Chapter 3

Set Theory

- Set theory is the dominant foundation for mathematics.
- The idea is that everything else in mathematics—numbers, functions, etc.—can be written in terms of sets, so that if you have a consistent description of how sets behave, then you have a consistent description of how everything built on top of them behaves.
- If predicate logic is the machine code of mathematics, set theory would be assembly language.
-

- The nice thing about set theory is that it requires only one additional predicate on top of the standard machinery of predicate logic.

- This is the **membership** or **element** predicate $\in$, where $x \in S$ means that x is an element of S. Here S is a set—a collection of elements—and the identify of S is completely determined by which x satisfy $x \in S$.

- Every other predicate in set theory can be defined in terms of $\in$.

-

Naive set theory

- **Naive set theory** is the informal version of set theory that corresponds to our intuitions about sets as unordered collections of objects (called **elements**) with no duplicates.

- An element of a set may also be a set (in which case it contains its own elements), or it may just be some object that is not a set (also known as an **urelement**, which is German for “primitive element”)

-

- {} = the **empty set** ∅, which has no elements.
- {Moe, Curly, Larry} = the **Three Stooges**.
- {0, 1, 2, . . .} = N, the **natural numbers**. Note that we are relying on the reader guessing correctly how to continue the sequence here.
- {{},{0},{1},{0,1},{0,1,2},7} = a set of sets of natural numbers, plus a stray natural number that is directly an element of the outer set.
-

Membership

- Membership in a set is written using the ∈ symbol (pronounced "is an element of," "is a member of," or just "is in"). So we can write Moe ∈ the Three Stooges or 4 ∈ N. We can also write ∉ for "is not an element of," as in Moe ∉ N, and the reversed symbol ∋ for "has as an element," as in N ∋ 4.

- A fundamental axiom in set theory (the **Axiom of Extensionality**; see §3.4) is that the only distinguishing property of a set is its list of members: if two sets have the same members, they are the same set.

-

- For nested sets like $\{\{1\}\}$, $\in$ represents only direct membership: the set $\{\{1\}\}$ only has one element, $\{1\}$, so $1 \notin \{\{1\}\}$.

- This can be confusing if you think of $\in$ as representing the English "is in," because if I put my lunch in my lunchbox and put my lunchbox in my backpack, then my lunch is in my backpack. But my lunch is *not* an element of {{my lunch} , my textbook, my slingshot}.

- In general, $\in$ is not transitive
  (see §9.3): it doesn't behave like $\leq$ unless there is something very unusual about the set you are applying it to.

-

set comprehension

- A rule for how to generate all of its elements
- set-builder notation
-
-

- $\{x|x\in N\wedge x>1\wedge(\forall y\in N:\forall z\in N:yz=x\rightarrow y=1\vee z=1)\}$=the prime numbers.
- $\{2x|x\in N\}$=the even numbers.
- $\{x|x\in N\wedge x<12\}=\{0,1,2,3,4,5,6,7,8,9,10,11\}$.
-

$$\{ \, x \mid 0 \leq x \leq 100, \, x = 1 \pmod 2 \, \}$$

begindisplaymath 2ex] `[ x | x <- [0..100], x ‘mod‘ 2 == 1 ]`

begindisplaymath 2ex] `[ x for x in range(0,101) if x % 2 == 1 ]`

Table 3.1: Set comprehension vs list comprehension. The first line gives the set of odd numbers between 0 and 100 written using set-builder notation. The other lines construct the odd numbers between 0 and 100 as ordered list data structures in Haskell and Python respectively.

- Some very high-level programming languages like Haskell or Python have a similar mechanism called **list comprehension** which does pretty much the same thing except the result is an ordered list.

- $\{n \in N | \exists x,y,z \in N \setminus \{0\} : x^n + y^n = z^n\}$. This is a fancy name for {1, 2}, but this fact is not obvious [Wil95].

.

Operations on sets

- $A \cup B = \{x \mid x \in A \vee x \in B\}$. The **union** of A and B.
- $A \cap B = \{x \mid x \in A \wedge x \in B\}$. The **intersection** of A and B.
- $A \setminus B = \{x \mid x \in A \wedge x \notin B\}$. The **set difference** of A and B.
- $A \triangle B = \{x \mid x \in A \oplus x \in B\}$. The **symmetric difference** of A and B.

