the binary point. Consequently, $y \notin f(\mathbb{N})$, showing that $f$ is not onto, and therefore not a bijection. □

It is useful to introduce several conventions for comparing the cardinalities of sets that continue to be meaningful when the sets are infinite. When the sets $A$ and $B$ are finite, it is clear what $|A| = |B|$ and $|A| < |B|$ mean, since we are simply comparing natural numbers. In particular, the former means that $A$ and $B$ have the same number of elements, and the latter means that $B$ has more elements. While these exact statements do not translate to infinite sets, the notation does if we are careful to define them in terms of functions. By definition, then

1. $|A| = |B|$ means that $A$ and $B$ are equinumerable. That is, there exists a bijection from $A$ to $B$ (and therefore from $B$ to $A$).
2. $|A| \leq |B|$ means that there exists an injection from $A$ to $B$, which may or may not be surjective.
3. $|A| < |B|$ means that there exists an injection from $A$ to $B$, but no such mapping is also a surjection. In other words, $f(A)$ is always a proper subset of $B$, for every function $f : A \to B$.

Note that these definitions hold for all sets, finite and infinite. In the case of finite sets, the meaning agrees exactly with the intuitive meaning (which is based on comparing natural numbers).

[2]The symbol $\aleph_0$ is called *aleph null*.

**Definition 1** The *power set* of a set $A$, denoted by either $\mathcal{P}(A)$ or $2^A$, is the set of all subsets of $A$; that is, $\mathcal{P}(A) \stackrel{\text{def}}{=} \{S \mid S \subseteq A\}$.

The notation $2^A$ requires some explanation; it actually denotes the set of all *functions* from the set $A$ to the set "2", that is, the set $\{0, 1\}$, which has two elements[3]. Because each function $f : A \to \{0, 1\}$ is naturally associated with a unique subset of $A$, namely the set $f^{-1}(1)$, there is an obvious one-to-one correspondence between the subsets of $A$ and the functions $f : A \to \{0, 1\}$. Consequently, it is common to use the notation $2^A$ to denote the power set of $A$. This idea also explains the notation $\mathbb{R}^2$, which is used to denote the set of ordered pairs of real numbers. This set is actually the set of all functions $f : \{0, 1\} \to \mathbb{R}$; ordered pairs result if we let 0 represent the first coordinate, and 1 represent the second coordinate.

**Theorem 5** *(Cantor's theorem) The power set of any set has a cardinality strictly greater than the original set: that is, $|\mathcal{P}(A)| > |A|$ for any set $A$, finite or infinite.*

**Proof:** Let $f : A \to 2^A$ be any given function. We shall use a form of diagonalization to show that some $S \in 2^A$ is left out of $f(A)$, so $f$ cannot possibly be surjective. In particular, consider the set

$$S \stackrel{\text{def}}{=} \{a \in A \mid a \notin f(a)\}.$$

By design, the set $S$ differs from the set $f(a)$ in at least one element for each $a \in A$. Consequently, $S \notin f(A)$, where $f(A)$ is the collection of all sets $f(a)$. Since $S \subseteq A$, and $S \notin f(A)$, we conclude that $f(A) \neq 2^A$. Thus, there exists no mapping from $A$ onto $2^A$, and hence no bijection. It follows that $A$ cannot be put into one-to-one correspondence with its own power set. □

Because the proof of theorem 5 makes no assumptions about the size of the original set $A$, we now have a way to construct a countably infinite sequence of infinite sets, each with a distinct cardinality; that is, by repeatedly taking the power set.

The above theorem revealed an anomaly in set theory known as *Cantor's paradox*, which prompted mathematicians to rethink the foundations of mathematics (and set theory in particular) around the turn of the century. Cantor noted that if the theorem is true for *all* sets, then it must hold for the "*set of all sets*." That is, the theorem implied that the power set of this massive set must be larger still. But how could this be? How could the power set contain *anything* that is not already in the set of *all* sets? This paradox was finally resolved (in several different ways) by essentially eliminating the troublesome notion of "the set of all sets," which was ultimately seen to be an ill-defined concept.

### 2.1.4 Explicit Bijections

The most direct method for establishing that two sets are equinumerable is to construct an explicit bijective mapping from one to the other (the direction being irrelevant). We now demonstrate this technique for several countably infinite sets.

[3]So, more formally, the notation should be $\{0, 1\}^A$.

