Discrete Mathematics

Frank (Peng) Fu

1 Basic Set Theory

Why set theory?

- Set theory provides a common language to describe collections of things in both mathematics and computer science. For example, we often talk about the set of natural numbers, a set of names/strings, etc.
- The knowledge of set theory can also be very useful in practice. In all popular programming languages, set is implemented as a kind of data structure, and the language provides libraries to manipulate sets.

1.1 Basic concepts and notations

First let us consider the following example of set using braces.

- Infinite set: natural numbers[1] ($\mathbb{N} = \{0, 1, 2, 3, ...\}$), integers ($\mathbb{Z} = \{..., -3, -2, -1, 0, 1, 2, 3, ...\}$).
- Finite set: $\{0, 1, 2, 3\}$, $\{\text{'}a\text{'}, \text{'}b\text{'}, ..., \text{'}z\text{'}\}$, $\{0, 1\}$, $\{\}$, $\{1\}$.

Remarks.

- The emptyset $\{\}$ is also written as $\emptyset$.
- The braces notation can be nested. For example, $\{\{0, 1, 2\}, \{3, 4\}\}$ is a set contains two elements and each element itself is a set.
- Although very rarely used in practice, a set can be *heterogeneous*. For example, $\{\text{'}a\text{'}, \text{“}hello\text{”}, 1, \emptyset, \mathbb{N}\}$ is a set contains five elements.
- A set with one element is also called *singleton* set.

Given a set A, one natural thing to ask is if a given thing e is in the set. We write $e \in A$ if e is in the set A, otherwise we write $e \notin A$. For example, $1 \in \mathbb{N}, \text{'}a\text{'} \notin \mathbb{N}, \text{'}a\text{'} \in \{\text{'}a\text{'}, \text{'}b\text{'}, ..., \text{'}z\text{'}\}$.

Given two sets A, B, we can ask if they are equal. Two sets are equal if they have the same elements. Questions, are the following equals?

- $\{0, 1, 2\} \stackrel{?}{=} \{2, 1, 0\}$
- $\{0, 1, 2, 2\} \stackrel{?}{=} \{2, 1, 0\}$
- $\{0, 1, 2\} \stackrel{?}{=} \{2 + 1, 1 + 2, 0\}$

[1] Note that 0 is a natural number.

- $\{0,1,2\} \stackrel{?}{=} \{3,2,0\}$

A more rigor definition of set equality is the following.

Definition 1. *Let A, B be sets. $A = B$ if for every $e \in A$, we have $e \in B$; and for every $e \in B$, we have $e \in A$.*

Given two sets, we can also ask if one set contains the other.

Definition 2. *Let A, B be sets. We write $A \subseteq B$ if all the elements of A are in B.*

- $\{0,1\} \stackrel{?}{\subseteq} \{0,1,2\}$
- $\{0,1,2\} \stackrel{?}{\subseteq} \{0,1\}$
- $\{0,1,2\} \stackrel{?}{\subseteq} \{0,1,2\}$
- $\{\} \stackrel{?}{\subseteq} \{0,1,2\}$
- $\{0,1,2\} \stackrel{?}{\subseteq} \{0,1,3\}$
- $\{0,1,3\} \stackrel{?}{\subseteq} \{0,1,2\}$

Note that we write $A \subset B$ if $A \subseteq B$ and $A \neq B$. In another word, B contains A, and it contains more things than A.

Theorem 1. *Let A be any set, we always have $\emptyset \subseteq A$.*

Theorem 2. *Let A, B be set, $A = B$ if and only if $A \subseteq B$ and $B \subseteq A$.*

1.2 Set comprehension, basic operations and cartesian product

Set comprehension notation [2] Informally, we often use ellipses to describe infinite set. For example, $\{0,1,2,3,\ldots\}$, $\{0,2,4,6,8,\ldots\}$. But sometimes this can be ambiguous. For example, what does the set $A = \{2,3,\ldots\}$ mean? So it can means at least two things, i.e., a set of numbers larger than one, or the set of prime numbers. So the better practice is to use set comprehension notation instead. For example, if we want the set A to mean primes, then we just need to write $\{x \mid x \in \mathbb{N}, x \text{ is a prime}\}$. If we want A to mean a set of numbers larger than one, we just write $\{x \mid x \in \mathbb{N}, x > 1\}$.

The set comprehension notation in general has the form $S = \{x \mid x \in A, \text{statements about } x\}$, where A is a set. Note that there can be many statements about x in the set comprehension notation. To check whether $e \in S$, we just need to check: $e \in A$ and e satisfies all the statements. Sometimes the requirement of $x \in A$ can be dropped when it is clear.

The following are some more examples.

- Even numbers: $\{x \mid x \in \mathbb{N}, x = 2k \text{ for some } k \in \mathbb{N}\}$
- Odd numbers: $\{x \mid x \in \mathbb{N}, x = 2k+1 \text{ for some } k \in \mathbb{N}\}$

With set comprehension notation, we can define the following operations on sets.

[2] It is also called *set builder* notation.

- $A \cup B \stackrel{\text{def}}{=} \{x \mid x \in A \text{ or } x \in B\}$.
  e.g. $\{0,1,2\} \cup \{1,2,3\} =?$
- $A \cap B \stackrel{\text{def}}{=} \{x \mid x \in A \text{ and } x \in B\}$.
  e.g. $\{0,1,2\} \cap \{1,2,3\} =?$
- $A/B \stackrel{\text{def}}{=} \{x \mid x \in A \text{ and } x \notin B\}$.
  e.g. $\{0,1,2\}/\{1,2,3\} =?$
- $\text{Pow}(A) \stackrel{\text{def}}{=} \{B \mid B \subseteq A\}$.
  e.g. $\text{Pow}(\{0,1,2\}) =?$

Cartesian products. Let $a, b \in A$. We write (a, b) to mean a *pair* with left component a, and right component b. We can compare pairs, $(a, b) = (c, d)$ if $a = c$ and $b = d$. Note that the order of the pair matter, in general, $(a, b) \neq (b, a)$ (unless $a = b$ of course).

Let A, B be sets, we define $A \times B = \{(a, b) \mid a \in A, b \in B\}$, and we say $A \times B$ is the Cartesian product of A and B. For example, let $A = \{0, 1, 2\}, B = \{3, 4\}$, what is $A \times B$?

1.3 The size of a finite set

Why we care about the size of a set? Well, I often get asked how many students I have in this class. Let's consider some examples. $\#\{\text{`}a\text{'}, \text{`}b\text{'}, \text{`}c\text{'}\} = 3$, $\#\emptyset = 0$. But how do we calculate the size of a finite set in general? Can we just count the listed elements?

Definition 3 (size of a finite set)**.** *Let A be a finite set. We write $\#A$ to mean the size of set A. If A is an empty set, then we define $\#A = 0$. If A is not an empty set, then it must be that $A = B \cup \{a\}$ for some $a \in A$ and $a \notin B$ for some set B, thus we define $\#A = \#B + 1$.*

For example, $\#\underbrace{\{0,1,1,2,2\}}_{\{0\}\cup\{1,1,2,2\}} = \#\underbrace{\{1,1,2,2\}}_{\{1\}\cup\{2,2\}} + 1 = \#\underbrace{\{1,1\}}_{\{1\}\cup\emptyset} + 1 + 1 = \#\emptyset + 1 + 1 + 1 = 3.$

The following are some theorems about the size of finite sets, we will be able to prove them in our later class.

Theorem 3. *Let A, B be finite sets.*

1. $\#(A \cup B) = \#A + \#B - \#(A \cap B)$
2. $\#\text{Pow}(A) = 2^{\#A}$

Can we at least verify the above theorem by some examples?

Note that infinite sets have sizes too! There are different kind of infinite sizes, e.g. it can be shown that the size of $\text{Pow}(\mathbb{N})$ is strictly larger than the size of $\mathbb{N}$. But we will have to defer this interesting topic to later of the class.

1.4 Partitions of a set

What is a partition of a set? For example, let $A = \{0, 1, 2, 3, 4\}$, then $B_1 = \{0, 1\}$, $B_2 = \{2, 3, 4\}$ is a partition. Note that $B_1 \cup B_2 = A$ and $B_1 \cap B_2 = \emptyset$. We also call B_1, B_2 a 2-partition of A. In general, there can be n-partition of a set. Partition is useful for proving theorems about the sizes of finite sets.

Definition 4. *Let A be a set, and $S_1, ..., S_n \subseteq A$. We say $S_1, ..., S_n$ form a n-partition of A if $A = S_1 \cup S_2 ... \cup S_n$ (or $A = \bigcup_{i=1}^{n} S_i$) and $S_i \cap S_j = \emptyset$ for $i \neq j$ and $1 \leq i, j \leq n$.*

Another example of 2-partition would be natural numbers can be partitioned into even and odd numbers. We will again left the proof of the following theorem for later.

Theorem 4. *Let S be a finite set. If S_1, S_2 is a 2-partition for S, then $\#S = \#S_1 + \#S_2$. This generalizes to n-partition of S as well. So if $S_1, ..., S_n$ is a n-partition for S, then $\#S = \#S_1 + \#S_2 + ... + \#S_n$.*

Proof. By definition of 2-partition, $S_1 \cap S_2 = \emptyset$. So by Theorem 3, we have $\#S = \#S_1 + \#S_2 - \#(S_1 \cap S_2) = \#S_1 + \#S_2$. □

The following is another application of n-partition.

Theorem 5. *Let A, B be finite sets, we have $\#(A \times B) = \#A \times \#B$.*

Proof. The following is a sketch of the proof.