Corresponding to implication is the notion of a **subset**:

- $A \subseteq B$ ("A is a subset of B") if and only if $\forall x: x \in A \rightarrow x \in B$.

- $A \supseteq B$ means that A is a **superset** of B, which is the same as saying $B \subseteq A$.

- We can also write $A \nsubseteq B$ to say that A is a not a subset of B, and the rather awkward-looking $A \subsetneq B$ to say that A is a **proper subset** of B, meaning that $A \subseteq B$ but $A \neq B$. (The standard version $A \subseteq B$ allows the case $A = B$.)

-

-

- Usually we will try to reserve “is in” for $\in$ and “is contained in” for $\subseteq$, but it’s safest to use the symbols (or “is an element/subset of”) to avoid any possibility of ambiguity.
-

- Sometimes one says A is **contained in** B if $A \subseteq B$.

- This is one of two senses in which A can be “in” B—it is also possible that A is in fact an element of B ($A \in B$).

- For example, the set A = {12} is an element of the set B = {Moe, Larry, Curly, {12}}, but A is not a subset of B, because A’s element 12 is not an element of B.

-

complement

- $\overline{A}$ ={x| x$\notin$A}. The set $\overline{A}$ is known as the **complement** of A.
-

- If we allow complements, we are necessarily working inside some fixed **universe**, since the complement $U = \bar{\emptyset}$ of the empty set contains all possible objects

•

- The set theory used in most of mathematics is defined by a collection of axioms that allow us to construct, essentially from scratch, a universe big enough to hold all of mathematics without apparent contradictions while avoiding the paradoxes that may arise in naive set theory.

- However, one consequence of this construction is that the universe (a) much bigger than anything we might ever use, and (b) *not* a set, making complements not very useful. The usual solution to this is to replace complements with explicit set differences: U \ A for some specific universe U instead of

- $\bar{A}$

Proving things about sets

- Given x and S, show $x \in S$. This requires looking at the definition of S to see if x satisfies its requirements, and the exact structure of the proof will depend on what the definition of S is.
-

- Given S and T , show S ⊆ T . Expanding the definition of subset, this means we have to show that every x in S is also in T. So a typical proof will pick an arbitrary x in S and show that it must also be an element of T . This will involve unpacking the definition of S and using its properties to show that x satisfies the definition of T.

-

- Given S and T, show $S = T$. Typically we do this by showing $S \subseteq T$ and $T \subseteq S$ separately. The first shows that $\forall x : x \in S \rightarrow x \in T$; the second shows that $\forall x : x \in T \rightarrow x \in S$. Together, $x \in S \rightarrow x \in T$ and $x \in T \rightarrow x \in S$ gives $x \in S \leftrightarrow x \in T$, which is what we need for equality.

- .

corresponding negative statements

- For $x \notin S$, use the definition of S as before.

- For $S \not\subseteq T$, we only need a counterexample: pick any one element of S and show that it's not an element of T .

-

Lemma 3.3.1. *The following statements hold for all sets S and T, and all predicates P:*

$$S \supseteq S \cap T \tag{3.3.1}$$

$$S \subseteq S \cup T \tag{3.3.2}$$

$$S \supseteq \{x \in S \mid P(x)\} \tag{3.3.3}$$

$$S = (S \cap T) \cup (S \setminus T) \tag{3.3.4}$$

Proof.

- (3.3.1) Let x be in $S \cap T$. Then $x \in S$ and $x \in T$, from the definition of $S \cap T$. It follows that $x \in S$. Since x was arbitrary, we have that for all x in $S \cap T$, x is also in T; in other words, $S \cap T \subseteq T$.

- (3.3.2). Let x be in S. Then $x \in S \vee x \in T$ is true, giving $x \in S \cup T$.

- (3.3.3) Let x be in $\{x \in S \mid P(x)\}$. Then, by the definition of set comprehension, $x \in S$ and $P(x)$. We don't care about $P(x)$, so we drop it to just get $x \in S$.

- (3.3.4). This is a little messy, but we can solve it by breaking it down into smaller problems.

  First, we show that $S \subseteq (S \setminus T) \cup (S \cap T)$. Let x be an element of S. There are two cases:

  1. If $x \in T$, then $x \in (S \cap T)$.
  2. If $x \notin T$, then $x \in (S \setminus T)$.

  In either case, we have shown that x is in $(S \cap T) \cup (S \setminus T)$. This gives $S \subseteq (S \cap T) \cup (S \setminus T)$.