**Theorem 6** *The set of all odd natural numbers has the same cardinality as* $\mathbb{N}$.

**Proof:** Define a function $f : \mathbb{N} \to \{1, 3, 5, \ldots\}$ by $f(n) = 2n + 1$. Then $f$ is surjective because $(n - 1)/2$ is a natural number for all $n \in \{1, 3, 5, \ldots\}$. Also, $f$ is injective because $f(i) = f(j)$ implies that $2i + 1 = 2j + 1$, or $i = j$. It follows that $f$ is a bijection, which implies that the sets are equinumerable. □

**Theorem 7** *The set of all integers,* $\mathbb{Z} = \{\cdots - 2, -1, 0, 1, 2, \ldots\}$*, is countable.*

**Proof:** Define a function $f : \mathbb{N} \to \mathbb{Z}$ by

$$f(n) = \frac{1 - (-1)^n(2n + 1)}{4}.$$

Observe that for any $k \in \mathbb{N}$, $f(2k) = -k$ and $f(2k + 1) = k + 1$. Thus, all even numbers are mapped to unique negative integers, and all odd numbers are mapped to unique positive integers. Furthermore, since every integer is the image of some natural number under $f$, it follows that $f$ is a bijection, so the sets are equinumerable. □

It is interesting to note that $\{1, 3, 5, \ldots\} \subset \mathbb{N} \subset \mathbb{Z}$, where the set containment is proper, yet all three sets are equinumerable. This counterintuitive behavior is in fact one of the distinguishing characteristics of infinite sets. We state this as a theorem:

**Theorem 8** *A set has an infinite number of elements if and only if it can be put into one-to-one correspondence with a proper subset of itself.*

As a final example, we now show that even Cartesian products of sets do not yield "bigger" sets in the sense of cardinality. This example also shows that it can be rather challenging to construct the desired bijection.

**Theorem 9** *The Cartesian product* $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$ *is countable.*

**Proof:** We shall prove this by constructing an explicit bijection $f$ that maps $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$ onto $\mathbb{N}$. Note that for the purposes of this proof we could just as well construct the mapping in the other direction, from $\mathbb{N}$ to $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$; the choice is simply a matter of convenience. Our construction will be broken into four steps.

**Step 1:** Partition the set $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$ into finite subsets defined by

$$P_n = \{(x, y, z) \in \mathbb{N} \times \mathbb{N} \times \mathbb{N} \mid x + y + z = n\}.$$

The sets $P_0, P_1, P_2, \ldots$ then partition the lattice of 3D integer points $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$ into a countable number of diagonal "slices," each containing only a finite number of points. Moreover, each point $(i, j, k)$ is in exactly one of the slices; namely $P_n$ where $n = i + j + k$.

**Step 2:** Assign the numbers $0, 1, 2, \ldots |P_n|-1$ to the elements of $P_n$ using some well-defined ordering; we shall use "lexicographic" ordering, which is easily defined on the integer triples. That is, we define $(x', y', z') < (x, y, x)$ to mean

$$\begin{array}{rl} x' < x & \text{or} \\ x' = x \text{ and } y' < y & \text{or} \\ x' = x, \;\; y' = y, \;\; \text{and} \;\; z' < z. & \end{array}$$

Of course, the last condition will never occur when comparing two points that belong to the same slice $P_n$ because the constraint $x + y + z = n$ ensures that two such points that agree in both $x$ and $y$ are actually identical.

Given a triple $(x, y, z) \in P_n$, what is its position in the ordered list? To compute this number, we count how many triples $(x', y', z')$ come before $(x, y, z)$ in the set $P_n$ with respect to this ordering. First, there are the cases in which $x' < x$. In each of these cases there are $n + 1 - x'$ possibilities for the remaining coordinates, $y'$ and $z'$, given that $x' + y' + z' = n$, and all coordinates must be natural numbers. Summing this expression for all $x' \in \{0, 1, 2, \ldots, x - 1\}$ we have

$$(n + 1) + n + (n - 1) + \cdots + (n + 2 - x) = \frac{x(2n - x + 3)}{2}.$$

In addition, there are $y$ possibilities for the case where $x' = x$ and $y' < y$, namely $y' \in \{0, 1, 2, \ldots, y - 1\}$. Therefore, the mapping

$$(x, y, z) \quad \rightarrow \quad \frac{x(2n - x + 3)}{2} + y$$

assigns a unique integer from the set $\{0, 1, 2, \ldots, |P_n| - 1\}$ to each triple in the set $P_n$.