Since A, B are finite sets, we have $A = \{a_1, a_2, ..., a_n\}$ for some $n \in \mathbb{N}$ and $B = \{b_1, b_2, ..., b_l\}$ for some $l \in \mathbb{N}$. So $\#A = n$ and $\#B = l$. Our goal is to show $\#(A \times B) = n \times l$.

Let $S_i = \{x \mid b \in B, x = (a_i, b)\}$ for $1 \leq i \leq n$. Note that $\#S_i = l$, because a_i is fixed. We can verify that $S_1, ..., S_n$ form a n-partition for S. Now by Theorem 4, we have $\#S = \#S_1 + \#S_2 + ... + \#S_n = \underbrace{l + l + ... + l}_{n} = n \times l = \#A \times \#B$.

□

2 Functions

Why do we care about functions? In computer science, function is a useful concept to describe the computational process, so functions can be represented by programs. For example, in programming, the program that sorts a list is a function. A program that search information from a database can also be viewed as a function.

Definition 5. *A function f from input set A to output set B (we write $f : A \to B$) is a mapping such that every $a \in A$ is mapped to a unique $b \in B$ (we write $f(a)$ for b in this case). The input set A is also called domain, the output set is also called range/codomain.*

Can you give me a simple concrete example of a function?

I can give you one: $(+) : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, where $(x, y) \mapsto x + y$ for any $(x, y) \in \mathbb{N} \times \mathbb{N}$. Here the notation $\mapsto$ means the mapping action (pronounced as *maps to*).

We can compose functions together to form new functions. For example, composing addition and multiplication give us polynomials. For example, say we have two functions $f(x) = 1 + x, g(x) = x^2$, then we can define a new function $h(x) = f(g(x)) = 1 + x^2$.

Definition 6 (Function composition)**.** *If $f : A \to B$ and $g : B \to C$ are functions, then we define $g \cdot f : A \to C$ as the composition of f and g, so $(g \cdot f)(a) = g(f(a))$ for every $a \in A$.*

2.1 Boolean functions

Most functions have infinite domain, but that is not to say function with finite domain are useless. In fact, there are plenty of useful functions with domains constructed from a two-element set $\mathbb{B} = \{\mathsf{T}, \mathsf{F}\}$.

Definition 7 (Some basic boolean functions)**.**

- *Boolean not function $(\neg) : \mathbb{B} \to \mathbb{B}$ is defined as $\neg\ \mathsf{F} = \mathsf{T}$ and $\neg\ \mathsf{T} = \mathsf{F}$.*
- *Boolean and function $(\wedge) : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ is defined as $\mathsf{F} \wedge \mathsf{F} = \mathsf{F}, \mathsf{F} \wedge \mathsf{T} = \mathsf{F}, \mathsf{T} \wedge \mathsf{F} = \mathsf{F}, \mathsf{T} \wedge \mathsf{T} = \mathsf{T}$.*

- *Boolean or function* $(\vee) : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ *is defined as* $\mathsf{F} \vee \mathsf{F} = \mathsf{F}, \mathsf{F} \vee \mathsf{T} = \mathsf{T}, \mathsf{T} \vee \mathsf{F} = \mathsf{T}, \mathsf{T} \vee \mathsf{T} = \mathsf{T}$.
- *Boolean implication function* $(\Rightarrow) : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ *is defined as* $\mathsf{F} \Rightarrow b = \mathsf{T}, \mathsf{T} \Rightarrow \mathsf{T} = \mathsf{T}, \mathsf{T} \Rightarrow \mathsf{F} = \mathsf{F}$.

Theorem 6. *Let* $a, b \in \mathbb{B}$. *We have the followings.*

- $\neg(\neg a) = a$.
- $\neg(a \wedge b) = (\neg a) \vee (\neg b)$.
- $\neg(a \vee b) = (\neg a) \wedge (\neg b)$.
- $a \Rightarrow b = (\neg a) \vee b$

Let us define a majority function majority : $\mathbb{B} \times \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ by composing $\wedge, \vee$. The majority function outputs T only when at least two of the inputs are T, otherwise it ouputs F.

Solution: $\text{majority}(a, b, c) = (a \wedge b) \vee (a \wedge c) \vee (b \wedge c)$.

2.1.1 Intuition about the boolean implication function

While the meaning of boolean not, and, or functions are fairly intuitive, the boolean implication function however seems a bit counter intuitive.

The following is my attempt to explain it. Given a condition $a \Rightarrow b$, and a situation about statements a and b. We would like to check if this particular situation *satisfies* the condition $a \Rightarrow b$. If the statement a does not holds, then we claim the situation satisfies the condition $a \Rightarrow b$. If the statement a holds, then the condition $a \Rightarrow b$ is satisfied only when the statement b holds. Note that this explanation does not talk about the validity or truthfulness of the condition $a \Rightarrow b$, nor does it talk about the validity or truthfulness of a and b, it only talk about whether a situation satisfies the condition $a \Rightarrow b$.

For example, consider the definition of subset relation $A \subseteq B$, it is the statement: "For every a, **if** $a \in A$, **then** $a \in B$". Now, Consider the situation $A = \emptyset$, $B = \{0, 1, 2\}$. In this case clearly $a \in A$ does not hold, therefore this situation satisfies the statement: "For every a, **if** $a \in A$, **then** $a \in B$". That is why $\emptyset \subseteq B$. Consider the situation $A = \{1\}, B = \emptyset$. In this case $1 \in A$, but $1 \notin B$, so this situation does not satisfies the statement "For every a, **if** $a \in A$, **then** $a \in B$". Consider the situation $A = \{1\}, B = \{1\}$. In this case $1 \in A$ and $1 \in B$, so this situation satisfies the statement "For every a, **if** $a \in A$, **then** $a \in B$".

Thus $\mathsf{F} \Rightarrow b = \mathsf{T}$ means that in the situation $a = \mathsf{F}$, $b = \mathsf{T}$ or $b = \mathsf{F}$, it satisfies the condition $a \Rightarrow b$. And $\mathsf{T} \Rightarrow \mathsf{T} = \mathsf{T}$ means that in the situation $a = \mathsf{T}$ and $b = \mathsf{T}$, the situation satisfies the condition $a \Rightarrow b$. And $\mathsf{T} \Rightarrow \mathsf{F} = \mathsf{F}$ means that in the situation $a = \mathsf{T}$ and $b = \mathsf{F}$, the situation does not satisfy the condition $a \Rightarrow b$.

2.2 Injective, surjective and bijective functions

There are three properties of functions that you should know.

Definition 8.
- *A function* $f : A \to B$ *is called injective if the following holds. If* $f(a) = f(b)$ *for some* $a, b \in A$, *then* $a = b$. *Alternatively, for every* $a, b \in A$, *if* $a \neq b$, *then* $f(a) \neq f(b)$.
- *A function* $f : A \to B$ *is called surjective if the following holds. For every* $b \in B$, *there exists an* $a \in A$ *such that* $f(a) = b$.
- *A function* $f : A \to B$ *is called bijective if it is both injective and surjective.*

There are some examples at Figure 1.

For an injective function, the inputs correspond to different outputs.

Questions: Is the addition function $(+)/\neg/\wedge/\vee$ an injective/surjective/bijective function?

Definition 9. *If* $f : A \to B$ *is a bijective function, we define its inverse* $f^{-1} : B \to A$ *to be the function such that* $f^{-1}(f(a)) = a$ *for every* $a \in A$.

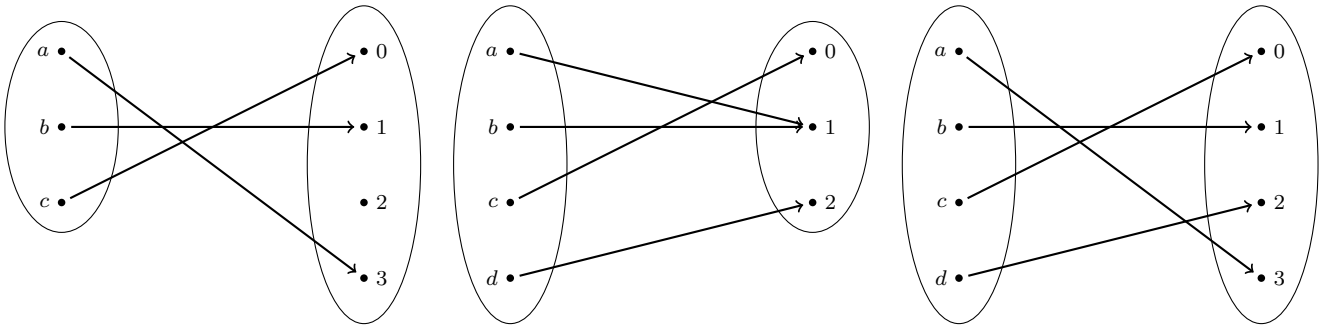


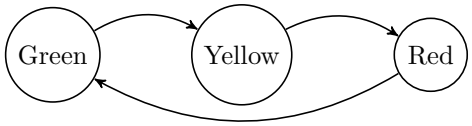
Figure 1: Examples of injective, surjective and bijective functions

3 Relations

Relation is a concept that is useful to characterize the connections between different sets. Relation is widely used in computer science for describing database (i.e., a collection of information about some entity) and for describing state transitions.

Definition 10. *A relation R on sets A, B is a subset $R \subseteq A \times B$. For $(a, b) \in R$, we often write aRb.*

It is common to talk about relation on set A, i.e., subsets of $A \times A$. And what is nice about relation on A is that it also admits diagram representation. Consider $A = \{\text{Red}, \text{Green}, \text{Yellow}\}$.



