# Background for Discrete Mathematics

Huck Bennett
Northwestern University

These notes give a terse summary of basic notation and definitions related to three topics in discrete mathematics: logic, sets, and functions. The notes target students taking Northwestern University's EECS 212 course.

## 1 Logic

### 1.1 Propositional logic

*Boolean variables* (also called *propositional variables*) are variables which are Boolean-valued, i.e., are either 'true' or 'false.' (Sometimes these values are shortened to 'T' and 'F', or to '0' and '1.') A *logical connective* is a function that takes one or more Boolean values as arguments, and outputs a Boolean value. For example, "AND," "OR," and "NOT" (written in symbols as $\wedge$, $\vee$, and $\neg$) are logical connectives. Table 1 summarizes these and several other of the most commonly used logical connectives.

| **Name** | **Notation** | **Description** |
|---|---|---|
| OR | $x \vee y$ | $x \vee y$ is true if and only if at least one of $x$ and $y$ is true. |
| AND | $x \wedge y$ | $x \wedge y$ is true if and only if both $x$ and $y$ are true. |
| NOT | $\neg x$ | $\neg x$ is true if and only if $x$ is false. |
| XOR | $x \oplus y$ | $x \oplus y$ is true if and only if exactly one of $x$ and $y$ is true. |
| IMPLIES | $x \Rightarrow y$ | $x \Rightarrow y$ is true if and only if $y$ is true or $x$ is false. |
| IFF (if and only if) | $x \Leftrightarrow y$ | $x \Leftrightarrow$ is true if and only if $x = y$. |

Table 1: Standard logical connectives. The truth tables for these functions appear in Table 2.

A *propositional formula* is either a Boolean variable, or a logical connective applied to the correct number of arguments, each of which is itself a propositional formula. For example, $x \wedge (\neg y \vee z)$ is a propositional formula over the Boolean variables $x$, $y$, and $z$ since $\wedge$ takes two arguments, and $x$ and $\neg y \vee z$ are themselves propositional formulas. If the variables in a propositional formula are instantiated with Boolean values, then the formula evaluates to either true or false.

A *truth table* is a list of what values a propositional formula evaluates to for each possible setting of its input variables. Table 2 gives truth tables for commonly used logical connectives, which match their descriptions in Table 1.

| $x$ | $y$ | $x \vee y$ |
|---|---|---|
| $F$ | $F$ | $F$ |
| $T$ | $F$ | $T$ |
| $F$ | $T$ | $T$ |
| $T$ | $T$ | $T$ |

| $x$ | $y$ | $x \wedge y$ |
|---|---|---|
| $F$ | $F$ | $F$ |
| $T$ | $F$ | $F$ |
| $F$ | $T$ | $F$ |
| $T$ | $T$ | $T$ |

| $x$ | $\neg x$ |
|---|---|
| $F$ | $T$ |
| $T$ | $F$ |

| $x$ | $y$ | $x \oplus y$ |
|---|---|---|
| $F$ | $F$ | $F$ |
| $T$ | $F$ | $T$ |
| $F$ | $T$ | $T$ |
| $T$ | $T$ | $F$ |

| $x$ | $y$ | $x \Rightarrow y$ |
|---|---|---|
| $F$ | $F$ | $T$ |
| $T$ | $F$ | $F$ |
| $F$ | $T$ | $T$ |
| $T$ | $T$ | $T$ |

| $x$ | $y$ | $x \Leftrightarrow y$ |
|---|---|---|
| $F$ | $F$ | $T$ |
| $T$ | $F$ | $F$ |
| $F$ | $T$ | $F$ |
| $T$ | $T$ | $T$ |

Table 2: Truth tables for the logical connectives in Table 1. Top row: 'OR' (left), 'AND' (center), and 'NOT' (right). Bottom row: 'XOR' (left), 'IMPLIES' (center), and 'IFF' (right).

## 1.2 First-order logic[1]

Formulas in first-order logic include all propositional formulas, and also include formulas that contain *quantifiers* and *predicates*. The two main quantifiers are the universal quantifier *for all*, written as '$\forall$', and the existential quantifier *there exists*, written as '$\exists$.' A predicate $P$ is a function whose output is Boolean-valued. If $P(x)$ is true for some input $x$, then we say that $x$ *satisfies* $P$.

A typical formula in first-order logic uses quantifiers and predicates to assert which elements in a set meet a given condition. Syntactically, if $X$ is a set and $P(x)$ is a predicate, then we write "$\forall x \in X,\ P(x)$" to assert that $P(x)$ is true for all $x \in X$, and similarly write "$\exists x \in X,\ P(x)$" to assert that there exists $x \in X$ for which $P(x)$ is true.