**Step 3:** We now shift the counts within each of the sets $P_n$ so that they are distinct from one another. Since all the sets are finite, we need only offset the elements of a given $P_n$ by the number of entries in all the previous sets, $P_0, P_1, \ldots, P_{n-1}$, which is simply the number of triples $(x, y, z)$ that satisfy $x+y+z < n$. We can compute this number directly by thinking of it as the number of distinct ways to place $n - 1$ "balls" into four "urns", three of which represent $x$, $y$, and $z$. Equivalently this is the number of distinct permutations of $n - 1$ balls and three "partitions," where each contiguous sequence of balls represents the contents of one urn. This number is given by the binomial coefficient

$$\binom{(n - 1) + 3}{3} = \frac{(n + 2)(n + 1)n}{6}.$$

**Step 4:** The bijection $f : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ is then simply the sum of the above terms,

$$f(x, y, z) \;\; = \;\; \frac{(n + 2)(n + 1)n}{6} + \frac{x(2n - x + 3)}{2} + y,$$

where $x$, $y$, and $z$ are arbitrary natural numbers, and $n = x + y + z$. □

### 2.1.5 The Schröder-BernsteinTheorem

Constructing an explicit bijection between two equinumerable sets can be very tedious, as the previous example shows. Fortunately the *Schröder-Bernsteintheorem* affords a much easier way to establish equinumerability, requiring only the existence of *injections*, not bijections. Before stating and proving this theorem, we first introduce some new terminology and prove several simpler theorems.

**Theorem 10** *Let $A$, $B$, and $C$ be sets such that $A \supseteq B \supseteq C$. If there exists a bijection $f : A \rightarrow C$, then there also exists a bijection $g : A \rightarrow B$.*

**Proof:** (See appendix.)

We now state and prove two different but equivalent versions of the Schröder-Bernsteintheorem, since both forms are convenient. The second form follows easily from the first.

**Theorem 11 (Schröder-Bernstein, I)** *Let $A$ and $B$ be arbitrary sets. If there exist injections $f : A \rightarrow B$ and $g : B \rightarrow A$ then there exists a bijection from $A$ to $B$.*

**Proof:** Let $A' = g(B)$, $B' = f(A)$, and $A'' = g(B')$, as depicted in Figure A.1. Then the function $g \circ f$ is a bijection from $A$ onto $A''$. Since $A \supseteq A' \supseteq A''$, by theorem 10 we may assert the existence of a bijection $h : A \rightarrow A'$. Also, since $g : B \rightarrow A'$ is a bijection, so is the inverse $g^{-1} : A' \rightarrow B$. It follows that $g^{-1} \circ h$ is a bijection from $A$ to $B$, which establishes the result. □

**Theorem 12 (Schröder-Bernstein, II)** *Let $A$ and $B$ be arbitrary sets. If $|A| \leq |B|$ and $|B| \leq |A|$ then $|A| = |B|$.*

**Proof:** If $|A| \leq |B|$ and $|B| \leq |A|$, then by definition there exist injections $f : A \rightarrow B$ and $g : B \rightarrow A$. The result now follows from theorem 11. □

Using the Schröder-Bernsteintheorem, we can now give a very simple proof of theorem 9. We need only demonstrate that there exist injections $f : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, and $g : \mathbb{N} \rightarrow \mathbb{N} \times \mathbb{N} \times \mathbb{N}$. The following functions suffice:

$$\begin{aligned} f(i,j,k) &= 2^i\, 3^j\, 5^k, \\ g(i) &= (i,0,0). \end{aligned}$$

It is easy to see that both of these functions are injective, so the result is established. Note that any three relatively prime numbers suffice for the construction of the function $f$ above; this follows from the fact that every integer has a unique prime factorization. This is a common trick used to define injections from $\mathbb{N}^k$ to $\mathbb{N}$. Essentially the same strategy can be used to prove the countability of the rational numbers, as we now show.