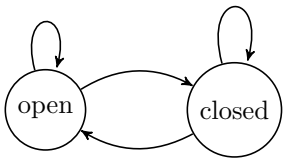
So the above diagram describes a transition relation of traffic lights,
i.e., $R = \{(\text{Green}, \text{Yellow}), (\text{Yellow}, \text{Red}), (\text{Red}, \text{Green})\}$.
Similarly, we can also describe an elevator moving relation on the set $\{\text{L1}, \text{L2}, \text{L3}\}$.
There are relation on infinite set as well, e.g. less $= \{(a, b) \mid a, b \in \mathbb{N}, a < b\}$ is a relation on $\mathbb{N}$.

3.1 Properties of relations

Definition 11. *Let R is a relation on A.*

- *R is reflexive: for every $a \in A$, we have a R a.*
- *R is symmetry: for every $a, b \in A$, if a R b, then b R a.*
- *R is transitive: for $a, b, c \in A$, if a R b and b R c, then a R c.*

Questions: Is the traffic lights/elevator/less relation reflexive/symmetry/transitive?
Consider another example, an automatic door that has two states $\{\text{open}, \text{closed}\}$. The following is a relation that characterizes the behavior of the door.



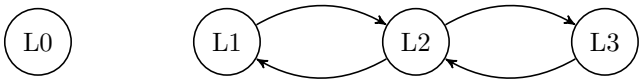
Is this relation reflexive/symmetry/transitive?

3.2 Closure operations on relations

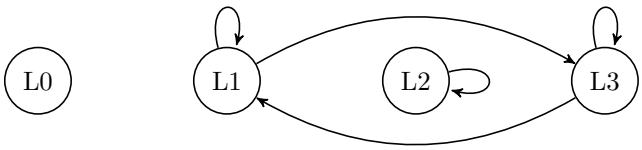
Like functions, we can compose relations to obtain new relations.

Definition 12.
Let R, S be relations on set A, we define the composition of R and S as the following.
$R \circ S = \{(a, c) \mid$ *there exists* $b \in A, (a, b) \in R, (b, c) \in S\}$.

Consider the elevator relation E on set $\{\text{L0}, \text{L1}, \text{L2}, \text{L3}\}$ (Let's say the basement does not have access to elevator).



What is the relation $E \circ E$?



What is the relation $(E \circ E) \circ E$?

Definition 13 (N-fold composition)**.** *Let R be a relation on A. We define the n-fold composition of R.*

- $R^0 = \{(a, a) \mid a \in A\}$.
- $R^{n+1} = R^n \circ R$.

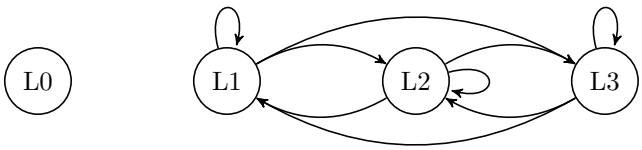
Definition 14 (Inverse relation)**.** *Let R be a relation on A, we define the inverse of R as $R^{-1} = \{(b, a) \mid (a, b) \in R\}$.*

Often the relation does not have the desirable property(such as reflexive, symmetry and transitive), so we can use the following closure operations to obtain a larger relation with such property.

Definition 15 (Closure operations)**.** *Let R be a relation on A.*

- *The reflexive closure of R is defined as $\{(a, a) \mid a \in A\} \cup R$.*
- *The symmetric closure of R is defined as $R^{-1} \cup R$.*
- *The transitive closure of R is defined as $R^+ = \bigcup_{n \in \mathbb{N}} R^n$ (notation for $R^0 \cup R^1 \cup R^2 \cup$ ...). In another word, the transitive closure of R is the smallest transitive relation on A that contains R.*

What is the relation $E \cup (E \circ E)$?



What is the relation E^+?

3.3 Equivalence relation

Definition 16. *A relation R is called equivalence relation if it is reflexive, symmetry and transitive. We write $a \sim b$ if a R b.*

Definition 17. *Let $a, b \in \mathbb{Z}$. We define $a|b$ if there exists a k such that $ka = b$.*

Example 1 (Congruence modulo n)**.** *Let $n \in \mathbb{Z}$ and $n > 0$. We define $a \equiv b \pmod{n}$ if $n|(a-b)$. Let $(\equiv_n) = \{(a,b) \mid (a,b) \in \mathbb{Z} \times \mathbb{Z}, a \equiv b \pmod{n}\} \subseteq \mathbb{Z} \times \mathbb{Z}$.*

Theorem 7. *$\equiv_n$ is an equivalence relation.*

Definition 18. *Let R be an equivalence relation and $a \in A$. We define $[a] = \{x \mid x \in A, x \sim a\}$. We call $[a]$ the equivalence class of a.*

Theorem 8. *Let R be an equivalence relation on A, and $a, b \in A$. Either $[a] = [b]$, or $[a] \cap [b] = \emptyset$.*

The above theorem implies that if there is an equivalence relation $\sim$ on the set A, then there is a *partition* on the set A.

Definition 19. *Let A be a set and $\sim$ be an equivalence relation on A. We define $A/_\sim = \{[a] \mid a \in A\}$.*

4 Basic Logic and Proof Methods

Logic and deduction are important for both mathematics and computer science. In mathematics, they provide means to formulate and prove theorems. In computer science, they allow us to reason about the correctness of the programs, to program automated reasoning systems, to build intelligent systems.

4.1 Basic concepts in logic

The following concepts are essential in logic: *proposition*, *predicate*, *implication*, *negation*, *conjunction*, *disjunction*, the *forall* and *exists* quantifiers.

Proposition. A proposition is a statement. E.g. "Today is Monday", "It is sunny today". We use capital letters P, Q, S to denote a proposition, they are called *propositional variables*.

Predicate. A predicate is an incomplete statement. E.g. "_ is even" is an incomplete statement (which can be written as $P(\_)$). We can fill in 3, then we get a statement "3 is even". In general, we fill in a variable x get a complete statement and use substitution to talk about specific instances of x. For example, we write $P(x)$ to mean "x is even", then $P(3)$ gives the statement "3 is even".

The most basic forms of statements are coming from propositions and predicates. We can compose these basic statements to obtain more statements using the followings.

Implication. "**If** x and y are even, **then** $x + y$ is even.", "x is an odd number **implies** that it is not divisible by 2". We write $A \Rightarrow B$ to denote the statement A implies the statement B.

Negation. " x is **not** even.", We write $\neg A$ to denote the negation of the statement A.

Conjunction. " x is an even number **and** x is a prime number.", We write $A \wedge B$ to denote the conjunction of A, B.

Disjunction. " x is an even number **or** x is a prime number.", We write $A \vee B$ to denote the disjunction of A, B.

Forall. "**For every** $x \in \mathbb{N}$, if x is even , then x is not odd." We write $\forall x.(x \in \mathbb{N} \wedge \text{Even}(x)) \Rightarrow \neg\text{Odd}(x)$. If A is a statement, then $\forall x.A$ is a statement.

Exists. "**There exists** $x \in \mathbb{N}$ such that x is even and x is prime.", it can be translated to $\exists x.x \in \mathbb{N} \Rightarrow \text{Even}(x) \wedge \text{Prime}(x)$. $\text{Even}(x)$ is defined as $\exists k.k \in \mathbb{N} \Rightarrow x = 2k$. $\text{Odd}(x)$ is defined as $\exists k.k \in \mathbb{N} \Rightarrow x = 2k+1$. If A is a statement, then $\exists x.A$ is also a statement.

As an exercise, try to translate a theorem into a statement that consists of $\forall, \exists, \Rightarrow, \neg, \wedge, \vee$.

4.2 Deduction

An *axiom* is a statement that is assumed to be true. Deduction (or proof) is a process to establish the validity of a statement based on existing axioms. For example, let "Socrates is a human" and "all men must die" be axioms. Then by rule of deduction, we can establish that "Socrates must die". All this sounds plausible, but on what basis can we conclude "Socrates must die" ?

The followings are some deduction rules that we commonly use.

- **Modus ponens**: From statements A and $A \Rightarrow B$, we conclude B.
- **Instantiation**: We write $A[x]$ to mean A is a statement contains the variable x. From a statement $\forall x.A[x]$, we conclude $A[t]$ for any individual t.
- **And-elimination-1**: From statement $A \wedge B$, we can conclude A.
- **And-elimination-2**: From statement $A \wedge B$, we can conclude B.
- **Or-introduction-1**: From statement A, we can conclude $A \vee B$.
- **Or-introduction-2**: From statement B, we can conclude $A \vee B$.
- **Exist-introduction**: From statement $A[t]$, we can conclude $\exists x.A[x]$.
- **Principle of explosion($\perp$-elimination)**: Contradiction (denoted by $\perp$) are usually of the forms $\neg A \wedge A$, or $(A \Rightarrow \neg A) \wedge (A \Rightarrow \neg A)$, or it can also be not obeying basic facts of arithmetic (e.g. $0 = 1$, $a|1$ for $a > 1$.). From a contradiction, we can conclude any statement B, i.e., $\perp \Rightarrow B$.

Now let $H(x)$ be the statement " x is a human" and $D(x)$ means "x must die". Then "Socrates is human" corresponds to $H(\text{Socrates})$, and "all men must die" corresponds to $\forall x.H(x) \Rightarrow D(x)$. By instantiation, we have $H(\text{Socrates}) \Rightarrow D(\text{Socrates})$. By modus ponens, we have $D(\text{Socrates})$.

