1 Mathematical Expression

As a starting point for formalizing our intuition of logic, we will define two mathematical notions that we will use repeatedly throughout the course: sets and functions. Much of the terminology here may be review for you (or at least appear vaguely familiar), but please pay careful attention to the **bolded terms**, as we will make heavy use of each of them throughout the course. Each of these terms has a specific technical meaning (given by our definition) that may be subtly different from your intuitive understanding. As we will stress again and again, *definitions* are precise statements about the meaning of a term or symbol; whenever we define something, it will be your responsibility to understand that definition so that you can understand—and later, reason about—statements using these terms at any point in the rest of this course and beyond.

Sets

Definition 1.1. A **set** is a collection of distinct objects, which we call **elements** of the set. A set can have a finite number of elements, or infinitely many elements. The **size** of a finite set A is the number of elements in the set, and is denoted by $|A|$. The **empty set** (the set consisting of zero elements) is denoted by $\emptyset$.

Before moving on, let us see some concrete examples of sets. These examples illustrate not just the versatility of what sets can represent, but also illustrate various ways of *defining* sets.

Example 1.1. A finite set can be described by explicitly listing all its elements between curly brackets, such as $\{a, b, c, d\}$ or $\{2, 4, -10, 3000\}$.

Example 1.2. A set of records of all people that work for a small company. Each record contains the person's name, salary, and age. For example:

$$\big\{(\text{Ava Doe}, \$70000, 53), (\text{Donald Dunn}, \$67000, 30), (\text{Mary Smith}, \$65000, 25), (\text{John Monet}, \$70000, 40)\big\}.$$

Example 1.3. Here are some familiar *infinite* sets of numbers. Note that we use the ... to indicate the continuation of a pattern of numbers.

- The set of natural numbers, $\mathbb{N} = \{0, 1, 2, \dots\}$.[1]
- The set of integers, $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$.
- The set of positive integers, $\mathbb{Z}^+ = \{1, 2, \dots\}$.
- The set of rational numbers, $\mathbb{Q}$.

[1] By convention in computer science, 0 is a natural number.

- The set of real numbers, $\mathbb{R}$.
- The set of non-negative real numbers, $\mathbb{R}^{\geq 0}$.

Example 1.4. The set of all finite strings over $\{0, 1\}$. A *finite string over* $\{0, 1\}$ is a finite sequence $b_0 b_1 b_2 \dots b_{k-1}$, where k is a natural number (called the *length* of the string)[2] and each of b_0, b_1, etc. is either 0 or 1. The string of length 0 is called the *empty string*, and is typically denoted by the symbol ϵ.

[2] For example, the length of the string 10100101 is eight.

Note that we have defined this set without explicitly listing all of its elements, but instead by describing exactly what properties its elements have. For example, using our definition, we can say that this set contains the element 01101000, but does not contain the element 012345.[3]

[3] Food for thought: how would you generate a list of all finite strings over 0, 1?

Example 1.5. A set can also be described as in this example:

$$\{x \mid x \in \mathbb{N} \text{ and } x \geq 5\}.$$

This is the set of all natural numbers which are greater than or equal to 5. The left part (before the vertical bar $|$) describes the elements in the set in terms of a variable x, and right part states the *condition(s)* on this variable that must be satisfied.[4]

[4] Tip: The $|$ can be read as "where".

As a more complex example, we can define the set of rational numbers as:

$$\mathbb{Q} = \left\{ \frac{p}{q} \;\middle|\; p, q \in \mathbb{Z} \text{ and } q \neq 0 \right\}.$$

We have only scratched the surface of the kinds of objects we can represent using sets. Later on in the course, we will enrich our set of examples by studying sets of computer programs, sequences of numbers, and graphs.

Operations on sets

We have already seen one set operation: the size operator, $|A|$. In this subsection, we'll list other common set operations that we will use in this course.

The following *boolean* set operations return either True or False. We only describe when these operations return True; they return False in all other cases.

- $x \in A$: returns True when x is an element of A; $y \notin A$ returns True when y is *not* an element of A.
- $A \subseteq B$: returns True when every element of A is also in B. We say in this case that A is a **subset** of B.

  Every set is a subset of itself, and the empty set is a subset of every set: $A \subseteq A$ and $\emptyset \subseteq A$ are always True.
- $A = B$: returns True when $A \subseteq B$ and $B \subseteq A$. In this case, A and B contain the exact same elements.

The following operations return sets:

- $A \cup B$, the **union** of A and B. Returns the set consisting of all elements that occur in A, in B, or in both.

  $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$.
- $A \cap B$, the **intersection** of A and B. Returns the set consisting of all elements that occur in both A and B.

  $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$.
- $A \setminus B$, the **difference** of A and B. Returns the set consisting of all elements that are in A but that are not in B.

  $A \setminus B = \{x \mid x \in A \text{ and } x \notin B\}$.
- $A \times B$, the **(Cartesian) product** of A and B. Returns the set consisting of all *pairs* (a, b) where a is an element of A and b is an element of B.