**Theorem 13** *The set of rational numbers, $\mathbb{Q}$, is countable.*

**Proof:** For every $q \in \mathbb{Q}$, there exist unique integers $k$ and $m$ such that $k \geq 0$, $k$ and $m$ are relatively prime, and $q = k/m$. We use this fact to assign a unique natural number to each rational number. Define $f : \mathbb{Q} \rightarrow \mathbb{N}$ by

$$f(q) = \begin{cases} 2^k\, 3^m & \text{if } m > 0 \\ 2^k\, 5^{-m} & \text{otherwise,} \end{cases}$$

where $q = k/m$, $k \geq 0$, and $k$ and $m$ are relatively prime (i.e. the fraction is in lowest terms). Because $k$ and $m$ are unique, $f$ is an injection from $\mathbb{Q}$ to $\mathbb{N}$. Because $\mathbb{N} \subset \mathbb{Q}$, the identity map is an injection in the other direction. By theorem 11, these injections imply that the sets are equinumerable. □

## 2.2 Elements of Logic

In this section we review the basic concepts of propositional (or Boolean) logic. Propositional formulas (also known as *Boolean* formulas) play a very fundamental role in computer science. As we shall see, determining whether such a formula can in fact be *satisfied* is an intractable problem, and lies at the foundation of the theory of NP-Completeness. We shall also find the formalism of propositional logic to be useful in other contexts as well, such as in formulating precise statements and proving theorems. For more extensive discussions of propositional logic see Nerode and Shore [**?**] and Davis et al. [**?**].

A *proposition* is a statement that is either true or false, such as "the moon is made of green cheese". Propositional logic deals with symbolic representations of propositions and relationships among them. The propositions themselves are typically denoted by single letters, such as $x$, $y$, $z$, often with subscripts, which are called *boolean variables* or *propositional letters*.[4] We shall denote the *truth values* "true" and "false" by the symbols $\top$ and $\bot$, respectively. Propositional *formulas* are built up by combining boolean variables and constants using parentheses and *logical connectives*, which are summarized below.[5]

1. "$\neg$" is negation
2. "$\wedge$" is conjunction, which means "and"
3. "$\vee$" is disjunction, which means "inclusive or"
4. "$\supset$" is material implication

[4]The terminology *propositional letters* emphasizes the fact that boolean variables are *atomic* in that their component symbols carry no independent meaning. Thus, even though boolean variables are frequently comprised of multiple symbols, such as subscripts, they can nevertheless be thought of as single symbols, or letters, within a finite alphabet.

[5]The symbols that we use to denote the logical connectives is fairly common but, unfortunately, there is no universally accepted notation. For example, one sometimes sees conjunction denoted by "$\cdot$" or "&", and negation denoted by "$\tilde{}$" or "$-$". It is also quite common to see either $\longrightarrow$ or $\Longrightarrow$ used instead of $\supset$ for material implication, and either $\longleftrightarrow$ or $\Longleftrightarrow$ used instead of $\equiv$ for material equivalence.

5. "$\equiv$" is material equivalence

All of the connectives are binary relations (i.e. taking two arguments) except for negation, which is a unary relation. The modifier "material" used for the implication and equivalence connectives above affirms that the connectives are concerned only with the truth values of the formulas they relate, not with their forms or their meanings; this is so with all of the connectives. Material implication and equivalence are therefore weaker than the corresponding notions that one might employ in every-day speech. For example, the statement "circles are round implies that six is even" is true if "implies" is taken in the material sense, since both the premise and the consequent are true. However, the evenness of six is in no way a *consequence* of the roundness of circles, so we would not assert that one "implies" the other according to the common usage of the word.

Here are some examples of boolean formulas:

$$(x \wedge y) \equiv (\neg x \vee \neg y) \tag{2.1}$$
$$(x \equiv y) \supset (\neg x \vee y) \tag{2.2}$$
$$(x \vee \neg y) \supset ((x \wedge y) \vee \neg(\neg x \wedge y)) \tag{2.3}$$
$$x_1 \wedge (x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_4 \vee \neg x_2) \tag{2.4}$$

Each of the formulas above contains a number of *sub-formulas*. Informally, a sub-formula is a well-formed formula (wff) appearing within a formula. For instance, formula (2.3) contains the sub-formulas $x$, $\neg y$, $x \vee \neg y$, $x \wedge y$, etc. We can define precisely what a *well-formed* boolean formula is (modulo some redundant parentheses) as follows. Let $V$ be a set of boolean variables; this may be an infinite set. Then the set $\mathcal{W}$ of well-formed propositional formulas over $V$ is defined by the following rules:

1. $V \subset \mathcal{W}$
2. $u \in \mathcal{W} \implies (\neg u) \in \mathcal{W}$
3. $u, v \in \mathcal{W} \implies (u \circ v) \in \mathcal{W}$, where $\circ$ is any of the binary logical connectives
4. No other strings of symbols are in $\mathcal{W}$.