4.3 Constructive proof methods

To prove a statement of the form $\forall x.A[x]$, we prove $A[y]$, where y is a fresh variable. To prove a statement of the form $\forall x.P(x) \Rightarrow Q(x)$, we assume $P(y)$ and try to prove $Q(y)$.

Theorem 9. *For every $x \in \mathbb{N}$, if x is odd, then x^2 is odd.*

Proof.
Suppose $x \in \mathbb{N}$ is odd.
By definition of odd numbers, we know that $x = 2a + 1$ for some $a \in \mathbb{N}$.
By basic arithmetic, we have $x^2 = (2a + 1)(2a + 1) = 4a^2 + 4a + 1 = 2(2a^2 + 2a) + 1$.
Thus x^2 is odd.

□

To prove a statement of the form $A \vee B \Rightarrow C$, we have to prove both $A \Rightarrow C$ and $B \Rightarrow C$.

Theorem 10. *Let A, B, C be sets. If $C \cap A = \emptyset$ and $C \cap B = \emptyset$, then $(A \cup C) \cap (B \cup C) = (A \cap B) \cup C$.*

Proof. Assume $C \cap A = \emptyset$ and $C \cap B = \emptyset$.

- We first prove that $(A \cup C) \cap (B \cup C) \subseteq (A \cap B) \cup C$. Let $x \in (A \cup C) \cap (B \cup C)$. By definition of set union and intersection, we have $x \in A \cup C$ **and** $x \in B \cup C$. Thus ($x \in A$ **or** $x \in C$) **and** ($x \in B$ **or** $x \in C$). This implies four possibilities: $x \in A, x \in B$, **or** $x \in A, x \in C$, **or** $x \in C, x \in B$, **or** $x \in C$.

  Since $x \in A, x \in C$ and $C \cap A = \emptyset$ give us a contradiction, by principle of explosion, we conclude $x \in (A \cap B) \cup C$. Similarly for $x \in C, x \in B$.

  Consider the case $x \in A, x \in B$, by definition of set union, we $x \in (A \cap B) \cup C$.

  Consider the case $x \in C$, we also have $x \in (A \cap B) \cup C$.

  Thus $(A \cup C) \cap (B \cup C) \subseteq (A \cap B) \cup C$.

- Let $x \in (A \cap B) \cup C$. This implies that $x \in A \cap B$ or $x \in C$.

  Suppose $x \in C$. This implies that ($x \in C$ **or** $x \in A$) **and** ($x \in C$ **or** $x \in B$). So $x \in (A \cup C) \cap (B \cup C)$.

  Suppose $x \in A$ and $x \in B$. This implies that ($x \in C$ **or** $x \in A$) **and** ($x \in C$ **or** $x \in B$). So $x \in (A \cup C) \cap (B \cup C)$.

□

To prove a negation $\neg P$, we assume P and try to derive a contradiction.

Theorem 11. *There is no smallest rational number greater than* 0*.*

Proof. Suppose there is a smallest rational number r.
But we have $0 < r/2 < r$, this implies that r is not the smallest rational number.
Contradiction. So there is no smallest rational number greater than 0.

□

Theorem 12. *There is no natural number that can be both even and odd.*

Proof. Suppose there is a number r that is even and odd.
Then there exists $k_1, k_2 \in \mathbb{N}$ such that $r = 2k_1 = 2k_2 + 1$.
This implies $2(k_1 - k_2) = 1$, where $k_1 - k_2 \in \mathbb{N}$.
This implies $2|1$, contradiction. □

Theorem 13 (Cantor's theorem)**.** *There does not exists a surjective function from* $\mathbb{N}$ *to* $\text{Pow}(\mathbb{N})$*.*

Proof. Suppose there is a surjective function $f : \mathbb{N} \to \text{Pow}(\mathbb{N})$.
By definition of surjective function, for every $A \in \text{Pow}(\mathbb{N})$, there exist a number $n \in \mathbb{N}$ such that $f(n) = A$.
Define $S = \{x \mid x \in \mathbb{N}, x \notin f(x)\}$.
Since $S \subseteq \mathbb{N}$, we have $S \in \text{Pow}(\mathbb{N})$.
Since f is surjective, there exist a n such that $f(n) = S$.
Now suppose $n \in S$, this means $n \notin f(n) = S$.
Suppose $n \notin S$, this means $n \in S$. Hence contradiction. □

Cantor's theorem has a fundamental impact in mathematics. It implies that the size of power set of natural numbers is in a sense strictly larger than the size of natural numbers, even though both are infinite sets. It means some infinite set are strictly larger than the other!

4.3.1 Proof by induction

Definition 20 (Induction)**.** *To prove a statement $A[n]$ holds for any natural number n: We first prove that $A[0]$ holds. Then let $n \in \mathbb{N}$, we assume $A[n]$ holds (this assumption is called inductive hypothesis), we prove that $A[n+1]$ holds.*

Why does induction make sense? Well, say our goal is to prove $A[n]$ holds for any natural number n. One way to prove it is to make sure all of $A[0], A[1], A[2], ...$ hold. Say we manage to prove $A[0]$, and we manage to prove $A[n] \Rightarrow A[n+1]$ hold for any $n \in \mathbb{N}$. Then by modus ponens, we can show that $A[1]$ holds, $A[2]$ holds, and so on. Therefore we conclude that $A[n]$ holds for any n.

Theorem 14. *Every natural number is either even or odd. ($\forall x. x \in \mathbb{N} \Rightarrow (\text{Even}(x) \vee \text{Odd}(x))$)*

Proof. Suppose $x \in \mathbb{N}$. We need to show $\text{Even}(x) \vee \text{Odd}(x)$. By induction on x, we consider the following cases.

Base case. $x = 0$. Since $\text{Even}(0)$ holds. We have $\text{Even}(0) \vee \text{Odd}(0)$.

Step case. Assume $\text{Even}(x) \vee \text{Odd}(x)$ holds.

Suppose $\text{Even}(x)$ holds, then we have $\text{Odd}(x+1)$, hence $\text{Odd}(x+1) \vee \text{Even}(x+1)$.

Suppose $\text{Odd}(x)$ holds, then we have $\text{Even}(x+1)$, hence $\text{Odd}(x+1) \vee \text{Even}(x+1)$.

Thus $\text{Odd}(x+1) \vee \text{Even}(x+1)$. □

Theorem 15. $1 + 2 + 3 + ... + n = \frac{n(n+1)}{2}$

Proof. By induction on n.

Base case. $n = 0$, we have $0 = 0$.

Step case. Assume $1 + 2 + 3 + ... + n = \frac{n(n+1)}{2}$, we need to show $1 + 2 + 3 + ... + n + (n+1) = \frac{(n+1)(n+2)}{2}$.

$(1 + 2 + 3 + ... + n) + (n+1) \stackrel{IH}{=} \frac{n(n+1)}{2} + (n+1) = \frac{(n+1)(n+2)}{2}$. □

Theorem 16. $1 + 2 + 2^2 + ... + 2^n = 2^{n+1} - 1$

Proof. By induction on n.

Base case. $n = 0$, we have $1 = 1$.

Step case. Assume $1+2+2^2+...+2^n = 2^{n+1}-1$, we need to show $1+2+2^2+...+2^n+2^{n+1} = 2^{n+1+1}-1$.

$1 + 2 + 2^2 + ... + 2^n + 2^{n+1} \stackrel{IH}{=} 2^{n+1} - 1 + 2^{n+1} = 2^{n+2} - 1$. □

4.4 Indirect proof methods

To prove $P \Rightarrow Q$, we prove $\neg Q \Rightarrow \neg P$ instead.

Theorem 17. *For every $n \in \mathbb{N}$, if n^2 is even, then n is even.*

Proof. Suppose n is not even.

By Theorem 12 and Theorem 14, we conclude that n is odd.

By Theorem 9, we have n^2 is odd.

By Theorem 12 and Theorem 14, we conclude n^2 is not even.

Thus For every $n \in \mathbb{N}$, if n^2 is even, then n is even. □

Using **law of excluded middle** $A \vee \neg A$.

Theorem 18. *Let $\sim$ be an equivalence relation on A, and $a, b \in A$. Either $[a] = [b]$, or $[a] \cap [b] = \emptyset$.*

Proof. By law of excluded middle, $a \sim b \vee a \not\sim b$. Suppose $a \sim b$. Then for every $x \in [a]$, we have $x \sim a \sim b$. So $x \in [b]$. Similarly, for every $y \in [b]$, we have $y \in [a]$. So we prove that $[a] = [b]$.

Suppose $a \not\sim b$. We need to show $[a] \cap [b] = \emptyset$. By contrapositive, we can assume $[a] \cap [b] = \emptyset$ and prove that $a \sim b$. Suppose $[a] \cap [b] \neq \emptyset$. This implies there exists $c \in A$ such that $c \in [a]$ and $c \in [b]$. So $a \sim c \sim b$.

Thus either $[a] = [b]$, or $[a] \cap [b] = \emptyset$. □

Theorem 19. *Recall that the definition of exponentiation can be extended to allow any real exponent. There exists irrational numbers a and b such that a^b is rational.*

Proof. We know that $\sqrt{2}$ is an irrational number.

By law of excluded middle, either $\sqrt{2}^{\sqrt{2}}$ is rational or it is not.

Suppose $\sqrt{2}^{\sqrt{2}}$ is rational, then in this case $a = b = \sqrt{2}$.

