Contents

Discrete Mathematics

Frank (Peng) Fu

July 23, 2020

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: $\{0, 1, 2, 3\}$, $\{`a', `b', ..., `z'\}$, $\{0, 1\}$, $\{\}$, $\{1\}$.

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, $\{`a', \text{“}hello\text{”}, 1, \emptyset, \mathbb{N}\}$ 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, $1 \in \mathbb{N}, `a' \notin \mathbb{N}, `a' \in \{`a', `b', ..., `z'\}$.

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?

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

[1] Note that 0 is a natural number.

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

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 $\#S = \#S_1 + \#S_2 + ... + \#S_n = \underbrace{l + l + ... + l}_{n} = n \times l = \#A \times \#B$.

□

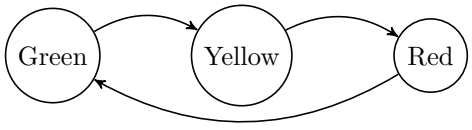
2 Basic relation theory

2.1 Introduction

Relation is a concept that is useful for characterizing the connections between different sets. Relation is widely used in computer science for describing database and state transitions.

Definition 5. *A relation R on sets A, B is a subset $R \subseteq A \times B$. We often write aRb when $(a, b) \in R$.*

It is common to talk about a relation R on set A, i.e., $R \subseteq A \times A$. What is nice about relation on A is that it also admits a diagram representation. Consider $A = \{\text{Red}, \text{Green}, \text{Yellow}\}$ and $R = \{(\text{Green}, \text{Yellow}), (\text{Yellow}, \text{Red}), (\text{Red}, \text{Green})\}$.



The above diagram describes the transition relation of a traffic light. For a relation R on a finite set A, we can represent all the members in A as nodes in a diagram, and the element $(a, b) \in R$ as an edge going from a to b.

As an exercise, please describe an elevator moving relation on the set $\{\text{L1}, \text{L2}, \text{L3}\}$.

Note that when A is an infinite set, the diagrammatic representation for $R \subseteq A \times A$ does not work anymore. In this case we have to work with the definition of the relation (Definition 5). An example of a relation on infinite set is the following, which represents the *less than* relation on natural numbers.

$$\text{less} = \{(a, b) \mid a, b \in \mathbb{N}, a < b\}$$

Definition 6 (Inverse relation)**.** *Let $R \subseteq A \times B$, we define the inverse of R as $R^{-1} = \{(b, a) \mid (a, b) \in R\} \subseteq B \times A$.*

Notice that if $R \subseteq A \times B$, then $R^{-1} \subseteq B \times A$. If R is a relation on A, i.e., $R \subseteq A \times A$, then we still have $R^{-1} \subseteq A \times A$. So when A is a finite set and R is a relation on A, the inverse of R means reversing the direction of all edges in R, while keeping the nodes unchanged.

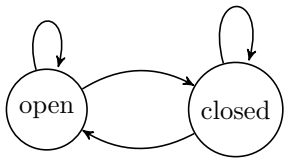
2.2 Properties of relations

Definition 7. *Let R is a relation on A.*

- *R is reflexive: for every $a \in A$, we have aRa.*
- *R is symmetric: for every $a, b \in A$, if aRb, then bRa.*
- *R is transitive: for $a, b, c \in A$, if aRb and bRc, then aRc.*

Here are some exercises you can do to see if the above definition makes sense:

1. Determine if the relations in Section 2.1 (i.e., the traffic light/elevator/less relations) are reflexive/symmetric/transitive.
2. The states of an automatic door form a set $\{\text{open}, \text{closed}\}$. The following is a relation that characterizes the behavior of the door.



   Is this relation reflexive/symmetric/transitive?

2.3 Failing preconditions

There are a few edge cases you will need to keep in mind when determining the reflexive/symmetric/transitive for a relation. They are all related to the preconditions of Definition 7 (which are all underlined). When a relation fails the preconditions in Definition 7, the corresponding property will be trivially satisfied (you will learn more about this later in the logic class[3]). Therefore, we have the following unusual examples.

- Let $A = \emptyset$. The empty set $\emptyset$ is a relation on A, and it is trivially reflexive, symmetric and transitive. This is because the empty set fails all the preconditions in Definition 7.
- Let A be an non-empty set. The empty set $\emptyset$ is a relation on A, and it is trivially symmetric and transitive, but it is not reflexive.
- The relation $\{(1, 2)\}$ (for $A = \{1, 2, 3, 4\}$) is trivially transitive, because there is no edge coming out of 2, it fails the precondition for transitivity.

It is important to note that failing precondition is not the same as failing the definition. For example $R = \{(1, 2)\}$ (for $A = \{1, 2, 3, 4\}$) is still **not** reflexive **nor** symmetric because it does not satisfy the definition of reflexivity and symmetry.

[3]Similar to how we argue that the empty set is a subset of any set.