  Conversely, we show that $(S \setminus T) \cup (S \cap T) \subseteq S$. Suppose that $x \in (S \setminus T) \cup (S \cap T)$. Again we have two cases:

  1. If $x \in (S \setminus T)$, then $x \in S$ and $x \notin T$.
  2. If $x \in (S \cap T)$, then $x \in S$ and $x \in T$.

  In either case, $x \in S$.

  Since we've shown that both the left-hand and right-hand sides of (3.3.4) are subsets of each other, they must be equal.

Axiomatic set theory

- The axioms most commonly used are known as **Zermelo-Fraenkel set theory with choice** or **ZFC**.

- The short version is that you can construct sets by (a) listing their members, (b) taking the union of other sets, (c) taking the set of all subsets of a set, or (d) using some predicate to pick out elements or subsets of some set.

•

These properties follow from the more useful axioms of ZFC:

- **Extensionality** Any two sets with the same elements are equal.[2]
- **Existence** The empty set $\varnothing$ is a set.[3]
- **Pairing** Given sets x and y, {x, y} is a set.[4]
- **Union** For any set of sets S={x,y,z,…}, the set $\bigcup S = x \cup y \cup z \cup \ldots$ exists.[5]
- **Power set** For any set S, the power set $P(S) = \{A \mid A \subseteq S\}$ exists.[6]
-

Specification For any set S and any predicate P, the set $\{x \in S \mid P(x)\}$ exists.[7] This is called **restricted comprehension**, and is an **axiom schema** instead of an axiom, since it generates an infinite list of axioms, one for each possible P. Limiting ourselves to constructing subsets of existing sets avoids Russell's Paradox, because we can't construct $S = \{x \mid x \notin x\}$. Instead, we can try to construct $S = \{x \in T \mid x \notin x\}$, but we'll find that S isn't an element of T, so it doesn't contain itself but also doesn't create a contradiction.

- 吉安鄉男士理髮師有規定，只能幫沒有理自己頭髮的男士理髮
- 有一位吉安鄉男士理髮師說：所有吉安鄉沒有幫自己理髮的男士的頭髮都是他理的
- 這位理髮師有沒有理他自己頭髮？

Russell paradox

- https://www.scientificamerican.com/article/what-is-russells-paradox/
- A confusing terminology, “not in”
- *x* = *{a: a* is not in *a}* leads to a contradiction in the same way as the description of the collection of barbers.
  Is *x* itself in the set *x*? Either answer leads to a contradiction.

Infinity There is a set that has $\emptyset$ as a member and also has $x \cup \{x\}$ whenever it has x.[8] This gives an encoding of $\mathbb{N}$ where $\emptyset$ represents 0 and $x \cup \{x\}$ represents $x + 1$. Expanding out the $x + 1$ rule shows that each number is represented by the set of all smaller numbers, e.g. $3 = \{0, 1, 2\} = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$, which has the nice property that each number n is represented by a set with exactly n elements, and that $a < b$ can be represented by $a \in b$.[9]

Without this axiom, we only get finite sets.

Cartesian products, relations, and functions

- Sets are unordered: the set {a, b} is the same as the set {b, a}. Sometimes it is useful to consider **ordered pairs** (a, b), where we can tell which element comes first and which comes second. These can be encoded as sets using the rule (a, b) = {{a} , {a, b}}

.

- Given sets A and B, their **Cartesian product** $A \times B$ is the set $\{(x, y) \mid x \in A \wedge y \in B\}$, or in other words the set of all ordered pairs that can be constructed
by taking the first element from A and the second from B. If A has n
elements and B has m, then $A \times B$ has nm elements.[14] For example,
$\{1, 2\} \times \{3, 4\} = \{(1, 3), (1, 4), (2, 3), (2, 4)\}$.

- $A \times B \neq B \times A$

-

- The existence of the Cartesian product of any two sets can be proved using the axioms we already have: if (x, y) is defined as {{x} , {x, y}}, then $\mathcal{P}$(A ∪ B) contains all the necessary sets {x} and {x, y} , and $\mathcal{P}$($\mathcal{P}$(A ∪ B)) contains all the pairs {{x} , {x, y}}.