Suppose $\sqrt{2}^{\sqrt{2}}$ is not rational. Let $a = \sqrt{2}^{\sqrt{2}}$ and $b = \sqrt{2}$. By property of exponentiation, we have $(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{\sqrt{2} \times \sqrt{2}} = \sqrt{2}^2 = 2$. So $a = \sqrt{2}^{\sqrt{2}}, b = \sqrt{2}$. □

Note that any proofs using law of excluded middle (LEM) is consider indirect proof. One side effect of proving an existential statement using LEM is that it does not give us the exact *witnesses*. In the above theorem, we do not know whether a, b should be $\sqrt{2}, \sqrt{2}$, or $\sqrt{2}^{\sqrt{2}}, \sqrt{2}$.

Why contrapositive proof is also a kind of indirect proof? The correctness of positive proof is based on LEM as well. To show $\neg Q \Rightarrow \neg P$ implies $P \Rightarrow Q$, we assume P and $\neg Q \Rightarrow \neg P$, and try to conclude Q. By LEM, $Q \vee \neg Q$. Suppose Q, we are done. Suppose $\neg Q$, by modus ponens, we have $\neg P$, which contradicts our assumption P. So by law of explosion, we can conclude Q.

Proof by contradiction: to prove F, we assume $\neg F$ and try to derive $\bot$.

5 A brief introduction to propositional logic and satisfiability

Definition 21 (Formulas of propositional logic)**.** *The formulas of propositional logic are of the following forms.*

$F ::= p \mid F_1 \Rightarrow F_2 \mid \neg F \mid F_1 \vee F_2 \mid F_1 \wedge F_2$

The truth tables for the propositional formulas are exactly the same as the description of boolean functions.

Definition 22 (Satisfiable)**.** *An assignment or valuation is a mapping from a set of propositional variables to their corresponding truth values. The formula G is satisfiable by ρ (written as $\rho \models G$) if G is evaluated to T by the assignment ρ.*

Definition 23. *A formula G is semantically valid (written as $\models G$) if for every possible assignment ρ, we have $\rho \models G$. The formula G is also called tautology.*

Definition 24 (Semantic entailment)**.** *We say F entails G (written as $F \models G$) if for every assignment ρ, $\rho \models F$ implies $\rho \models G$.*

Definition 25. *We say F and G are semantically equivalent if $F \models G$ and $G \models F$.*

Theorem 20 (Deduction theorem)**.** *If $G \models F$, then $\models G \Rightarrow F$.*

Theorem 21. *F is satisfiable iff $\neg F$ is not valid.*

Definition 26 (Conjunctive normal form)**.** *Let a literal be $L = \neg p \mid p$. A conjunctive normal form is of the form $C_1 \wedge C_2 \wedge ... \wedge C_n$, where $C_i = L_1 \vee L_2 \vee ... \vee L_m$. We often call each C_i a disjunctive clause.*

Given a CNF, how to determine its validity?

Theorem 22. *A disjunction of literals $L_1 \vee L_2 \vee .. \vee L_m$ is valid iff there are $1 \leq i, j \leq m$ and $i \neq j$ such that $L_i = \neg L_j$.*

Proof. Exercise. □

So to determine if a CNF $C_1 \wedge C_2 \wedge ... \wedge C_n$ is valid, we need to check C_i is valid for $1 \leq i \leq n$. To check if $C_i = L_1 \vee L_2 \vee ... \vee L_m$ is valid, we just find a L_i, L_j such that $L_i = \neg L_j$.

For example, determine the validity of $(\neg q \vee p \vee r) \wedge (\neg p \vee r) \wedge q$.

How to convert a formula into CNF? We can use the following process to convert a formula into CNF.

1. **Remove implications**: $F \Rightarrow G = \neg F \vee G$.
2. **Propagate negations**: $\neg(\neg F) = F$, $\neg(F_1 \wedge F_2) = \neg F_1 \vee \neg F_2$ and $\neg(F_1 \vee F_2) = \neg F_1 \wedge \neg F_2$.
3. **Or distributions**: $(F_1 \wedge F_2) \vee G = (F_1 \vee G) \wedge (F_2 \vee G)$, $G \vee (F_1 \wedge F_2) = (G \vee F_1) \wedge (G \vee F_2)$.

For example, convert $\neg p \wedge q \Rightarrow p \wedge (r \Rightarrow q)$ to CNF.

Other use of CNF, we can synthesize a boolean formula from a truth table.

The key application of CNF is in solving satisfiability (SAT problem). As we know, SAT is trivial if the formula is in DNF (disjunctive normal form), but most problem are given in CNF, and converting CNF to DNF is impractical since it increases the number of clauses exponentially.

Algorithm for solving SAT problem for CNF does exist, a lot of modern SAT solvers are based DPLL (Davis-Putnam-Logemann-Loveland) algorithm. The basic DPLL algorithm assume a formula is in CNF, and try to reduce the number of guesses via the notion of *unit clause*. A unit clause is a disjunctive clause where only one variable's truth value are unknown. For example, p is a unit clause since p's truth value is unknown. If $p_1 = p_2 = \mathsf{F}$, then $p_1 \vee p_2 \vee \neg p_3$ is a unit clause because p_3's truth value is unknown. In DPLL, since we aim to find a satisfiable assignment, so we set the truth value of the variable in a unit clause to make the unit clause truth.

Definition 27 (A basic DPLL algorithm)**.**

1. ***Guess*** *a truth value of a propositional variable.*
2. ***Deduce*** *the truth value of a propositional variable from a unit clause.*
3. ***Backtrack*** *if a contradiction is reached, flip the truth value of the previous guess and resume.*

Example 2. *We will use comma to denote the conjunction, consider the CNF:* $x_2 \vee \neg x_3 \vee x_4, \neg x_1 \vee \neg x_2, \neg x_1 \vee \neg x_3 \vee \neg x_4, x_1$.

1.	***Deduce***	$x_1 = \mathsf{T}$	$x_2 \vee \neg x_3 \vee x_4, \cancel{\neg x_1} \vee \neg x_2, \cancel{\neg x_1} \vee \neg x_3 \vee \neg x_4, \cancel{x_1}$
2.	***Deduce***	$x_1 = \mathsf{T}, x_2 = \mathsf{F}$	$\cancel{x_2} \vee \neg x_3 \vee x_4, \cancel{\neg x_2}, \neg x_3 \vee \neg x_4$
3.	***Guess***	$x_1 = \mathsf{T}, x_2 = \mathsf{F}, x_3 = \mathsf{T}$	$\cancel{\neg x_3} \vee x_4, \cancel{\neg x_3} \vee \neg x_4$
4.	***Backtrack*** *3*	$x_1 = \mathsf{T}, x_2 = \mathsf{F}, x_3 = \mathsf{F}$	$\cancel{\neg x_3 \vee x_4}, \cancel{\neg x_3 \vee \neg x_4}$
5.	***Success!***	$x_1 = \mathsf{T}, x_2 = \mathsf{F}, x_3 = \mathsf{F}$	$\emptyset$

6 Brief introduction to recursive definitions and functions

Recursive functions are important because they are essentially descriptions of algorithms. We have already encounter a recursive function, the size function for finite sets is a recursive function from set of finite sets to the set of natural numbers.

Definition 28 (Peano numbers)**.** *A Peano number(***PN***) is defined (generated) by the followings.*

- Z *is a Peano number.*
- *If* n *is a Peano number, then* $\mathsf{S}(n)$ *is a Peano number.*

Informally, Z corresponds to 0, and $\mathsf{S}(n)$ corresponds to the successor of the number n. We also call Z and S the *constructors* of natural numbers. We say Z is a *non-recursive constructor* and S is a *recursive constructor*.

Once we identify the constructors of the natural numbers, we can define function by *pattern matching* on the argument.

Definition 29. *The predecessor function* $\mathsf{pred} : \mathbf{PN} \to \mathbf{PN}$ *can be defined as the following.*
$\mathsf{pred}(\mathsf{Z}) = \mathsf{Z}$
$\mathsf{pred}(\mathsf{S}(n)) = n$

In the above definition, we pattern match on the input and decide what to do in each case.
Another example of recursive definition.

Definition 30 (Binary tree)**.** *A binary tree of number is defined (generated) by the followings.*

- $\mathsf{Leaf}(n)$ *is a binary tree, where* $n \in \mathbb{N}$*.*
- *If* t_1 *and* t_2 *are binary trees, then* $\mathsf{Node}(n, t_1, t_2)$ *is a binary tree, where* $n \in \mathbb{N}$*.*

Here we say Leaf and Node are the constructors of binary tree.

6.1 Recursive functions for Peano Numbers

Addition is usually thought of a primitive operation (taken as given) for natural numbers, but we can define addition as a recursive function for Peano numbers.

Definition 31 (General scheme to define recursive function)**.**

- *Basis definition: define the function for all the non-recursive constructors.*
- *Recursive definition: define the function for all the recursive constructors. In the definition, we can use the results of the function on the components of the recursive constructor.*

Definition 32 (Addition)**.**
$\mathsf{add} : \mathbf{PN} \times \mathbf{PN} \to \mathbf{PN}$
$\mathsf{add}(\mathsf{Z}, n) = n$
$\mathsf{add}(\mathsf{S}(m), n) = \mathsf{S}(\mathsf{add}(m, n))$

Definition 33 (Mulitiplication)**.**
$\mathsf{multiply} : \mathbf{PN} \times \mathbf{PN} \to \mathbf{PN}$
$\mathsf{multiply}(\mathsf{Z}, n) = \mathsf{Z}$
$\mathsf{multiply}(\mathsf{S}(m), n) = \mathsf{add}((\mathsf{multiply}(m, n)), n)$

One way to make sure the addition function is correct is to prove it behaves like addition. For Peano numbers, we have the following induction principle.