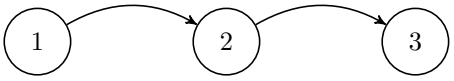
2.4 Closure operations on relations

Often a relation we consider does not have all the desirable properties (i.e., it may not be reflexive, symmetric, or transitive), in this case we can use the following closure operations to obtain a larger relation that has the properties.

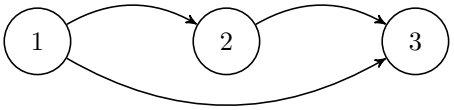
Definition 8 (Closures)**.** *Let R be a relation on A.*

- *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 the smallest transitive relation (denoted by R^+) on A that contains R. So if there is another relation S that contains R and S is transitive, then $R^+ \subseteq S$.*

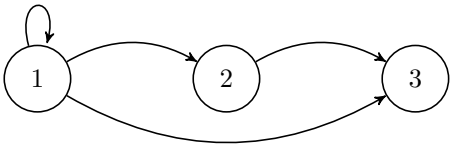
Consider the following relation $R = \{(1, 2), (2, 3)\}$ on $\{1, 2, 3\}$.



It is not a transitive relation because $(1, 3)$ is not in R. So we add $(1, 3)$ to R, obtaining the relation $R_1 = \{(1, 2), (2, 3), (1, 3)\}$, and we can see that it is transitive.



It turns out R_1 is the transitive closure of R. Because any other transitive relation that contains R would also contain R_1. For example, the following $R_2 = \{(1, 2), (2, 3), (1, 3), (1, 1)\}$ is also transitive.

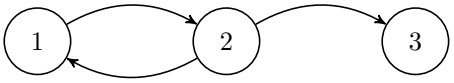


But R_2 is not the transitive closure of R because it is not the smallest transitive relation that contains R (note that $R_1 \subseteq R_2$).

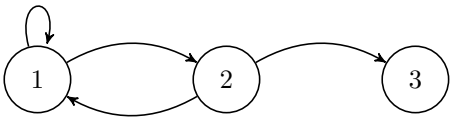
Definition 9 (Obtaining transitive closure)**.** *The following is a simple manual process to make a transitive closure from R (assuming A is a finite set).*

1. *Let $S := R$*
2. *Is S a transitive relation?*
3. *If yes, done, return S.*
4. *If no because there are $(a, b), (b, c) \in S$, but $(a, c) \notin S$.*
   Update $S := S \cup \{(a, c)\}$. Goto (2).

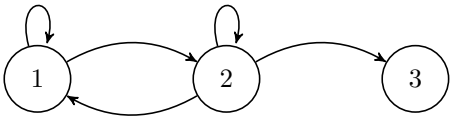
Let us now see another example. Consider the following relation $R = \{(1,2),(2,1),(2,3)\}$ on $\{1,2,3\}$.



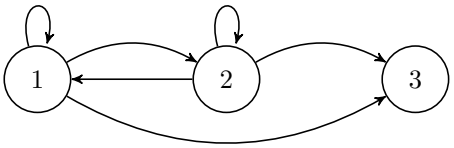
It is not transitive because $(1,2),(2,1) \in R$, but $(1,1) \notin R$. So we add $(1,1)$ and obtain the following relation R_1.



R_1 is still not transitive because $(2,1),(1,2) \in R_1$, but $(2,2) \notin R_1$. So we add $(2,2)$ to R_1 and obtain the following relation R_2.



R_2 is still not transitive because $(1,2),(2,3) \in R_2$, but $(1,3) \notin R_2$. So we add $(1,3)$ to R_2 and obtain the following relation R_3.



Now R_3 is transitive. So R_3 is the transitive closure for R.

The following is a few exercises you can do to try out Definition 9. Let $A = \{1,2,3,4\}$.

- Obtain the transitive closure of $\{(1,2),(2,3),(3,4)\}$.
- Obtain the transitive closure of $\{(1,2),(2,3),(3,4),(4,1)\}$.

Note that the method in Definition 9 does not scale well when A and R is large, for that we need more clever algorithms. So we leave it as a question for you to consider after finishing this course!

3 Functions

Why do we care about functions? In computer science, function is a useful concept to describe the computational process, a lot of computational tasks can be described as functions. 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.