-

functions

- A special class of relations are **functions.** A function from a **domain** A to a **codomain**[15] B is a relation on A and B (i.e., a subset of $A \times B$ such that every element of A appears on the left-hand side of exactly one ordered pair.

- We write $f{:}A \to B$ as a short way of saying that f is a function from A to B, and for each $x \in A$ write $f(x)$ for the unique $y \in B$ with $(x,y) \in f$.

- The set of *all* functions from A to B is written as B^A: note that the order of A and B is backwards here from A → B. Since this is just the subset of $\mathcal{P}(A \times B)$, consisting of functions as opposed to more general relations, it exists by the Power Set and Specification axioms.

幂集

-

- Often, a function is specified not by writing out some huge set of ordered pairs, but by giving a rule for computing f(x). An example: $f(x) = x^2$. Particular trivial functions can be defined in this way anonymously; another way to write $f(x) = x^2$ is as the anonymous function $x \to x^2$.

-

- $f(x) = x^2$. Note: this single rule gives several different functions, e.g. $f : R \to R$, $f : Z \to Z$, $f : N \to N$, $f : Z \to N$. Changing the domain or codomain changes the function.
- $f(x)=x+1$.
- Floor and ceiling functions: when x is a real number, the **floor** of x (usually written $\lfloor x \rfloor$) is the largest integer less than or equal to x and the **ceiling** of x (usually written $\lceil x \rceil$) is the smallest integer greater

than or equal to x. E.g., $\lfloor 2 \rfloor = \lceil 2 \rceil = 2$, $\lfloor 2.337 \rfloor = 2$, $\lceil 2.337 \rceil$

-

- The function from {0, 1, 2, 3, 4} to {a, b, c} given by the following table:

  0 a

  1 c

  2 b

  3 a

  4 b

-

Sequences

- Functions let us define sequences of arbitrary length: for example, the infinite sequence $x_0, x_1, x_2, \ldots$ of elements of some set A is represented by a function $x : N \to A$.

- A shorter sequence (a_0, a_1, a_2) would be represented by a function $a : \{0, 1, 2\} \to A$.

-

- The **subscript** takes the place of a function argument: we treat x_n as **syntactic sugar** for $x(n)$.

- Finite sequences are often called **tuples.**

- We think of the result of taking the Cartesian product of a finite number of sets $A \times B \times C$ as a set of tuples (a, b, c), even though the actual structure may be $((a, b), c)$ or $(a, (b, c))$ depending on which product operation we do first.

-

- We can think of the Cartesian product of k sets (where k need not be 2) as a set of sequences indexed by the set {1 . . . k} (or sometimes {0 . . . k – 1}).
- $A \times B \times C$, the set of functions from {1, 2, 3} to $A \cup B \cup C$ with the property that for each function $f \in A \times B \times C$, $f(1) \in A$, $f(2) \in B$, and $f(3) \in C$)
- Technically this means that $A \times B \times C$ is not the same as $(A \times B) \times C$ or $A \times (B \times C)$.
  - $(A \times B) \times C$, the set of all ordered pairs whose first element is an ordered pair in $A \times B$ and whose second element is in C
  - $A \times (B \times C)$, the set of ordered pairs whose first element is in A and whose second element is in $B \times C$.

.

Cartesian products over indexed collections of sets can be written using product notation (see §6.2), as in

$$\prod_{i=1}^{n} A_n$$

Functions of more (or less) than one argument

- If $f : A \times B \to C$, then we write $f(a,b)$ for $f((a,b))$. In general we can have a function with any number of arguments (including 0); a function of k arguments is just a function from a domain of the form $A_1 \times A_2 \times \ldots A_k$ to some codomain B.

.

Composition of functions

- Two functions $f : A \to B$ and $g : B \to C$ can be **composed** to give a **composition** $g \circ f$.

- This is a function from A to C defined by $(g \circ f)(x) = g(f(x))$. Composition is often implicit in definitions of functions: the function $x \to x^2 + 1$ is the composition of two functions $x \to x + 1$ and $x \to x^2$.

-

Functions with special properties

- We can classify functions f : A → B based on how many elements x of the domain A get mapped to each element y of the codomain B.

- If every y is the image of at least one x, f is **surjective**.