Definition 34 (Induction Principle for Peano Numbers)**.** *Let $A[n]$ be a statement about the Peano number n. We have the following induction principle for Peano number.*
$A[\mathsf{Z}] \wedge (\forall n.n \in \mathbf{PN} \wedge A[n] \Rightarrow A[\mathsf{S}(n)]) \Rightarrow \forall n.n \in \mathbf{PN} \Rightarrow A[n]$

Theorem 23. $\forall n.n \in \mathbf{PN} \Rightarrow \mathsf{add}(n, \mathsf{Z}) = n.$

Proof. We prove this theorem by induction.
Base case. $n = \mathsf{Z}$. By definition of add, we have $\mathsf{add}(\mathsf{Z}, \mathsf{Z}) = \mathsf{Z}$.
Step case. Let $n \in \mathbf{PN}$, we assume $\mathsf{add}(n, \mathsf{Z}) = n$ as induction hypothesis (IH). We have $\mathsf{add}(\mathsf{S}(n), \mathsf{Z}) = \mathsf{S}(\mathsf{add}(n, \mathsf{Z})) \overset{IH}{=} \mathsf{S}(n)$.
By induction, we conclude. □

Theorem 24. $\forall n.n \in \mathbf{PN} \Rightarrow \forall m.m \in \mathbf{PN} \Rightarrow \mathsf{add}(n, m) = \mathsf{add}(m, n).$

Proof. Exercise. □

6.2 Recursive function for Natural numbers

A lot of concept we encounter can be described using recursive function as well. For example, the informal summation $0 + 1 + 2 + 3 + ... + n$ can be formally described by the following recursive function.

Definition 35.
$\mathsf{sum} : \mathbb{N} \to \mathbb{N}$
$\mathsf{sum}(0) = 0$
$\mathsf{sum}(n + 1) = (n + 1) + \mathsf{sum}(n)$

The summation function we define above can also be written as the following piecewise function.

$$\mathsf{sum}(n) = \begin{cases} 0 & \text{if } n = 0 \\ n + \mathsf{sum}(n-1) & \text{if } n > 0 \end{cases}$$

We will stick with the format in Definition 35 in this note.

Another use of recursive function is to describe infinite sequences. Consider the well-known *Fibonacci sequence*: $0, 1, 1, 2, 3, 5, 8, 13, 21, ...$ The first two number of the sequence is $0, 1$, and from then on, every number in the sequence is the addition of previous two numbers. We can describe such sequence formally as a recursive function $\mathbb{N} \to \mathbb{N}$, where the input is the position of the sequence and the output is number at that position.

Definition 36 (Fibonacci function)**.**
$\mathsf{fib} : \mathbb{N} \to \mathbb{N}$
$\mathsf{fib}(0) = 0$
$\mathsf{fib}(1) = 1$
$\mathsf{fib}(n + 2) = \mathsf{fib}(n + 1) + \mathsf{fib}(n)$

We can verify the above definition by looking at $\mathsf{fib}(0), \mathsf{fib}(1), \mathsf{fib}(2), \mathsf{fib}(3), ...$
Consdier the sequence $0, 1, 3, 6, 10, 15, ...$ It can be described by the following recursive function.

Definition 37.
$f : \mathbb{N} \to \mathbb{N}$
$f(0) = 0$
$f(n + 1) = (n + 1) + f(n)$

Note that $0, 1, 3, 6, 10, 15, ...$ can be generated by $f(0), f(1), f(2), ...$

7 Counting: applied finite sets

Definition 38 (Product rule/multiplication principle)**.** *Let $A_1, ..., A_n$ be finite sets. Since $\#(A_1 \times ... \times A_n) = \#A_1 \times ... \times \#A_n$, there all $\#A_1 \times ... \times \#A_n$ possible ways to obtain a n-tuple.*

Example 3.

1. *How many functions are there from a set with m elements to a set with n elements?*
2. *How many injective functions are there from a set with m elements to one with n elements?*
3. *A standard postal code in Canada has 6 position, the first, third and fifth position must be a letter, the second, fourth, sixth position must be a number. For example, B3H 4J1. How many different postal codes are possible?*

   Answer: $26 \times 10 \times 26 \times 10 \times 26 \times 10$

Definition 39 (Addition principle)**.** *Let $A_1, ..., A_n$ be finite sets and $A_i \cap A_j = \emptyset$ for all $i \neq j$ and $1 \leq i, j \leq n$. Then $\#(A_1 \cup ... \cup A_n) = \#A_1 + ... + \#A_n$.*

Example 4. *How many length-4 non-repetitive lists can be made from the symbols A, B, C, D, E, F, G, if the list must contain an E?*

Definition 40 (Subtraction principle)**.** *If $S \subseteq A$, then $\#(A/S) = \#A - \#S$.*

Example 5. *How many length-4 lists can be made from the symbols A, B, C, D, E, F, G if the list has at least one E, and repetition is allowed?*

Definition 41 (Principle of inclusion–exclusion)**.** *Let A, B be finite sets. $\#(A \cup B) = \#A + \#B - \#(A \cap B)$.*

Example 6.

1. *A computer company receives 350 applications from computer graduates for a job planning a line of new Web servers. Suppose that 220 of these applicants majored in computer science, 147 majored in business, and 51 majored both in computer science and in business. How many of these applicants majored neither in computer science nor in business?*

7.1 Permutations: listings of a finite set

We can define recursive function on $\mathbb{N}$ as well, we can identify a natural number as either 0 or of the form $n + 1$ for some $n \in \mathbb{N}$.

Definition 42. *Let A be a finite set such that $\#A = n$ and $r \in \mathbb{N}$ such that $0 \leq r \leq n$. We write $P(n, r)$ to mean the possible listings of r elements of A. We define $P(n, r)$ recursively as a function $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ the following:*
$P(n, 0) = 1$.
$P(n, m + 1) = P(n, m) \times (n - m)$.

Note that the condition $m \leq n$ has to be met for $P(n, m)$ to be a well-defined function.

The justification of the definition of $P(n, m)$ is by product rule. We write $[i]$ to mean there are i possible choices to fill in a position. If we have m places, then $\underbrace{[n], [n-1], ..., [n-m]}_{m}$.

To extend m to $m + 1$, we have $\underbrace{[n], [n-1], ..., [n-m]}_{m}, [n - (m + 1)]$.

Definition 43 (Factorial function)**.** *We give a recursive definition of the factorial function* $! : \mathbb{N} \to \mathbb{N}$ *below (using postfix notation).*

$0! = 1$

$(n+1)! = (n+1) \times n!$

Observation: Let A be a finite set and $\#A = n$. There are $n!$ possible permutations (arrangements) for listing all the elements non-repeatedly in A.

Definition 42 is often informally written using ellipsis: $P(n,m) = \underbrace{n \times (n-1) \times ... \times (n-m+1)}_{m}$ assuming $0 < m \leq n$.

Theorem 25. *If* $n, m \in \mathbb{N}$ *and* $0 \leq m \leq n$*, then* $P(n,m) = \frac{n!}{(n-m)!}$.

Proof. By induction on m.

Base case: $m = 0$. We have $1 = 1$.

Step case: Assume $P(n,m) = \frac{n!}{(n-m)!}$ for any n such that $0 \leq m \leq n$ (Induction hypothesis).

Let $n' \in \mathbb{N}$ and $0 \leq m+1 \leq n'$. We have $P(n', m+1) = P(n', m) \times (n'-m) \stackrel{IH}{=} \frac{n'!}{(n'-m)!} \times (n'-m) = \frac{n'!}{(n'-(m+1))!}$. □

Example 7.

1. *How many ways are there to select a first-prize winner, a second-prize winner, and a third-prize winner from 100 different people who have entered a contest?*

7.2 Combinations: subsets of a finite set

Definition 44. *Let* A *be a finite set such that* $\#A = n$ *and* $r \in \mathbb{N}$ *such that* $0 \leq r \leq n$*. We write* $C(n,r)$*(sometimes* $\binom{n}{r}$*) to mean all the possible* r*-element subsets of* A*. We define* $C(n,r)$ *recursively as a function* $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *the following:*

$C(n,m) = 1$ *if* $m = n$ *or* $m = 0$.

$C(n+1, m+1) = C(n,m) + C(n, m+1)$ *otherwise.*

The justification for $C(n+1, m+1) = C(n,m) + C(n, m+1)$ (also called Pascal's identity, where $m+1 < n+1$) is by addition principle: since in this case $\#A = n+1$, $A = B \cup \{a\}$ where $a \notin B$. A subset E of A such that $\#E = m+1$ either contain a, in which case there are $C(n,m)$ possibilities; Or doesn't, in which case there are $C(n, m+1)$ possibilities.

Example 8. *1. How many ways are there to select five players from a 10-member tennis team to make a trip to a match at another school?*

Since there are $2^{\#A}$ all possible subsets for A, on the other hand, we can add $C(\#A, 0)$ (emptyset), $C(\#A, 1)$ (all the subsets of size 1),..., $C(\#A, \#A)$ (all the subsets of size $\#A$) together. This allows us to *discover* the following theorem[3].

Theorem 26. *For every* $n \in \mathbb{N}$*, we have* $C(n,0) + C(n,1) + ... + C(n,n) = 2^n$.

Proof. We prove this theorem by induction on n.