Note that we are using "$\implies$" as a *meta-symbol* that also denotes implication, which in this context is an assertion; that is, a statement that is necessarily true.[6] Similarly, the meta-symbol "$\Longleftrightarrow$" is frequently used as an assertion that two boolean formulas are equivalent; that is, they produce the same values given the same truth assignments for their variables. For example, $x \vee \neg x \Longleftrightarrow \top$. The meta-symbols "$\implies$" and "$\Longleftrightarrow$" are never *part* of a boolean formula; rather, they are used to make statements *about* boolean formulas (hence the modifier "meta-").

It is customary to omit many of the parenthesis that the above rules would include. For example, we normally do not include parentheses enclosing the entire formula, so we write $x \wedge y$ rather than $(x \wedge y)$. Also, because of the associativity of both conjunction and disjunction (see section 2.2.2), it is

[6] Conventions vary as to which symbols are the connectives and which are the meta-symbols. (See footnote 5.) When examining unfamiliar literature, be sure to first understand all such conventions that are employed.

customary to omit the parentheses in formulas such as $(x \wedge y) \wedge (z \wedge w)$, and write $x \wedge y \wedge z \wedge w$ instead. Similarly, we write $\neg\neg x$ rather than $(\neg(\neg x))$, and by assuming that $\neg$ has higher precedence than the other connectives, we may write $\neg x \wedge \neg y$ rather than $(\neg x) \wedge (\neg y)$. Note that we have employed these conventions in formulas (2.1) through (2.4).

The rules defined in the previous section specify only the *syntax*, or "form," of boolean formulas, not their meanings. To specify the meanings, we first need the meanings of the logical connectives themselves, which can be completely defined by a *truth table*. The following truth table specifies the values resulting from each of the connectives given all possible truth values of the arguments, which appear in the two left-most columns. Here we have also included definitions for the symbols "$\uparrow$" and "$\downarrow$," which denote *nand* (a contraction for "not and") and *nor* (a contraction for "not or"), respectively.[7]

| $x$ | $y$ | $\neg x$ | $x \wedge y$ | $x \vee y$ | $x \supset y$ | $x \equiv y$ | $x \uparrow y$ | $x \downarrow y$ |
|---|---|---|---|---|---|---|---|---|
| $\bot$ | $\bot$ | $\top$ | $\bot$ | $\bot$ | $\top$ | $\top$ | $\top$ | $\top$ |
| $\bot$ | $\top$ | $\top$ | $\bot$ | $\top$ | $\top$ | $\bot$ | $\top$ | $\bot$ |
| $\top$ | $\bot$ | $\bot$ | $\bot$ | $\top$ | $\bot$ | $\bot$ | $\top$ | $\bot$ |
| $\top$ | $\top$ | $\bot$ | $\top$ | $\top$ | $\top$ | $\top$ | $\bot$ | $\bot$ |

The truth values of more complex formulas are found by repeated application of these definitions beginning with the values of the variables themselves. In fact, truth tables can also be used to determine the value of a boolean formula as a function of its variables. For example, consider formula (2.1). We first list each sub-formula across the top of the table, organizing them in such a way that every sub-formula of an expression appears in a column to its left. Then we can easily fill in the table, working left-to-right, and top-to-bottom, by applying the definitions of the individual logical connectives. We obtain the following table:

| $x$ | $y$ | $x \wedge y$ | $\neg x$ | $\neg y$ | $\neg x \vee \neg y$ | $(x \wedge y) \equiv (\neg x \vee \neg y)$ |
|---|---|---|---|---|---|---|
| $\bot$ | $\bot$ | $\bot$ | $\top$ | $\top$ | $\top$ | $\bot$ |
| $\bot$ | $\top$ | $\bot$ | $\top$ | $\bot$ | $\top$ | $\bot$ |
| $\top$ | $\bot$ | $\bot$ | $\bot$ | $\top$ | $\top$ | $\bot$ |
| $\top$ | $\top$ | $\top$ | $\bot$ | $\bot$ | $\bot$ | $\bot$ |

In this particular instance, we see that formula (2.1), which appears above the right-most column of the truth table, is *unsatisfiable*, as all of the resulting values are false. When one or more of the entries in a column is true, the formula is called *satisfiable*. For instance, in the table above we see that each of the sub-formulas is satisfiable. Using similar truth tables it is easy to verify that formulas (2.2) and (2.3) are *valid*, which means that they always result in true regardless of the values assigned to their variables, and that formula (2.4) is satisfiable but not valid.