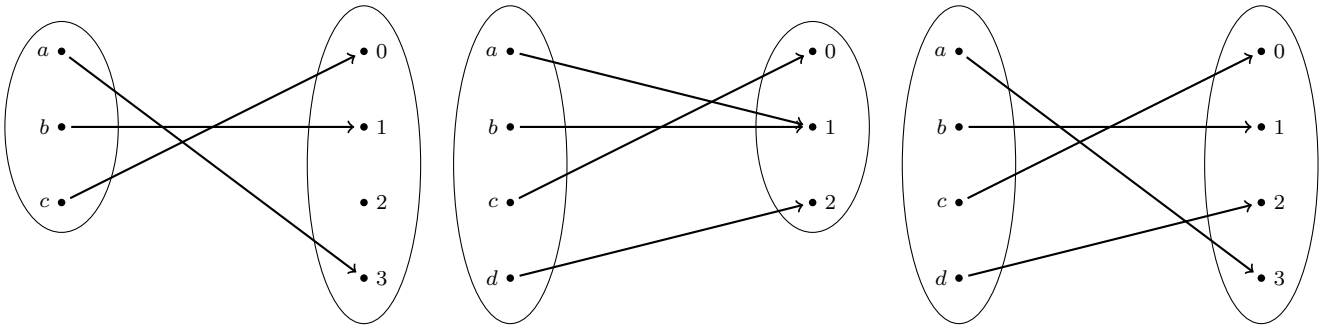


Figure 1: Examples of injective, surjective and bijective functions

Definition 10. *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: not : $\mathbb{B} \to \mathbb{B}$, where $\text{not}(0) = 1$ and $\text{not}(1) = 0$ for the set of booleans $\mathbb{B} = \{0, 1\}$.

We can compose functions together to form new functions. For example, say we have two functions not : $\mathbb{B} \to \mathbb{B}$, or : $\mathbb{B} \times \mathbb{B} \to \mathbb{B}$, then we can define a new function $\text{imply}(x, y) = \text{or}(\text{not}(x), y)$, where imply : $\mathbb{B} \times \mathbb{B} \to \mathbb{B}$.

3.1 Injective, surjective and bijective functions

There are three properties of functions that you should know.

Definition 11.

- *A function $f : A \to B$ is called injective if the following holds: for every $a, b \in A$, if $a \neq b$, then $f(a) \neq f(b)$.*

  Alternatively, if $f(a) = f(b)$ for some $a, b \in A$, then $a = b$.

- *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 of functions on finite sets at Figure 1. Note that for functions on infinite sets, we have to use logical arguments to prove whether a function is injective, surjective or bijective.

3.2 Recursive functions for natural numbers

Recursive functions are important because they are essentially descriptions of algorithms. There are a range of programming languages called *functional programming languages* where the programs are based on the idea of functions. Moreover, most mainstream programming languages also support recursion by one way or another.

A lot of concepts we encountered so far 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 12.

$\mathsf{sum} : \mathbb{N} \to \mathbb{N}$

$\mathsf{sum}(0) = 0$

$\mathsf{sum}(n) \mid \textit{if } n > 0 = n + \mathsf{sum}(n - 1)$

If you try to calculate the value of $\mathsf{sum}(5)$ for example, you should be able to convince yourself that it is equal to $0+1+2+3+4+5$.

The summation function we define above is the same thing 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 to the format in Definition 12 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 numbers of the sequence are $0, 1$, from then on, every number in the sequence is the addition of previous two numbers. We can describe these kind of sequences formally as recursive functions $\mathbb{N} \to \mathbb{N}$, where the input set denotes the position in the sequence and the output is the number at that position.

Definition 13 (Fibonacci function)**.**
$\mathsf{fib} : \mathbb{N} \to \mathbb{N}$
$\mathsf{fib}(0) = 0$
$\mathsf{fib}(1) = 1$
$\mathsf{fib}(n) \mid \textit{if } n > 1 = \mathsf{fib}(n-1) + \mathsf{fib}(n-2)$

We can convince ourselves that above definition indeed describe the fibonacci sequence by calculating the values for $\mathsf{fib}(0), \mathsf{fib}(1), \mathsf{fib}(2), \mathsf{fib}(3), ...$

Now consider the sequence $1, 2, 6, 24, 120, ...$ generated by $n!$ (the factorial of n). Can you find a recursive function for $(n!)$?

Definition 14. *In general, we use the following scheme to define recursive functions on natural numbers.*

- *Base cases:*

  Define the function from zero upto a number (e.g. $f(0) = 0, f(1) = 1$.).

- *Recursive cases:*

  Define the function for the rest of the cases.

  E.g., $f(n) = ...$ if $n > 1$, and we can use $f(n-1), f(n-2)$ and so on in the recursive cases.

Now say we have simple calculator that only supports ± 1 and recursion. Can we program an addition function?

Definition 15 (Addition)**.**
$\mathsf{add} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$
$\mathsf{add}(0, y) = y$
$\mathsf{add}(x, y) \mid \textit{if } x > 0 = 1 + \mathsf{add}(x-1, y)$

You should do a calculation for $\mathsf{add}(4, 3)$ to test the addition function. Let us look at another example, can we program a subtraction function for our calculator? Of course!

Definition 16 (Subtraction)**.**
$\mathsf{minus} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$
$\mathsf{minus}(0, y) = 0$
$\mathsf{minus}(x, 0) \mid \textit{if } x > 0 = x$
$\mathsf{minus}(x, y) \mid \textit{if } x > 0 \textit{ and } y > 0 = \mathsf{minus}(x-1, y-1)$

You should do a calculation for $\mathsf{minus}(4,3)$ to test the function. Note that the above $\mathsf{minus}(x,y)$ will return 0 if $x \leq y$, this makes sense because there is no negative number in $\mathbb{N}$!