Base case: $n = 0$. In this case $C(0,0) = 1 = 2^0$

Step case: Let $n \in \mathbb{N}$, we assume $C(n,0) + C(n,1) + ... + C(n,n) = 2^n$ as induction hypothesis (IH). We want to prove $C(n+1,0) + C(n+1,1) + ... + C(n+1, n+1) = 2^{n+1}$. Since $C(n+1,0) = C(n+1, n+1) = 1$, we have the following

[3]This kind of discovery is also called combinatorial proof in some textbook.

$$C(n+1,0)+C(n+1,1)+...+C(n+1,n+1)=1+1+C(n+1,1)+C(n+1,2)+...+C(n+1,n)=$$
$$1+1+C(n,0)+C(n,1)+C(n,1)+C(n,2)+...+C(n,n-1)+C(n,n)=$$
$$(C(n,0)+C(n,1)+...+C(n,n-1)+1)+(1+C(n,1)+C(n,2)+...+C(n,n)) \stackrel{IH}{=} 2^n+2^n=2^{n+1}.$$

□

Another way of constructing $C(n,r)$ is by considering its relation to r-permutations $P(n,r)$. We know that $P(n,r)$ can be constructed by the following: first we obtain $C(n,r)$ r-subsets. Then for each r-subset, we do a permutation, so $P(n,r)=C(n,r)\times r!$.

Theorem 27. *For $n\in\mathbb{N}$ and $0\le r\le n$, we have $P(n,r)=C(n,r)\times r!$. By Theorem 25, we have $P(n,r)=\frac{n!}{(n-r)!}$. So we just need to prove $C(n,r)=\frac{n!}{(n-r)!\times r!}$.*

Proof. We prove by induction on n.

Base case: $n=0=r$. $C(n,r)=1=\frac{0!}{0!\times 0!}$.

Step case: Let $n\in\mathbb{N}$, assume for any $0\le r\le n$, we have $C(n,r)=\frac{n!}{(n-r)!\times r!}$ as inductive hypothesis(IH). We need to prove that for any $0\le r'\le n+1$, we have $C(n+1,r')=\frac{(n+1)!}{(n+1-r')!\times r'!}$.

Suppose $r'=n+1$ or $r'=0$. In this case $C(n+1,r')=\frac{(n+1)!}{(n+1-r')!\times r'!}=1$.

Suppose $0<r'<n+1$. In this case we have the following:

$$C(n+1,r')=C(n,r')+C(n,r'-1)\stackrel{IH}{=}\frac{n!}{(n-r')!\times r'!}+\frac{n!}{(n-(r'-1))!\times(r'-1)!}=$$
$$\frac{n!\times(n-r'+1)}{(n-r'+1)\times(n-r')!\times r'!}+\frac{n!\times r'}{(n-(r'-1))!\times(r'-1)!\times r'}=\frac{(n+1)!}{(n+1-r')!\times r'!}$$

□

Theorem 28. *If $0\le r\le n$, then $C(n,r)=C(n,n-r)$.*

Proof. Exercise. □

A well-known theorem in counting is the following *binomial theorem*.

Theorem 29 (Binomial theorem)**.** *Let x,y be variables and $n\in\mathbb{N}$. We have $(x+y)^n=C(n,0)\cdot x^n+C(n,1)\cdot x^{n-1}y+...+C(n,n-1)\cdot xy^{n-1}+C(n,n)\cdot y^n$.*

Intuitively, the binomial theorem makes sense because for example when we calculate $(x+y)^4=(x+y)(x+y)(x+y)(x+y)$, we must have the following terms x^4,x^3y,x^2y^2,xy^3,y^4. To determine the coefficient of x^4, the x must be coming from each of the sum, so it should be $C(4,4)=C(4,0)$. To determine the coefficient of x^3y, we can see x must coming from three of the four sums, so there are $C(4,3)$ ways to obtain it so the coefficient of x^3y is $C(4,3)=C(4,1)$.

Now let us prove the binomial theorem by induction.

Theorem 30. *Let x,y be variables and $n\in\mathbb{N}$. We have $(x+y)^n=C(n,0)\cdot x^n+C(n,1)\cdot x^{n-1}y+...+C(n,n-1)\cdot xy^{n-1}+C(n,n)\cdot y^n$.*

Proof. Base case. $n=0$. We have $(x+y)^0=1$. Note that when $n=0$, the right hand side is 1.

Step case. Let $n\in\mathbb{N}$, assume $(x+y)^n=C(n,0)\cdot x^n+C(n,1)\cdot x^{n-1}y+...+C(n,n-1)\cdot xy^{n-1}+C(n,n)\cdot y^n$ as inductive hypothesis (IH). We need to show $(x+y)^{n+1}=C(n+1,0)\cdot x^{n+1}+C(n+1,1)\cdot x^ny+...+C(n+1,n)\cdot xy^n+C(n+1,n+1)\cdot y^{n+1}$. One the left hand side, we have the following:

$$(x+y)^{n+1}=(x+y)^n(x+y)\stackrel{IH}{=}(C(n,0)\cdot x^n+C(n,1)\cdot x^{n-1}y+...+C(n,n-1)\cdot xy^{n-1}+C(n,n)\cdot y^n)(x+y)=$$
$$(C(n,0)\cdot x^{n+1}+C(n,1)\cdot x^ny+...+C(n,n-1)\cdot x^2y^{n-1}+C(n,n)\cdot xy^n)+(C(n,0)\cdot x^ny+C(n,1)\cdot$$
$$x^{n-1}y^2+...+C(n,n-1)\cdot xy^n+C(n,n)\cdot y^{n+1})$$

On the right hand side, we have the following.

$$C(n+1,0)\cdot x^{n+1} + C(n+1,1)\cdot x^n y + ... + C(n+1,n)\cdot xy^n + C(n+1,n+1)\cdot y^{n+1} =$$
$$C(n,0)\cdot x^{n+1} + C(n,0)\cdot x^n y + C(n,1)\cdot x^n y + ... + C(n,n-1)\cdot xy^n + C(n,n)\cdot xy^n + C(n,n)\cdot y^{n+1} =$$
$$(C(n,0)\cdot x^{n+1} + C(n,1)\cdot x^n y + ... + C(n,n-1)\cdot x^2 y^{n-1} + C(n,n)\cdot xy^n) + (C(n,0)\cdot x^n y + C(n,1)\cdot$$
$$x^{n-1}y^2 + ... + C(n,n-1)\cdot xy^n + C(n,n)\cdot y^{n+1})$$

So we prove the step case. □

8 Basic number theory

8.1 Primes and divisibility

Definition 45 (Divisibility)**.** *Let $p, n \in \mathbb{N}$. We write $n|p$ if there exist $k \in \mathbb{N}$ such that $p = kn$.*

Definition 46. *We say p is prime if there does not exist $1 < n < p$ such that $n|p$. If there exists $1 < n < p$ such that $n|p$, we say p is composite.*

Theorem 31. *Let $n, a, b \in \mathbb{N}$.*

1. *If $n|a$ and $n|b$, then $n|a+b$.*
2. *If $n|a+b$ and $n|a$, then $n|b$.*

Theorem 32. *Every natural number greater than 1 can be written uniquely as a prime or as the product of two or more primes.*

Currently, there are no known *efficient* algorithm to factor large number into products of primes.

Definition 47. *Let $a, b, n \in \mathbb{N}$.*

- *We say n is a **common divisor** of a, b if $n|a$ and $n|b$.*
- *We say n is the **greatest common divisor** of a, b if $n|a$ and $n|b$. And if m is also a common divisor, then $m|n$. We write $\mathsf{gcd}(a,b)$ to denote the greatest common divisor.*
- *We say a, b are **coprime/relatively prime** if $\mathsf{gcd}(a,b) = 1$.*
- *We say n is the **least common multiple** of a, b if $a|n$ and $b|n$. And if there exists m such that $a|m$ and $b|m$, then $n|m$. We write $\mathsf{lcm}(a,b)$ to denote the least common multiple.*

Theorem 33. *Suppose $a, b \in \mathbb{N}$. We have $ab = \mathsf{lcm}(a,b) \times \mathsf{gcd}(a,b)$.*

Note that we have $\mathsf{lcm}(0,0) = \mathsf{gcd}(0,0) = 0$.

8.2 Basic number theoretic algorithms

Definition 48 (Euclidean division algorithm)**.** *We define the following division function that return a pair of quotient and remainder* $\mathsf{divMod} : \mathbb{N} \times (\mathbb{N}/\{0\}) \to \mathbb{N} \times \mathbb{N}$.
$\mathsf{divMod}(n,m) = (0,n)$ *if* $n < m$.
$\mathsf{divMod}(n,m) = (q+1, r)$ *otherwise, where* $(q,r) = \mathsf{divMod}(n-m, m)$.

Note that since divMod is a function, this means for any $(a,b) \in \mathbb{N} \times (\mathbb{N}/\{0\})$, there exists a unique output $\mathsf{divMod}(a,b)$.

The natural number division can be extended to define integer division (making divMod a function $\mathbb{Z} \times (\mathbb{Z}/\{0\}) \to \mathbb{Z} \times \mathbb{Z}$), we leave this as an exercise.

Definition 49 (Strong Induction)**.** *To prove a statement $A[n]$ about the natural number n: We first prove that $A[0]$ holds. Then we assume $A[i]$ holds for any $0 \leq i < n \in \mathbb{N}$ (this assumption is also called inductive hypotheses), we prove that $A[n]$ holds.*

Strong induction is *strong* in the sense that the inductive hypothesis we get to assume more, i.e., $A[i]$ holds for all $0 \leq i < n$. We are going to use strong induction to prove the following Euclidean division theorem.