Another more compact way to construct a truth table is to associate a column of truth values with each variable and connective in a single formula. After assigning all possible truth values (consistently) to the variables, we then fill in the columns associated with the connectives. For example, such a truth table for formula (2.2) would look like this:

[7] Another symbol commonly used in the literature for nand is the vertical bar, as in $(x|y)$.

| ( | $x$ | $\equiv$ | $y$ | ) | $\supset$ | ( | $\neg$ | $x$ | $\vee$ | $y$ | ) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\bot$ | $\top$ | $\bot$ | | $\top$ | | $\top$ | $\bot$ | $\top$ | $\bot$ | |
| | $\bot$ | $\bot$ | $\top$ | | $\top$ | | $\top$ | $\bot$ | $\top$ | $\top$ | |
| | $\top$ | $\bot$ | $\bot$ | | $\top$ | | $\bot$ | $\top$ | $\bot$ | $\bot$ | |
| | $\top$ | $\top$ | $\top$ | | $\top$ | | $\bot$ | $\top$ | $\top$ | $\top$ | |

This truth table demonstrates that formula (2.2) is valid, since the *principal connective*, the one that is applied last ($\supset$ in this case), always results in $\top$. (See Carnap [**?**] for further discussion of this type of truth table.)

### 2.2.1 Truth Assignments and Satisfiability

A *truth assignment* is a function $\mathcal{V} : V \rightarrow \{\top, \bot\}$, where $V$ is a set of boolean variables. That is, $\mathcal{V}$ associates a truth value (true or false) with each variable in $V$. We say that a truth assignment $\mathcal{V}$ is *appropriate* for a boolean formula $\mathcal{F}$ if every variable that appears in $\mathcal{F}$ is assigned a truth value by $\mathcal{V}$. The meaning of $\mathcal{V}$ can be extended to boolean formulas over $V$ by means of the following definitions:

1. $\mathcal{V}(u \wedge v) = \begin{cases} \top & \text{if } \mathcal{V}(u) = \top \text{ and } \mathcal{V}(v) = \top \\ \bot & \text{otherwise} \end{cases}$

2. $\mathcal{V}(u \vee v) = \begin{cases} \top & \text{if } \mathcal{V}(u) = \top \text{ or } \mathcal{V}(v) = \top \\ \bot & \text{otherwise} \end{cases}$

3. $\mathcal{V}(\neg u) = \begin{cases} \bot & \text{if } \mathcal{V}(u) = \top \\ \top & \text{otherwise} \end{cases}$

4. $\mathcal{V}(u \supset v) = \begin{cases} \bot & \text{if } \mathcal{V}(u) = \top \text{ and } \mathcal{V}(v) = \bot \\ \top & \text{otherwise} \end{cases}$

5. $\mathcal{V}(u \equiv v) = \begin{cases} \top & \text{if } \mathcal{V}(u) = \mathcal{V}(v) \\ \bot & \text{otherwise} \end{cases}$

To handle extra parentheses we could, of course, introduce another explicit rule: $\mathcal{V}((u)) = \mathcal{V}(u)$. This provides a convenient formal definition for the truth value of a boolean formula given the truth values of the underlying variables. Think of these rules as providing an algorithmic alternative to a truth table; indeed, each of these rules corresponds very closely to a function that one would use, say in a Lisp environment, to recursively evaluate a boolean formula.