The following are a few exercises you can do.

- Is the above minus function an injective function?
- Can you define the multiplication function mult for natural numbers using only ± 1, $\mathsf{add}, \mathsf{minus}$?
- Can you define the exponential function $\mathsf{exp}(x,y)$ for natural numbers (such that it gives x^y) using only ± 1, $\mathsf{add}, \mathsf{minus}, \mathsf{mult}$?
- Can you think of other interesting functions that you can define like above?

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 $\forall x.(x \in \mathbb{N} \wedge \mathrm{Even}(x)) \Rightarrow \neg \mathrm{Odd}(x)$. 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 $\exists x.x \in \mathbb{N} \Rightarrow \mathrm{Even}(x) \wedge \mathrm{Prime}(x)$. $\mathrm{Even}(x)$ is defined as $\exists k.k \in \mathbb{N} \Rightarrow x = 2k$. $\mathrm{Odd}(x)$ 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 6. *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 7. *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 8. *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 9. *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 10 (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 17 (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 11. *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 12. $1 + 2 + 3 + ... + n = \frac{n(n+1)}{2}$

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

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

Theorem 13. $1 + 2 + 2^2 + ... + 2^n = 2^{n+1} - 1$

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

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

4.4 Indirect proof methods

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

Theorem 14. *For every $n \in \mathbb{N}$, if n^2 is even, then n is even.*

Proof. Suppose n is not even.

By Theorem 9 and Theorem 11, we conclude that n is odd.

By Theorem 6, we have n^2 is odd.

By Theorem 9 and Theorem 11, 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 15. *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, $a \sim b \vee a \nsim b$. 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 $a \nsim b$. 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 16. *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 $\bot$.

4.5 A brief introduction to propositional logic and satisfiability

Definition 18 (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 19 (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 $\top$ by the assignment ρ.*

Definition 20. *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 21 (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 22. *We say F and G are semantically equivalent if $F \models G$ and $G \models F$.*

Theorem 17 (Deduction theorem)**.** *If $G \models F$, then $\models G \Rightarrow F$.*

Theorem 18. *F is satisfiable iff $\neg F$ is not valid.*

Definition 23 (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 19. *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 24 (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 1. *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!***	$x_1 = \mathsf{T}, x_2 = \mathsf{F}, x_3 = \mathsf{F}$	$\emptyset$

5 Counting: applied finite sets

Definition 25 (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 2.

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 26 (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 3. *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 27 (Subtraction principle)**.** *If* $S \subseteq A$*, then* $\#(A/S) = \#A - \#S$.

Example 4. *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 28 (Principle of inclusion–exclusion)**.** *Let A, B be finite sets.* $\#(A \cup B) = \#A + \#B - \#(A \cap B)$.

Example 5.

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?*

5.1 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 29. *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 30 (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 29 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 20. *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 $P(n', m+1) = P(n', m) \times (n' - m) \overset{IH}{=} \frac{n'!}{(n'-m)!} \times (n' - m) = \frac{n'!}{(n'-(m+1))!}$. □

Example 6.

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?*

5.2 Combinations: subsets of a finite set

Definition 31. *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 7. *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[4].

Theorem 21. *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

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

□

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 22. *For $n \in \mathbb{N}$ and $0 \leq r \leq n$, we have $P(n,r) = C(n,r) \times r!$. By Theorem 20, 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 $0 \leq r \leq n$, we have $C(n,r) = \frac{n!}{(n-r)! \times r!}$ as inductive hypothesis(IH). We need to prove that for any $0 \leq r' \leq n+1$, 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:

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

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

□

Theorem 23. *If $0 \leq r \leq n$, then $C(n,r) = C(n,n-r)$.*

Proof. Exercise. □

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

Theorem 24 (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 25. *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 $(x+y)^{n+1} = C(n+1,0) \cdot x^{n+1} + C(n+1,1) \cdot x^n y + ... + C(n+1,n) \cdot xy^n + C(n+1,n+1) \cdot y^{n+1}$. One the left hand side, we have the following:

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

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. □

6 Basic number theory

6.1 Divisibility and modulo arithmetic

Definition 32 (Divisibility)**.** *Let $a, b \in \mathbb{Z}$. We write $a|b$ if there exist $k \in \mathbb{Z}$ such that $b = ka$. The number a is called a factor/divisor of b.*

You should be able to prove the following theorem.

Theorem 26. *Let $n, a, b \in \mathbb{Z}$.*

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

Theorem 27 (Division theorem)**.** *Let $a, b \in \mathbb{Z}$ and $b > 0$. There exists unique $q, r \in \mathbb{Z}$ such that $a = qb + r$ and $0 \leq r < b$. The number q is called quotient, and the number r is called remainder.*

For the space reason, we will not go over the proof for division theorem at this point.

Note that the q, r are unique in the division theorem, moreover, r can only be $0 \leq r < b$. We often denoted q by $\text{div}(a, b)$, and we denote r by $\text{mod}(a, b)$.