The equality, disequality, and inequality symbols $=$, $\neq$, $<$, $>$, $\leq$, and $\geq$ are all predicates that take two real numbers as input. So, for example, the formula $\forall n \in \mathbb{N},\ n \geq 0$ asserts that all natural numbers are at least zero, and the formula $\exists z \in \mathbb{Z},\ z \geq 0$ asserts that there exists an integer that is at least 0. Both of these formulas are true. On the other hand, the formula $\forall z \in \mathbb{Z},\ z \geq 0$ is a well-formed formula in first-order logic, but is false because it incorrectly asserts that all integers are nonnegative.

Formulas may also contain multiple quantifiers. For example, we can assert that there exists a real number strictly between any two distinct integers $z_1, z_2$ with $z_1 < z_2$ as follows:

$$\forall z_1 \in \mathbb{Z},\ \forall z_2 \in \mathbb{Z},\ (z_1 < z_2) \Rightarrow (\exists r \in \mathbb{R},\ (r > z_1) \wedge (r < z_2))\ . \tag{1}$$

To understand this formula, it may be helpful to think about what it is saying for a particular instantiation of its variables. For example, it asserts that if we pick integers $z_1 = 4$ and $z_2 = 5$, then because $z_1 < z_2$ we can always find a real number, such as $r = 4.237$, which satisfies $r > z_1$ and $r < z_2$.

It is always possible to write a formula in first-order logic with the quantifiers at the beginning, and this will often be most convenient. Indeed, the formula in Equation (1) is equivalent to the following formula, which has all of its quantifiers at the beginning:

$$\forall z_1 \in \mathbb{Z},\ \forall z_2 \in \mathbb{Z},\ \exists r \in \mathbb{R},\ (z_1 < z_2) \Rightarrow ((r > z_1) \wedge (r < z_2))\ .$$

Given predicates $P$ and $Q$, we define the *contrapositive* of the formula $P(x) \Rightarrow Q(x)$ to be $\neg Q(x) \Rightarrow \neg P(x)$. A given value $x$ satisfies $P(x) \Rightarrow Q(x)$ if and only if it satisfies $\neg Q(x) \Rightarrow \neg P(x)$. In other words, if the domain of $P$ and $Q$ is $X$, it holds that $\forall x \in X,\ (P(x) \Rightarrow Q(x)) \Leftrightarrow (\neg Q(x) \Rightarrow \neg P(x))$. It is straightforward to prove this by treating $P$ and $Q$ as propositional variables and observing that the truth tables of $P \Rightarrow Q$ and $\neg Q \Rightarrow \neg P$ are the same.

[1] This section depends on Sections 2 and 3.

# 2 Sets

A *set* is an unordered collection of *elements* in which each element appears at most once. A set may be empty, finite, or infinite. One may define a set in several ways:

1. Explicitly. We write a set explicitly as a comma-delimited list of elements between curly braces. For example, the set of schools in the Big Ten West Division is

$$Y := \{\text{Illinois}, \text{Iowa}, \text{Minnesota}, \text{Nebraska}, \text{Northwestern}, \text{Purdue}, \text{Wisconsin}\}\ .^2$$

2. In terms of operations such as union, intersection, and complement applied to other sets (see the operations in Table 4). For example, if $Y$ is the set of schools in the Big Ten West Division defined above, and $Z$ is the set of private universities in the United States, then the set of private universities in the Big Ten West Division is

$$Y \cap Z = \{\text{Northwestern}\}\ .$$

3. Using *set builder notation.* Set builder notation is a way to define a set as the subset of elements in a larger set that satisfy some predicate. Given a set $X$ and a predicate $P(x)$, we denote the set of all elements $x \in X$ that satisfy $P(x)$ by $\{x \in X : P(x)\}$, which we read as "the set of elements $x$ in $X$ such that $P(x)$ is true." Set builder notation is closely related to the concept of *list comprehension* in Python and other programming languages.

   We give two examples of set builder notation. As a straightforward first example, we can use set builder notation to specify the set of all schools in the Big Ten West Division that start with an 'I':

$$\{y \in Y : y \text{ starts with an 'I'}\} = \{\text{Illinois}, \text{Iowa}\}\ .$$

   We can also use set builder notation to define the set of prime numbers by using a slightly more complicated predicate:

$$\{p \in \mathbb{Z} : p \geq 2 \wedge \forall a \in \mathbb{Z},\ (a > 1 \wedge a < p) \Rightarrow a \nmid p\} = \{2, 3, 5, 7, 11, \ldots\}\ .$$

   Here $x \mid y$, read as "$x$ divides $y$," means that $x$ is a factor of $y$, and $x \nmid y$, read as "$x$ does not divide $y$," means that $x$ is not a factor of $y$. So, the above formula uses set builder notation to define the set of prime numbers as the set of integers $p$ greater than or equal to 2 such that no integer $a$ strictly between 1 and $p$ divides $p$.