With the notion of a truth assignment thus extended to arbitrary boolean formulas, we identify a number of important special types of formulas, each with its own terminology and notation. These types are summarized below, where $\mathcal{F}$ denotes an arbitrary boolean formula, $\mathcal{V}$ denotes a truth assignment that is appropriate for $\mathcal{F}$, and "$\models$" and "$\not\models$" are new meta-symbols. In several cases, the terminology column lists multiple phrases that are used synonymously.

| terminology | notation | meaning |
|---|---|---|
| $\mathcal{V}$ *verifies* $\mathcal{F}$<br>$\mathcal{V}$ *satisfies* $\mathcal{F}$<br>$\mathcal{V}$ is a *model* for $\mathcal{F}$ | $\mathcal{V} \models \mathcal{F}$ | $\mathcal{V}(\mathcal{F}) = \top$ |
| $\mathcal{V}$ *falsifies* $\mathcal{F}$ | $\mathcal{V} \not\models \mathcal{F}$ | $\mathcal{V}(\mathcal{F}) = \bot$ |
| $\mathcal{V}$ is *satisfiable* | | $\mathcal{V}(\mathcal{F}) = \top$ for *some* truth assignment $\mathcal{V}$ |
| $\mathcal{F}$ is *falsifiable* | | $\mathcal{V}(\mathcal{F}) = \bot$ for *some* truth assignment $\mathcal{V}$ |
| $\mathcal{F}$ is *contingent* | | $\mathcal{V}_1(\mathcal{F}) = \top$ and $\mathcal{V}_2(\mathcal{F}) = \bot$ for some truth assignments $\mathcal{V}_1$ and $\mathcal{V}_2$ |
| $\mathcal{F}$ is a *tautology*<br>$\mathcal{F}$ is *valid* | $\models \mathcal{F}$ | $\mathcal{V}(\mathcal{F}) = \top$ for *all* truth assignments $\mathcal{V}$ |
| $\mathcal{F}$ is a *contradiction*<br>$\mathcal{F}$ is *unsatisfiable* | $\not\models \mathcal{F}$ | $\mathcal{V}(\mathcal{F}) = \bot$ for *all* truth assignments $\mathcal{V}$ |

Note that a *contingent* formula is one that is both satisfiable and falsifiable, or equivalently, neither a tautology nor a contradiction. Using the new notation, we can now precisely state the relationship between the logical connectives $\supset$ and $\equiv$, and their corresponding meta-symbols $\implies$ and $\iff$ when the latter are applied to boolean formulas. Specifically, if $u$ and $v$ are boolean formulas,

1. We write $u \implies v$ if and only if $\models (u \supset v)$
2. We write $u \iff v$ if and only if $\models (u \equiv v)$

Thus, the meta-symbols implicitly refer to all possible truth assignments appropriate to the formulas involved. For this reason meta-symbols are frequently used in the statement of theorems, which typically assert the validity of some implication or equivalence. For example, the transitivity of material implication is asserted as

$$(x \supset y) \wedge (y \supset z) \implies (x \supset z), \tag{2.5}$$

which is equivalent to asserting that

$$\models ((x \supset y) \wedge (y \supset z)) \supset (x \supset z). \tag{2.6}$$

This latter fact can be easily demonstrated using a truth table.

### 2.2.2 Elementary Theorems

Listed below are a number of fundamental properties of the logical connectives. These properties allow us to perform certain manipulations on boolean formulas to obtain equivalent (often simpler) formulas. Each of these properties can be easily verified by replacing the meta-symbol by the corresponding logical connective (e.g. replace $\iff$ with $\equiv$) and then using a truth table to establish the validity of the resulting formula.

1. $x \wedge y \iff y \wedge x$ (commutativity of conjunction)
2. $x \vee y \iff y \vee x$ (commutativity of disjunction)
3. $x \iff (x \wedge x)$ (idempotence of conjunction)
4. $x \iff (x \vee x)$ (idempotence of disjunction)
5. $x \equiv y \iff y \equiv x$ (commutativity of material equivalence)
6. $(x \wedge y) \wedge z \iff x \wedge (y \wedge z)$ (associativity of conjunction)
7. $(x \vee y) \vee z \iff x \vee (y \vee z)$ (associativity of disjunction)
8. $x \wedge (y \vee z) \iff (x \wedge y) \vee (x \wedge z)$ (distributivity of conjunction over disjunction)
9. $x \vee (y \wedge z) \iff (x \vee y) \wedge (x \vee z)$ (distributivity of disjunction over conjunction)
10. $\neg(x \wedge y) \iff \neg x \vee \neg y$ (De Morgan's law)
11. $\neg(x \vee y) \iff \neg x \wedge \neg y$ (De Morgan's law)
12. $\neg\neg x \iff x$ (law of double negation)
13. $x \supset y \iff \neg x \vee y$ (principle of material implication)
14. $x \equiv y \iff (x \wedge y) \vee (\neg x \wedge \neg y)$ (principle of material equivalence)
15. $(x \supset y) \wedge (y \supset z) \implies (x \supset z)$ (transitivity of material implication)
16. $(x \equiv y) \wedge (y \equiv z) \implies (x \equiv z)$ (transitivity of material equivalence)
17. $(x \supset y) \wedge (y \supset x) \iff (x \equiv z)$ (double implication is equivalence)
18. $(x \wedge y) \supset z \iff x \supset (y \supset z)$ (principle of exportation)
19. $(x \supset y) \iff (\neg y \supset \neg x)$ (principle of transposition for material implication)
20. $(x \equiv y) \iff (\neg x \equiv \neg y)$ (principle of transposition for material equivalence)