Theorem 28. *Let $a, b, m \in \mathbb{Z}$ and $a > 0, m > 0$. We have the following.*

- *If $a|b$, then we have $mod(b, a) = 0$.*
- *If $b = ka + c$ for some $k, c \in \mathbb{Z}$, then we have $\text{mod}(b, a) = \text{mod}(c, a)$.*

Proof. Exercise. □

Definition 33. *Let $a, b, m \in \mathbb{Z}$ and $m > 0$. We say a is congruent/equivalent to b modulo m, denoted by $a \equiv b \pmod{m}$, if $\text{mod}(a, m) = \text{mod}(b, m)$.*

Recall that an **equivalence relation** is reflexive, symmetric and transitive. As an exercise, you should be able prove the following theorem, which says that the congruence modulo m is in fact an equivalence relation.

Theorem 29. *Let $m \in \mathbb{Z}$ and $m > 0$. The relation $R_m = \{(a, b) \mid a \equiv b \pmod{m}\}$ is an equivalence relation.*

Theorem 30. *Let $a, b, c, d, m \in \mathbb{Z}$ and $m > 0$. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a - c \equiv b - d \pmod{m}$.*

Proof. Suppose $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$. By the division theorem, and the definition of congruence modulo m, we have $a = k_1 m + r_1$ and $b = k_2 m + r_1$ for some $k_1, k_2, r_1 \in \mathbb{Z}$ and $0 \leq r_1 < m$. Moreover, $c = k_3 m + r_2$ and $d = k_4 m + r_2$ for some $k_3, k_4, r_2 \in \mathbb{Z}$ and $0 \leq r_2 < m$. Thus $a - c = (k_1 - k_3)m + (r_1 - r_2)$ and $b - d = (k_2 - k_4)m + (r_1 - r_2)$. Thus $\text{mod}(a - c, m) = \text{mod}(r_1 - r_2, m) = \text{mod}(b - d, m)$ by Theorem 28 (2). Hence $a - c \equiv b - d \pmod{m}$. □

Let $m \in \mathbb{Z}$ and $m > 0$. By division theorem, we know that upto equivalence modulo m, there can only be m possible remainders. We use $\mathbb{Z}_m$ to denote the set $\{0, 1, ..., m - 1\}$. We can do basic arithmetic (e.g., addition and multiplication) on these remainders.

Definition 34 (Arithmetic modulo m)**.** *Let $m \in \mathbb{Z}$ and $m > 0$. Let $a, b \in \mathbb{Z}_m$. We define the following.*

- *$a +_m b = \text{mod}(a + b, m)$. Note that $\text{mod}(a + b, m) \in \mathbb{Z}_m$.*
- *$a \times_m b = \text{mod}(a \times b, m)$. Note that $\text{mod}(a \times b, m) \in \mathbb{Z}_m$.*

The following are a few properties of arithmetic modulo m, you should be able to prove it.

Theorem 31. *Let $m \in \mathbb{Z}$ and $m > 0$. Let $a, b, c \in \mathbb{Z}_m$. We have the following.*

- *$a +_m b = b +_m a$ and $a \times_m b = b \times_m a$.*
- *$(a +_m b) +_m c = a +_m (b +_m c)$ and $(a \times_m b) \times_m c = a \times_m (b \times_m c)$.*
- *If $a > 0$, then $a +_m (m - a) = 0$. We say the $(m - a)$ is the additive inverse of a. Note that when $a = 0$, the additive inverse for 0 is 0.*
- *$a \times_m (b +_m c) = (a \times_m b) + (a \times_m c)$.*

6.2 Primes and GCD

Definition 35. *We say $p \in \mathbb{N}$ is prime if there does not exist an $n \in \mathbb{N}$ such that $1 < n < p$ and $n|p$. If there exists an $n \in \mathbb{N}$ such that $1 < n < p$ and $n|p$, we say p is composite.*

We can show that every natural number admits a *prime factorization*.

Theorem 32. *If n is a natural number greater than 1, then $n = p_1 \times ... \times p_l$ for some primes $p_1, ..., p_l$.*

Proof. We prove this by strong induction on n.

- Base case: $n = 2$. This case is true since 2 is a prime.
- Step case: Let $n > 1$. We assume for any $1 < k < n$, we have $k = p_1 \times ... \times p_l$ for some primes $p_1, ..., p_l$ as inductive hypothesis (IH). We need to show that $n = q_1 \times ... \times q_k$ for some primes $q_1, ..., q_k$. Since n can either be a prime or a composite, we only have two cases to consider.
  - When n is a prime. In this case we are done.
  - When n is a composite. In this case there exists $1 < m < n$ such that $m|n$. In another word, $n = mb$ for some $1 < b < n$. Thus by IH, $m = e_1 \times ... \times e_h$ and $b = c_1 \times ... \times c_j$ for some primes $e_1, ..., e_h, c_1, ..., c_j$. Thus $n = m \times n = e_1 \times ... \times e_h \times c_1 \times ... \times c_j$.