## 2.1 Important sets and set operations

We define a number of important sets in Table 3, and define a number of important operations on sets in Table 4. There are a number of containment relationships between sets in Table 3. For any $n \in \mathbb{Z}^+$, $[n] \subseteq \mathbb{N}$ and $\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R}$. There is a close correspondence between several of the basic set operations in Table 4 and logical connectives in propositional logic. Namely, union, intersection, and complement correspond to $\vee$, $\wedge$, and $\neg$, respectively.

[2] The symbol ':=' means "is defined to be," which we sometimes use instead of '=,' which can also mean logical equality. This usage is analogous to the difference between '=' and '==' in programming.

| **Name** | **Notation** | **Description** |
|---|---|---|
| Empty set | $\emptyset$ | The unique set containing no elements. |
| Counting sets | $[n] = \{1, \ldots, n\}$ | The set of integers from 1 to $n$, inclusive. |
| Natural numbers | $\mathbb{N} = \{0, 1, 2, \ldots\}$ | The counting numbers, starting at 0. |
| Integers | $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ | All whole numbers. |
| Positive integers | $\mathbb{Z}^+ = \{1, 2, 3, \ldots\}$ | The counting numbers, starting at 1. |
| Rational numbers | $\mathbb{Q} = \{0, 1, -3, -\frac{1}{2}, \frac{4}{9}, \frac{22}{7}, \ldots\}$ | The set of ratios of integers with non-zero denominator. |
| Real numbers | $\mathbb{R} = \{0, 1, -\frac{4}{9}, e, \pi, \ldots\}$ | The set of all distances between points on a line. |
| Open ranges | $(a, b) = \{x \in \mathbb{R} : a < x < b\}$ | The set of real numbers between $a$ and $b$, exclusive. |
| Closed ranges | $[a, b] = \{x \in \mathbb{R} : a \leq x \leq b\}$ | The set of real numbers between $a$ and $b$, inclusive. |

Table 3: Some important sets.

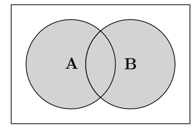
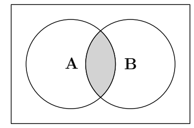
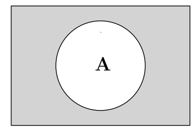
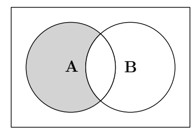
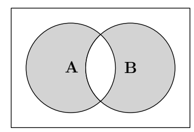
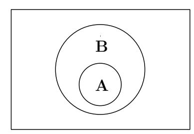
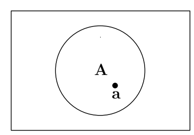
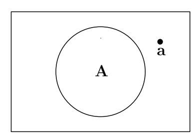
| **Name** | **Notation** | **Description** | **Picture/Example** |
|---|---|---|---|
| Union | $A \cup B$ | The set of all elements in $A$ or $B$. |   |
| Intersection | $A \cap B$ | The set of all elements in $A$ and $B$. |   |
| Complement | $\overline{A}$ or $A^c$ | The set of all elements not in $A$. |   |
| Difference | $A \setminus B$ | The set of all elements in $A$ but not in $B$. |   |
| Symmetric difference | $A \triangle B$ | The set of all elements in exactly one of $A$ and $B$. |   |
| Subset | $A \subseteq B$ | $A$ is a subset of $B$. |   |
| Containment | $a \in A$ | The element $a$ is in the set $A$. |   |
| Non-containment | $a \notin A$ | The element $a$ is not in the set $A$. |   |
| Cartesian product | $A \times B$ | The set of all ordered pairs $(a, b)$ where $a \in A$ and $b \in B$. | $\{x, y\} \times \{1, 2\}$ $= \{(x, 1), (x, 2), (y, 1), (y, 2)\}.$ |
| Power set | $\mathcal{P}(A)$ | The set of all subsets of $A$. | $\mathcal{P}(\{1, 2, 3\}) = \{\emptyset, \{1\}, \{2\}, \{3\},$ $\{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}.$ |
| Cardinality | $\|A\|$ | The number of elements in $A$. | $\|\{a, b, c\}\| = 3.$ |

Table 4: Operations on sets.