  $A \times B = \{(x, y) \mid x \in A \text{ and } y \in B\}$.
- $\mathcal{P}(A)$, the **power set** of A, returns the set consisting of all subsets of A.[5] For example, if $A = \{1, 2, 3\}$, then

$$\mathcal{P}(A) = \big\{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\big\}.$$

  $\mathcal{P}(A) = \{S \mid S \subseteq A\}$.

[5] Food for thought: what is the relationship between $|A|$ and $|\mathcal{P}(A)|$?

Functions

Definition 1.2. Let A and B be sets. A **function** $f : A \to B$ is a mapping from elements in A to elements in B. A is called the **domain** of the function, and B is called the **codomain** of the function.

For example, if A and B are both the set of integers, then the (predecessor) function $Pred : \mathbb{Z} \to \mathbb{Z}$, defined by $Pred(x) = x - 1$, is the function that maps each integer x to the integer before it. Given this definition, we know that $Pred(10) = 9$ and $Pred(-3) = -4$.

A more formal definition of the term "mapping" above is a subset of the Cartesian product $A \times B$, where every element of A appears exactly once. For example, we can define the *Pred* function as the following set:

$$\{\dots, (-2, -3), (-1, -2), (0, -1), (1, 0), (2, 1), \dots\}.$$

One important distinction between the domain and codomain of a function is in what they require of that function. For a function $f : A \to B$, its domain A is the set of possible inputs for the function, and f *must* have a valid value for every single one of those inputs. So for example, the function $g(x) = \frac{1}{x}$ cannot have domain $\mathbb{R}$, since $g(0)$ is not defined.[6] However, the codomain B only has to *contain* the possible outputs of f—not every element of B needs to be a possible output. Continuing our example, the function $g(x) = \frac{1}{x}$ can have codomain $\mathbb{R}$, since $\frac{1}{x}$ is always a real number, even though $g(x)$ never outputs 0.

[6] We could choose $\mathbb{R} \setminus \{0\}$ as g's domain.

Sometimes it is useful to discuss the exact of possible outputs of a function. For this, we have one more definition.

Definition 1.3. Let $f : A \to B$ be a function. We define the **range** of f to be the set consisting of its possible outputs. Formally, this is the set $\{f(x) \mid x \in A\}$.

Note that the range of f is always a subset of its codomain B, but does not necessarily equal B.

You might wonder: why bother having separate definitions for codomain and range, why not just always define functions with their exact range? There are two reasons why this isn't always feasible:

- Functions don't always have a range that is easy to describe or compute. For example, the function $f(x) = (1 + \sin(x))^{\cos(x)}$ over the domain $\mathbb{R}$ always outputs a non-negative real number, so we can pick its codomain to be $\mathbb{R}^{\geq 0}$, but finding its precise range requires more work.
- Later on, we'll be analysing properties of *arbitrary* functions with a given domain and codomain, for example, an arbitrary function $f : \mathbb{R} \to \mathbb{R}$. In these cases, we'll want to include functions whose range is potentially much smaller than $\mathbb{R}$ in our analysis.

For these reasons, we'll generally define function codomains using standard numeric sets like $\mathbb{N}$ and $\mathbb{R}$, and leave the range of a function unstated unless it is required by the particular problem at hand.

Function arity

Functions can have more than one input. For sets $A_1, A_2, \dots, A_k$ and B, a k**-ary function** $f : A_1 \times A_2 \times \cdots \times A_k \to B$ is a function that takes k arguments, where for each i between 1 and k, the i-th argument of f must be an element of A_i, and where f returns an element of B. We have common English terms for small values of k: *unary*, *binary*, and *ternary* functions take one, two, and three inputs, respectively. For example, the addition operator $+ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a binary function that takes two real numbers and returns their sum. For readability, we usually write this function as $x + y$ instead of $+(x, y)$.

Predicates

A **predicate** is a function whose codomain is {True, False}.[7] For example, we can define the predicate $Odd : \mathbb{N} \to \{\text{True}, \text{False}\}$ by mapping all even numbers to False, and all odd numbers to True. Given a predicate P and element x of its domain, we say that x **satisfies** P when $P(x)$ is True.

[7] In other courses, you may see True and False represented as the numbers 1 and 0, respectively.

Predicates and sets have a natural equivalence that we will sometimes make use of in this course. Given a predicate $P : A \to \{\text{True}, \text{False}\}$, we can define the set $\{x \mid x \in A \text{ and } P(x) = \text{True}\}$, i.e., the set of elements of A which satisfy P. On the flip side, given a subset $S \subseteq A$, we can define the predicate $P : A \to \{\text{True}, \text{False}\}$ by $P(x) = \text{True}$ if $x \in S$, and $P(x) = \text{False}$ if $x \notin S$. For example, consider the predicate $Even : \mathbb{N} \to \{\text{True}, \text{False}\}$ that is True exactly when its