Theorem 34 (Euclidean division theorem)**.** *Let $a, d \in \mathbb{N}$ and $d > 0$. If $\mathsf{divMod}(a, d) = (q, r)$, then $a = dq + r$ and $0 \leq r < d$.*

Proof. We prove this by strong induction on a.

Base case: $a = 0$. In this case $\mathsf{divMod}(0, d) = (0, 0)$. Therefore $0 = d \times 0 + 0$ and $0 < d$.

Step case: Let $a \in \mathbb{N}$. We assume (as inductive hypothesis) for any $0 \leq k < a$, if $\mathsf{divMod}(k, d) = (p, s)$, then $k = dp + s$ and $0 \leq s < d$.

We need to show: If $\mathsf{divMod}(a, d) = (q, r)$, then $a = dq + r$ and $0 \leq r < d$.

Suppose $a < d$. Then $\mathsf{divMod}(a, d) = (0, a)$. Therefore $a = d \times 0 + a$ and $a < d$.

Suppose $a = d$. We have $\mathsf{divMod}(a - d, d) = \mathsf{divMod}(0, d) = (0, 0)$. Thus $\mathsf{divMod}(a, d) = (1, 0)$. So $a = d \times 1 + 0$ and $0 < d$.

Suppose $a > d$. Since $d > 0$, so $0 < a - d < a$. In this case $\mathsf{divMod}(a, d) = (u + 1, v)$ and $(u, v) = \mathsf{divMod}(a - d, d)$. By induction hypothesis, if $\mathsf{divMod}(a - d, d) = (u, v)$, then $a - d = du + v$ and $0 \leq v < d$. Thus we have $a = d(u + 1) + v$ and $0 \leq v < d$.

So by strong induction, we conclude. □

We can calculate the greatest common divisor via the following recursive function.

Definition 50.

$\mathsf{gcd} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$

$\mathsf{gcd}(a, 0) = a.$

$\mathsf{gcd}(a, b) = \mathsf{gcd}(b, r)$, *if* $b > 0$, *where* $(q, r) = \mathsf{divMod}(a, b)$.

Theorem 35 (Bézout's Identity)**.** *Let $a, b \in \mathbb{N}$. There exists $s, t \in \mathbb{Z}$ such that $s \cdot a + t \cdot b = \mathsf{gcd}(a, b)$. We call s, t Bézout coefficients.*

Note that Bézout coefficients are not unique. We can extend the gcd function in Definition 50 to also return Bézout coefficients.

Definition 51.

$\mathsf{extGCD} : \mathbb{N} \times \mathbb{N} \to \mathbb{N} \times \mathbb{Z} \times \mathbb{Z}$

$\mathsf{extGCD}(a, 0) = (a, 1, 0).$

$\mathsf{extGCD}(a, b) = (g, t, s - q \times t)$, *if* $b > 0$, *where* $(q, r) = \mathsf{divMod}(a, b)$ *and* $(g, s, t) = \mathsf{extGCD}(b, r)$.

Theorem 36. *Let $a, b \in \mathbb{N}$. If $(g, s, t) = \mathsf{extGCD}(a, b)$, then $g = s \cdot a + t \cdot b$ and $\mathsf{gcd}(a, b) = g$.*

Proof. We prove this by strong induction on b.

- Base case. $b = 0$. We have $(a, 1, 0) = \mathsf{extGCD}(a, 0)$. So $a = 1 \cdot a + 0 \cdot 0$ and $a = \mathsf{gcd}(a, 0)$.
- Step case. Suppose for any $0 \leq k < b$, if $(g, s, t) = \mathsf{extGCD}(a, k)$, then $g = s \cdot a + t \cdot k$ and $\mathsf{gcd}(a, k) = g$ for any $a \in \mathbb{N}$ (IH).

  So $\mathsf{extGCD}(a, b) = (g, t, s - q \times t)$, where $(q, r) = \mathsf{divMod}(a, b)$ and $(g, s, t) = \mathsf{extGCD}(b, r)$. Hence $a = qb + r$. Since $r < b$, by IH, we have $g = s \cdot b + t \cdot r$ and $\mathsf{gcd}(b, r) = g$.

  Thus $\mathsf{gcd}(a, b) = \mathsf{gcd}(b, r) = g$ and $t \cdot a + (s - q \times t) \cdot b = (t \times (qb + r)) + (s - q \times t) \cdot b = tqb + tr + sb - qtb = tr + sb = g$

□

8.3 Congruence and modulo arithmetic

Theorem 37. *Let $a, m \in \mathbb{Z}$ and $m > 0$. then there exists unique $q, r \in \mathbb{Z}$ such that $a = mq + r$ and $0 \leq r < m$. We write $a \equiv r \pmod{m}$ for $a = mq + r$. We write $a \bmod m$ for r.*

Theorem 38. *Suppose $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$. We have the following.*

- $a + c \equiv b + d \pmod{m}$.
- $ac \equiv bd \pmod{m}$.

Theorem 39. *Let $a, b, m \in \mathbb{Z}$ and $m > 0$. Then we have the followings.*

- $(a + b) \bmod m = ((a \bmod m) + (b \bmod m)) \bmod m$.
- $(a \cdot b) \bmod m = ((a \bmod m) \cdot (b \bmod m)) \bmod m$.

Theorem 40 (Modular inverse)**.** *Let $b, c \in \mathbb{N}$. If $\gcd(b, c) = 1$, then there exists $a \in \mathbb{N}$ such that $ab \equiv ba \equiv 1 \pmod{c}$.*

Proof. Let $b, c \in \mathbb{N}$. Suppose $\gcd(b, c) = 1$. By Bézout's identity, we have $sb + tc = 1$. Thus $a \equiv s \pmod{c}$. □

Lemma 1. *Let $a, b, c \in \mathbb{N}$. If $\gcd(a, b) = 1$ and $a|bc$, then $a|c$.*

Theorem 41. *Let p, q be primes, $b \in \mathbb{N}$ and $p \neq q$ and $x \in \mathbb{N}$. If $x \equiv b \pmod{p}$ and $x \equiv b \pmod{q}$, then $x \equiv b \pmod{pq}$.*

Proof. Since $x = k_1 p + b$ and $x = k_2 q + b$ for some $k_1, k_2 \in \mathbb{N}$, we have $k_1 p = k_2 q$. So $q|k_1$ (i.e., $k_1 = k_3 q$ for some $k_3 \in \mathbb{N}$), which means $x = k_1 p + b = k_3 qp + b$. □

Lemma 2 (Modular binomial expansion)**.** *Let p be a prime and $x, y \in \mathbb{N}$. Then $(x+y)^p \equiv x^p + y^p \pmod{p}$*

Theorem 42 (Fermat's little theorem)**.** *Let $a \in \mathbb{N}$, p be prime. We have $a^p \equiv a \pmod{p}$.*

Proof. We prove this theorem by induction on a.

- Base case: $a = 0$. In this case we have $0^p \equiv 0 \pmod{p}$.
- Step case: Let $a \in \mathbb{N}$. We assume $a^p \equiv a \pmod{p}$ as inductive hypothesis. We have $(a + 1)^p \equiv a^p + 1 \equiv a + 1 \pmod{p}$. Note that the first modular equality is by Lemma 2, and the second one is by induction.

□

Theorem 43. *Let $a \in \mathbb{N}$, p be prime and $p \not| a$. As a consequence of Fermat's little theorem, We have $a^{p-1} \equiv 1 \pmod{p}$.*

8.4 A quick tour of RSA

Definition 52 (RSA)**.**

Public key n, e	*Priviate key* p, q, d
n	$n = pq$*, where* p, q *are generated large primes.*
e	d*, where* e *is generated such that* $\mathsf{gcd}(e, (p-1)(q-1)) = 1$. d *can be calculated from* e*, i.e.,* $de \equiv 1 \bmod (p-1)(q-1)$

Encryption*: Let* m *be a number represents plaintext. We can calculate the ciphertext* $c = (m^e \bmod n)$.
Decryption*: Calculate* $c^d \bmod n$.
Protocol*: Each person has her own private key and public key, where the private key are keep private, the public key can be sent to the internet.*

If Alice want to send Bob a message m*, she just need to obtain Bob's public key* (n, e) *from the internet, and calculate* $c = \mathsf{mod}(m^e, n)$ *and send* c *to Bob.*

When Bob receives c*, he just need to fetch his private key* (n, d) *and calculates* $c^d \bmod n$.

Why RSA is secure? The security of RSA lies in given c, n, e, it is computationally hard to find out m. One way to find out m is to factor n into a product of two primes, but this is a hard problem.

Why RSA is correct? The following theorem establish the correctness of RSA.

Theorem 44. *Let* p, q *be different primes,* $n = pq$ *and* $de \equiv 1 \bmod (p-1)(q-1)$*. Then for any number* $0 \leq m < \min(p, q)$*,* $(m^e)^d \equiv m \bmod n$.

Proof. Since $de \equiv 1 \bmod (p-1)(q-1)$, we have $de = k(p-1)(q-1) + 1$ for some $k \in \mathbb{N}$. So by Fermat's little theorem, we have $(m^e)^d \equiv m^{de} \equiv m^{k(p-1)(q-1)+1} \equiv m^{k(p-1)(q-1)} \cdot m \equiv m \bmod p$ and $(m^e)^d \equiv m^{de} \equiv m^{k(p-1)(q-1)+1} \equiv m^{k(p-1)(q-1)} \cdot m \equiv m \bmod q$. By Theorem 41, we have $(m^e)^d \equiv m \bmod pq$. □