Note that material implication and material equivalence can always be replaced by equivalent formulas containing only the connectives $\wedge$, $\vee$, and $\neg$; thus, the material connectives are included only as a convenience, not for their expressive power.

### 2.2.3 Normal Forms

A boolean formula is said to be in *conjunctive normal form*, abbreviated CNF, if it is expressed as a conjunction of zero or more disjunctions; that is, a conjunction of zero or more *clauses*, where each clause is itself a disjunction of zero or more literals. A *literal* is a boolean variable, the negation of a boolean variable, or one of the constants $\top$ or $\bot$. An example of a formula in CNF form is

$$(x \vee \neg y \vee z) \wedge (\neg x \vee \neg w) \wedge w \wedge \neg x \tag{2.7}$$

Note that all negations in a CNF formula appear in front of variables, not in front of the clauses or the entire formula. It is customary to denote a CNF formula in *clause set* form, where all parentheses and logical connectives are removed, and negation is represented by bars over the variables. For example, formula (2.7) would be represented by the clause set

$$\{\,\{x, \overline{y}, z\}\,, \{\overline{x}, \overline{w}\}\,, \{w\}\,, \{\{\overline{x}\}\,\}\,. \tag{2.8}$$

This is a very appropriate representation, since it emphasizes the fact that clause ordering is irrelevant, as is the ordering of literals within a clause. Set notation is also consistent with the fact that both repeated literals and repeated clauses are unnecessary, and can be removed without altering the meaning of the formula.

A *conjunction* of literals is satisfied if and only if *all* conjuncts are satisfied, and a *disjunction* of literals is satisfied if and only if *at least one* disjunct is satisfied. Thus, an empty clause is a contradiction, since none of its literals can be satisfied. Similarly, an empty clause set is a tautology, since all of its clauses are satisfied. In other words, the clause set $\emptyset$ is always satisfied, and the clause set $\{\emptyset\}$ is unsatisfiable. It is customary to denote the empty clause by $\square$; thus, the formula $\{\square\}$ is unsatisfiable, as is $\{\{x, y, \overline{z}\}\,, \square\}$, and any other clause set that contains $\square$.

Every boolean formula is equivalent to another formula over the same variables that is in CNF form. To see this, observe that we can "transform" any given boolean formula into CNF form, preserving equivalence at each step, by performing four types of transformations, repeating each one until it can no longer be applied then moving on to the next.

1. If there is a connective other than $\neg$, $\wedge$, or $\vee$, replace it with an equivalent sub-formula that includes only these basic connectives using rules (13) and (14).
2. Distribute a negation symbol across a conjunction or a disjunction using De Morgan's laws, which are rules (10) and (11).
3. If there is a repeated negation, reduce it using the law of double negation, rule (12).
4. Distribute a disjunction across a conjunction using rule (9).

When the final step can no longer be applied, the formula will be in CNF form; all negation signs will be adjacent to variables, and all conjunctions will join either literals or disjunctions of literals. The resulting formula will be equivalent to the original formula because each rule preserves equivalence.

A boolean formula is in *disjunctive normal form* (DNF) if it consists of a disjunction of conjunctions. For example, the formula

$$(\neg b) \vee (a \wedge b \wedge \neg c) \vee (\neg a \wedge d),$$

is in DNF form. Here the lists of conjunctions are called *dual clauses* to distinguish them from the clauses of CNF formulas. Let DNF-SAT denote the language consisting of satisfiable boolean formulas that are in DNF form. Show that DNF-SAT is in **P**, the class of problems that can be decided in polynomial time.