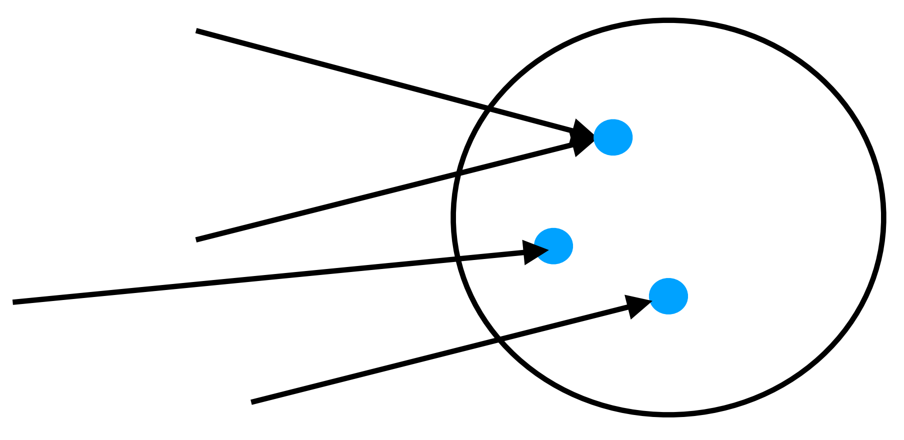
- If every y is the image of at most one x, f is **injective**.

- If every y is the image of exactly one x, f is **bijective**.

-

Surjections

- A function f : A → B that covers every element of B is called **onto**, **surjective**, or a **surjection.** This means that for any y in B, there exists some x in A such that y = f(x). An equivalent way to show that a function is surjective is to show that its **range** $\{f(x) \mid x \in A\}$ is equal to its codomain.
-



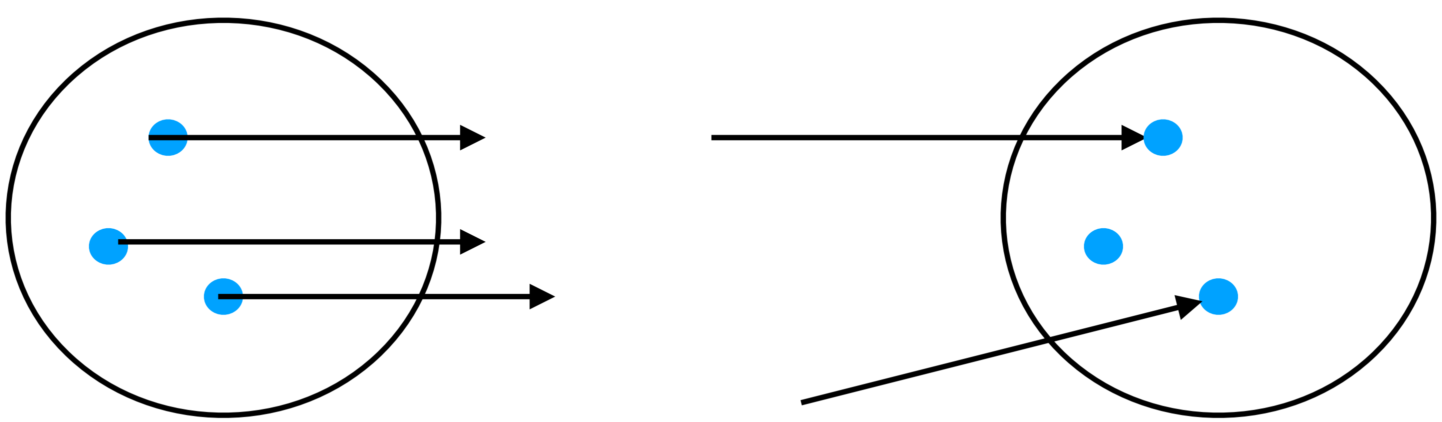
- B

- For example, the function $f(x) = x^2$ from N to N is not surjective, because its range includes only perfect squares. The function $f(x) = x + 1$ from N to N is not surjective because its range doesn't include 0. However, the function $f(x) = x + 1$ from Z to Z *is* surjective, because for every y in Z there is some x in Z such that $y = x + 1$.

Injections

- If $f : A \rightarrow B$ maps distinct elements of A to distinct elements of B (i.e., if $x \neq y$ implies $f(x) \neq f(y)$), it is called **one-to-one**, **injective**, or an **injection**.