# 3 Functions

A *function* $f$ is a mapping of each element in a set $X$, called the *domain*, to a unique element in another set $Y$, called the *codomain* or *range*. In symbols, $f : X \to Y$ denotes this relationship, which we read as, "$f$ is a function that maps elements from $X$ to elements of $Y$." The familiar notation $f(x) = y$ specifies that $f$ maps the element $x \in X$ to the element $y \in Y$. It is possible that two elements in the domain map to the same element in the codomain (i.e., $f(x_1) = f(x_2)$ where $x_1 \neq x_2$), but not that a single element in the domain maps to two elements in the codomain (i.e., if $f(x) = y_1$ and $f(x) = y_2$ then it must be that $y_1 = y_2$). A function is called *Boolean-valued* or a *predicate* if its codomain is $\{0, 1\}$, is called *integer-valued* if its codomain is $\mathbb{Z}$, and is called *real-valued* if its codomain is $\mathbb{R}$.

## 3.1 Images and preimages

Given a function $f : X \to Y$ and a subset $S \subseteq X$, we define the *image* of $S$ under $f$ to be $f(S) := \{f(x) : x \in S\}$. We define the image of $f$ to be $\text{Im}(f) := f(X)$, i.e., the image of the domain $X$ of $f$ under $f$. We define the *preimage* of an element $y \in Y$ under $f$ to be $f^{-1}(y) := \{x \in X : f(x) = y\}$, and similarly define the preimage of a set $T \subseteq Y$ to be $f^{-1}(T) := \{x \in X : \exists y \in T,\ f(x) = y\}$. In other words, the image $f(S)$ of a set $S \subseteq X$ is the set of all elements in $Y$ to which $f$ maps some element in $S$, and the preimage $f^{-1}(T)$ of a set $T \subseteq Y$ is the set of all elements in $X$ that $f$ maps to elements in $T$. See Figure 1.

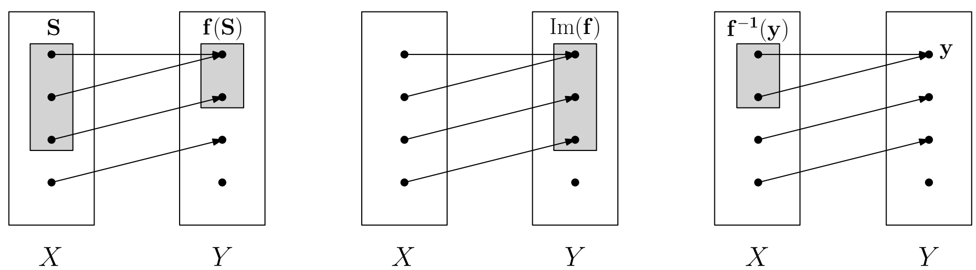



Figure 1: Three visual depictions of the same function $f$ with different subsets of the domain and codomain highlighted in gray: the image $f(S)$ of a set $S \subseteq X$ under $f$ (left), the image $\text{Im}(f)$ of $f$ (equivalent to the image $f(X)$ of the domain $X$ under $f$) (center), and the preimage of $y \in Y$ (right).

## 3.2 Injective, surjective, and bijective functions

A function $f : X \to Y$ is called *injective* or *1-to-1* if for all $y \in Y$, $\left|f^{-1}(Y)\right| \leq 1$, i.e., if for every $y \in Y$ there exists at most one element in $X$ that maps to $Y$. A function $f$ is called *surjective* or *onto* if $\mathrm{Im}(f) = Y$, i.e., if for every element $y \in Y$ there exists an element $x \in X$ such that $f(x) = y$. A function that is both injective and surjective is called *bijective*. See Figure 2.

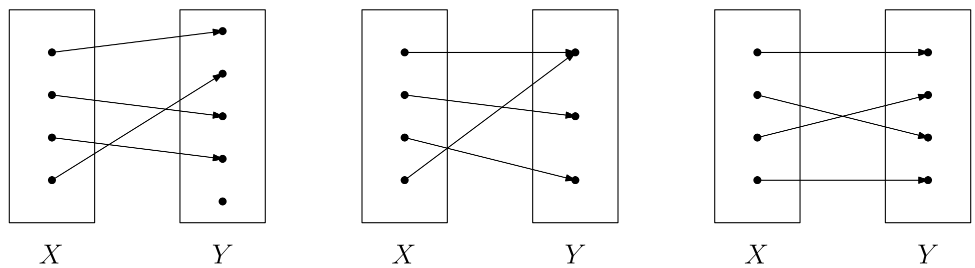



Figure 2: A visual representation of injective (left), surjective (center), and bijective (right) functions with domain $X$ and codomain $Y$ ($X$ and $Y$ differ in each case). The function on the left is injective because no two distinct elements in $X$ map to the same element in $Y$, but not surjective because no element in $X$ maps to the bottom element in $Y$. The function in the center is surjective because some element in $X$ maps to every element in $Y$, but not injective because two distinct elements map to the top element in $Y$. The function on the right is bijective because it is both injective and surjective.