□

Obtaining prime factorization is of practical interest. However, currently, there are no known *efficient* algorithm to factor large number into products of primes. In fact, the famous RSA encryption system [5] is based on the difficulty of prime factorization.

We can use the so called *trial division* to obtain the prime factorization. It is based on the following theorem.

Theorem 33. *If n is a composite, then there exists a prime $p \leq \sqrt{n}$ such that $p|n$.*

Proof. Since n is a composite, we have $n = ab$, where $1 < a < n$ and $1 < b < n$. We can show that (using proof by contradiction) either $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$, since a and b can not be both greater than $\sqrt{n}$. Now by Theorem 32, we know that a (or b) are either prime or has a prime factor p such that $p|a$ (or $p|b$). So we conclude that there exists a prime $p \leq \sqrt{n}$ such that $p|n$. □

The contrapositive of the above theorem says that if a number n is not divisible by any primes $p \leq \sqrt{n}$, then n is prime. As an example, you can show that 101 is a prime.

Now we describe an algorithm for factoring a natural number n.

Definition 36. *Let $n \in \mathbb{N}$ and $n > 1$.*

1. *Let $a := n$.*
2. *$P = \{p \mid 2 \leq p \leq \sqrt{a}, p \text{ is prime}\}$.*
3. *We begin with 2, if there exists $p \in P$ such that $p|a$, then in this case we found a prime factor p and $a = pb$. Set $a := b$, then goto (2). Otherwise, it means that a itself is a prime factor, goto (4).*
4. *Return the $p_1 \times ... \times p_n$, where $p_1, ..., p_n$ are all the prime factors found.*

[5] You can learn more about RSA in the upper lever course Math 4116.

As an example, let's consider $n = 7007$. Note that $2, 3, 5$ do not divide 7007. So the smallest prime factor is 7, with $7007 = 7 \times 1001$. Similarly, the smallest prime factor of 1001 is 7, with $1001 = 7 \times 143$. Now consider 143, we just need to test if any of $\{2, ..., 11\}$ is a prime factor (since $\sqrt{143} < 12$). We found that $143 = 11 \times 13$. Therefore, we found a prime factorization $7007 = 7^2 \times 11 \times 13$.

An *partial order* is a relation that is reflexive, antisymmetric and transitive. An example of partial order is $\leq$ on $\mathbb{N}$. You can show that the relation $a|b$ on $\mathbb{N}$ is also a partial order. In the following notion of *greatest* common divisor and *least* common multiple, the words greatest/least are relative to the partial order $a|b$.

Definition 37. *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 is a common divisor of a, b and for any other common divisor m, we have $m|n$. We write $\gcd(a, b)$ to denote the greatest common divisor.*
- *We say a, b are **coprime/relatively prime** if $\gcd(a, b) = 1$.*
- *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 $\mathrm{lcm}(a, b)$ to denote the least common multiple.*

Theorem 34. *Suppose $a = p_1^{a_1} \times p_2^{a_2} \times ... \times p_n^{a_n}$ and $b = p_1^{b_1} \times p_2^{b_2} \times ... \times p_n^{b_n}$, where $p_1, ..., p_n$ are primes and pairwise distinct, and $a_1, ..., a_n, b_1, ...b_n \geq 0$. Then*

- $\gcd(a, b) = p_1^{\min(a_1, b_1)} \times p_2^{\min(a_2, b_2)} \times ... \times p_n^{\min(a_n, b_n)}$, *where* min *returns the smaller number.*
- $\mathrm{lcm}(a, b) = p_1^{\max(a_1, b_1)} \times p_2^{\max(a_2, b_2)} \times ... \times p_n^{\max(a_n, b_n)}$, *where* max *returns the larger number.*

Theorem 35. *Suppose $a, b \in \mathbb{N}$. We have $ab = \mathrm{lcm}(a, b) \times \gcd(a, b)$.*

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

6.3 Euclidean division and gcd algorithms

We have learned the division theorem (Theorem 27) in previous section. For the sake of simplicity, we will focus on natural numbers instead of integers. Now we are going to give a recursive function that calculates the quotient and remainder. Giving numbers $a, b \in \mathbb{N}$ and $b > 0$, the idea of calculating quotient and remainder of a and b is the following, we repeatedly subtract b from a, until we obtain a number r that is smaller than a, then that r is the remainder and the number of times that we perform the subtraction is the quotient.

Definition 38 (Euclidean division algorithm)**.** *We define the following division function that return a pair of quotient and remainder.*

$\mathsf{divMod} : \mathbb{N} \times (\mathbb{N} - \{0\}) \to \mathbb{N} \times \mathbb{N}$.

$\mathsf{divMod}(n, m) \mid$ *if* $n < m = (0, n)$

$\mathsf{divMod}(n, m) \mid$ *if* $n \geq m = (q + 1, r)$, *where* $(q, r) = \mathsf{divMod}(n - m, m)$.