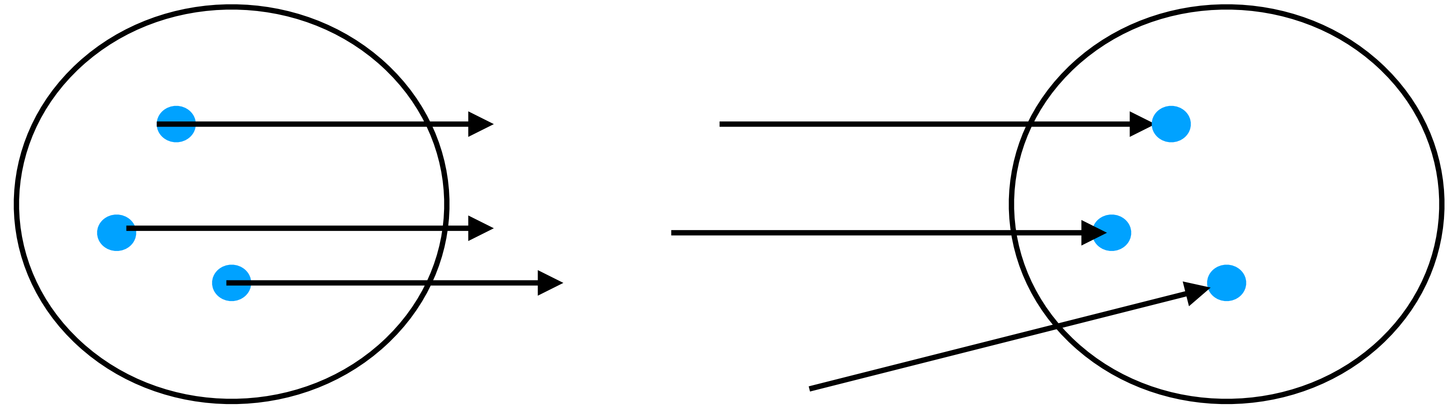
- By contraposition, an equivalent definition is that $f(x) = f(y)$ implies $x = y$ for all x and y in the domain. For example, the function $f(x) = x^2$ from N to N is injective. The function $f(x) = x^2$ from Z to Z is *not* injective (for example, $f(-1) = f(1) = 1$). The function $f(x) = x + 1$ from N to N is injective.



- A
- B

Bijections

- A function that is both surjective and injective is called a **one-to-one correspondence**, **bijective**, or a **bijection.** Any bijection f has an **inverse** function f^{-1}; this is the function {(y,x) | (x,y) ∈ f}. Of the functions we have been using as examples, only f (x) = x + 1 from Z to Z is bijective.



- A
- B

Bijections and counting

- Bijections let us define the size of arbitrary sets without having some special means to count elements. We say two sets A and B have the same **size** or **cardinality** if there exists a bijection $f : A \leftrightarrow B$.
-

- Often it is convenient to have standard representatives of sets of a given cardinality. A common trick is to use the **von Neumann ordinals**, which are sets that are constructed recursively so that each contains all the smaller ordinals as elements.

-

- The empty set ∅ represents 0, the set {0} represents 1, {0, 1} represents 2, and so on. The first infinite ordinal is $\omega = \{0, 1, 2, . . .\}$, which is followed by $\omega + 1 = \{0, 1, 2, . . . ; \omega\}$, $\omega + 2 = \{0, 1, 2, . . . ; \omega, \omega + 1\}$, and so forth; there are also much bigger ordinals like ω^2 (which looks like ω many copies of ω stuck together), ω^{ω} (which is harder to describe, but can be visualized as the set of infinite sequences of natural numbers with an appropriate ordering), and so on.

- Given any collection of ordinals, it has a smallest element, equal to the intersection of all elements: this means that von Neumann ordinals are **well-ordered** (see §9.5.6). So we can define the cardinality $|A|$ of a set A formally as the unique smallest ordinal B such that there exists a bijection $f : A \leftrightarrow B$.

- http://planetmath.org/vonneumannordinal

- https://www.quora.com/How-will-you-define-numbers-in-a-formal-way

.