For example, consider the following evaluation of $\mathsf{divMod}(10, 3)$:

$$\mathsf{divMod}(10, 3) = (q_1 + 1, r_1), \text{ where } (q_1, r_1) = \mathsf{divMod}(7, 3).$$
$$\mathsf{divMod}(7, 3) = (q_2 + 1, r_2), \text{ where } (q_2, r_2) = \mathsf{divMod}(4, 3).$$
$$\mathsf{divMod}(4, 3) = (q_3 + 1, r_3), \text{ where } (q_3, r_3) = \mathsf{divMod}(1, 3).$$
$$\mathsf{divMod}(1, 3) = (0, 1).$$
$$\text{So } r_1 = r_2 = r_3 = 1, \text{ and } q_1 = q_2 + 1 = q_3 + 1 + 1 = 0 + 1 + 1 = 2.$$
$$\text{Therefore } \mathsf{divMod}(10, 3) = (q_1 + 1, r_1) = (3, 1).$$

We can verify that 3 is indeed the quotient and 1 is the remainder, i.e., $10 = 3 \times 3 + 1$. The following theorem shows that the divMod function we defined indeed performs the division as expected.

Theorem 36 (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. □

Sometimes it is convenient to separate the divMod function into two separate functions, namely div function for calculating quotient, and a mod function for calculating remainder. We give the following definition.

Definition 39.

$\mathsf{div} : \mathbb{N} \times (\mathbb{N} - \{0\}) \to \mathbb{N}$
$\mathsf{div}(a, b) = 0$ *if* $a < b$.
$\mathsf{div}(a, b) = 1 + \mathsf{div}(a - b, b)$, *if* $a \geq b$.

$\mathsf{mod} : \mathbb{N} \times (\mathbb{N} - \{0\}) \to \mathbb{N}$
$\mathsf{mod}(a, b) = a$ *if* $a < b$.
$\mathsf{mod}(a, b) = \mathsf{mod}(a - b, b)$, *if* $a \geq b$.

Euclidean GCD algorithm Recall that n is the greatest common divisor of a, b if n is a common divisor of a, b and for any other common divisor m, we have $m|n$. Our method of calculating $\gcd(a, b)$ relies on prime factorization of a, b, which is not very efficient when a and b are large. So can we find another way to calculate $\gcd(a, b)$? Euclid came up with the very first efficient algorithm based on division theorem: Let $a, b \in \mathbb{N}$ and $b > 0$. By division theorem, we have $a = kb + r$ for some unique $k, r \in \mathbb{N}$ such that $0 \leq r < b$. Now to calculate $\gcd(a, b)$, it is enough to calculate $\gcd(b, r)$. Why does this make sense? Because we know that $\gcd(b, r)|b$ and $\gcd(b, r)|r$ by definition of $\gcd(b, r)$, therefore we have $\gcd(b, r)|a$. So $\gcd(b, r)$ is a common divisor of a and b. It turns out it is also the greatest common divisor for a and b (the proof is left in the exercises).

Let's look at an example, $\gcd(40, 28)$. By division theorem, we have the following.

$$40 = 1 \times 28 + 12.$$
$$28 = 2 \times 12 + 4.$$
$$12 = 3 \times 4 + 0.$$

Thus $\gcd(4, 0) = 4$. Moreover, we have $\gcd(4, 0)|0, \gcd(4, 0)|4, \gcd(4, 0)|12, \gcd(4, 0)|28, \gcd(4, 0)|40$. So we conclude $\gcd(4, 0) = \gcd(40, 28) = 4$.

Based on the above observations, we define the following recursive function for calculating $\gcd(a, b)$.

Definition 40.
$\mathsf{gcd} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$
$\mathsf{gcd}(a, 0) = a.$
$\mathsf{gcd}(a, b) \mid \textit{if } b > 0 = \mathsf{gcd}(b, \mathsf{mod}(a, b)).$

Note that since $\mathsf{gcd}(a, 0)|0$ and $\mathsf{gcd}(a, 0)|a$, so we define $\mathsf{gcd}(a, 0) = a$. As an example, the following is the evaluation process for $\mathsf{gcd}(40, 28)$.

$$\mathsf{gcd}(40, 28) = \mathsf{gcd}(28, \mathsf{mod}(40, 28)) = \mathsf{gcd}(28, 12)$$
$$= \mathsf{gcd}(12, \mathsf{mod}(28, 12)) = \mathsf{gcd}(12, 4) = \mathsf{gcd}(4, \mathsf{mod}(12, 4)) = \mathsf{gcd}(4, 0) = 4.$$

6.4 Bézout's identity and extended GCD

Definition 41 (Modular inverse)**.** *Let $m \in \mathbb{N}$ and $m > 0$, we say $b \in \mathbb{Z}_m$ is the modular inverse of $a \in \mathbb{Z}_m$ if $ab \equiv 1 \pmod{m}$.*

As an example, let us consider $m = 5$ and $a = 3$. Then the modular inverse of a is 2 because $2a = 6 \equiv 1 \pmod 5$. Note that given a number $a \in \mathbb{Z}_m$, its modular inverse need not exist. Consider $m = 4$ and $a = 2$, we can not find any number from $\mathbb{Z}_4 = \{0, 1, 2, 3\}$ that is the modular inverse of 2. So now the question is, when does the inverse exist for a modulo m? Bézout's identity provides a nice answer. In short, if $\mathsf{gcd}(a, m) = 1$ (i.e., a and m are relatively prime), then the inverse exist for a modulo m. Before we explain why, let us first take a look at what is Bézout's idenity.