1

Integers The integers are the set $\mathbb{Z} = \{\ldots, -2, -1, 0, -1, 2, \ldots\}$. We represent each integer z as an ordered pair (x, y), where $x = 0 \vee y = 0$; formally, $\mathbb{Z} = \{(x, y) \in \mathbb{N} \times \mathbb{N} \mid x = 0 \vee y = 0\}$. The interpretation of (x, y) is $x - y$; so positive integers z are represented as $(z, 0)$ while negative integers are represented as $(0, -z)$. It's not hard to define addition, subtraction, multiplication, etc. using this representation.

z

Deterministic finite state machines A **deterministic finite state machine** is a tuple $(\Sigma, Q, q_0, \delta, Q_{\text{accept}})$ where Σ is an **alphabet** (some finite set), Q is a **state space** (another finite set), $q_0 \in Q$ is an **initial state**, $\delta : Q \times \Sigma \to Q$ is a **transition function** specifying which state to move to when processing some symbol in Σ, and $Q_{\text{accept}} \subseteq Q$ is the set of **accepting states**. If we represent symbols and states as natural numbers, the set of all deterministic finite state machines is then just a subset of $\mathcal{P}(\mathbb{N}) \times \mathcal{P}(\mathbb{N}) \times \mathbb{N} \times \left(\mathbb{N}^{\mathbb{N}\times\mathbb{N}}\right) \times \mathcal{P}(\mathbb{N})$ satisfying some consistency constraints.

$A \cup B$

- $\aleph_0 + \aleph_0 = \aleph_0$. In other words, it is possible to have two sets A and B that both have the same size as $\mathbb{N}$, take their disjoint union, and get another set $A + B$ that has the same size as $\mathbb{N}$. To give a specific example, let $A = \{2x \mid x \in \mathbb{N}\}$ (the **even numbers**) and $B = \{2x + 1 \mid x \in \mathbb{N}\}$ (the **odd numbers**). These have $|A| = |B| = |\mathbb{N}|$ because there is a bijection between each of them and $\mathbb{N}$ built directly into their definitions. It's also not hard to see that A and B are disjoint, and that $A \cup B = \mathbb{N}$. So $|A| = |B| = |A| + |B|$ in this case.

$|A \cup B|$

$\aleph_0 \cdot \aleph_0 = \aleph_0$. Example: A bijection between $\mathbb{N} \times \mathbb{N}$ and $\mathbb{N}$ using the **Cantor pairing function** $\langle x, y \rangle = (x+y+1)(x+y)/2+y$. The first few values of this are $\langle 0, 0 \rangle = 0$, $\langle 1, 0 \rangle = 2 \cdot 1/2 + 0 = 1$, $\langle 0, 1 \rangle = 2 \cdot 1/2 + 1 = 2$, $\langle 2, 0 \rangle = 3 \cdot 2/2 + 0 = 3$, $\langle 1, 1 \rangle = 3 \cdot 2/2 + 1 = 4$, $\langle 0, 2 \rangle = 3 \cdot 2/2 + 2 = 5$, etc. The basic idea is to order all the pairs by increasing $x+y$, and then order pairs with the same value of $x+y$ by increasing y. Eventually every pair is reached.

$\mathbb{N}^* = \{\text{all finite sequences of elements of } \mathbb{N}\}$ has size $\aleph_0$. One way to do this to define a function recursively by setting $f([]) = 0$ and $f([\text{first}, \text{rest}]) = 1 + \langle \text{first}, f(\text{rest}) \rangle$, where first is the first element of

the sequence and rest is all the other elements. For example,

$$\begin{aligned}
f(0,1,2) &= 1 + \langle 0, f(1,2) \rangle \\
&= 1 + \langle 0, 1 + \langle 1, f(2) \rangle \rangle \\
&= 1 + \langle 0, 1 + \langle 1, 1 + \langle 2, 0 \rangle \rangle \rangle \\
&= 1 + \langle 0, 1 + \langle 1, 1 + 3 \rangle \rangle = 1 + \langle 0, 1 + \langle 1, 4 \rangle \rangle \\
&= 1 + \langle 0, 1 + 19 \rangle \\
&= 1 + \langle 0, 20 \rangle \\
&= 1 + 230 \\
&= 231.
\end{aligned}$$

Countable sets

The sets $\mathbb{N}$, $\mathbb{N}^2$, and $\mathbb{N}^*$ all have the property of being **countable**, which means that they can be put into a bijection with $\mathbb{N}$ or one of its subsets. Countability of $\mathbb{N}^*$ means that anything you can write down using finitely many symbols (even if they are drawn from an infinite but countable alphabet) is countable. This has a lot of applications in computer science: one of them is that the set of all computer programs in any particular programming language is countable.