Theorem 37 (Bézout's Identity)**.** *Let $a, b \in \mathbb{N}$. There exists $s, t \in \mathbb{Z}$ such that $\mathsf{gcd}(a, b) = s \cdot a + t \cdot b$. We call s, t Bézout coefficients.*

Note that Bézout coefficients are not unique and they can be negatives. For example, $\mathsf{gcd}(10, 12) = 2 = -1 \times 10 + 1 \times 12 = 11 \times 10 + (-9) \times 12$.

Now let us go back to the modular inverse question. When $\mathsf{gcd}(a, m) = 1$, by Bézout's identity, we know there exists $s, t \in \mathbb{Z}$ such that $\mathsf{gcd}(a, m) = 1 = sa + tm$, which means $1 \equiv sa \pmod m$, thus $\mathrm{mod}(s, m) \in \mathbb{Z}_m$ is the modular inverse of a. This observation tells us that if we know how to calculate Bézout coefficients, then we know how to find the modular inverse.

So how do we calculate Bézout coefficients? We will use the division theorem again. Let us consider the example $\mathsf{gcd}(40, 28)$. We have the following.

$$40 = 1 \times 28 + 12$$
$$28 = 2 \times 12 + 4$$
$$12 = 3 \times 4 + 0$$

So the $\gcd(40, 28) = \gcd(4, 0) = 4$. If we shift things around, we will have $4 = 28 - 2 \times 12$ and $12 = 40 - 1 \times 28$. Now we can substitute the equation for 12 into the equation for 4, then we get $4 = 28 - 2 \times (40 - 1 \times 28) = -2 \times 40 + 3 \times 28$. Thus Bézout's coefficients for $\gcd(40, 28)$ are -2 and 3. The above process of using substitution to obtain Bézout's coefficients is called back substitution.

Now consider the equation obtained from division theorem when we divide a by b: $a = qb + r$. As we know, $\gcd(a, b) = \gcd(b, r)$. Now suppose we know the Bézout's coefficients for $\gcd(b, r)$ is s, t, (i.e., $\gcd(b, r) = sb + tr$). Since $r = a - qb$, we then substitute r in $\gcd(a, b) = sb + tr$, we get $\gcd(a, b) = sb + t(a - qb) = ta + (s - tq)b$. This means that if we know how to calculate Bézout's coefficients for $\gcd(b, r)$, then we can obtain Bézout's coefficients for $\gcd(a, b)$. Based on this observation, we define the following recursive function that calculates Bézout's coefficients.

Definition 42 (Extended GCD)**.** *The following* $\mathsf{extGCD}(a, b)$ *calculate the gcd and the Bézout's coefficients.*
$\mathsf{extGCD} : \mathbb{N} \times \mathbb{N} \to \mathbb{N} \times \mathbb{Z} \times \mathbb{Z}$
$\mathsf{extGCD}(a, 0) = (a, 1, 0).$
$\mathsf{extGCD}(a, b) \mid \textit{if } b > 0 = (g, t, s - q \times t), \textit{ where } (q, r) = \mathsf{divMod}(a, b) \textit{ and } (g, s, t) = \mathsf{extGCD}(b, r).$

Note that $a = \mathsf{gcd}(a, 0) = 1 \times a + 0 \times 0$. Of course we could have define $\mathsf{extGCD}(a, 0) = (a, 1, k)$ for any $k \in \mathbb{Z}$, since $a = \mathsf{gcd}(a, 0) = 1 \times a + k \times 0$, this explains why Bézout's coefficients are not unique. In this note we stick with $\mathsf{extGCD}(a, 0) = (a, 1, 0)$.

Let us now look at the following evaluation process for $\mathsf{extGCD}(662, 414)$.

$$\mathsf{extGCD}(662, 414) = (g_1, t_1, s_1 - q_1 t_1), \text{ where } (q_1, r_1) = \mathsf{divMod}(662, 414) = (1, 248) \text{ and } (g_1, s_1, t_1) = \mathsf{extGCD}(414, 248).$$
$$\mathsf{extGCD}(414, 248) = (g_2, t_2, s_2 - q_2 t_2), \text{ where } (q_2, r_2) = \mathsf{divMod}(414, 248) = (1, 166) \text{ and } (g_2, s_2, t_2) = \mathsf{extGCD}(248, 166).$$
$$\mathsf{extGCD}(248, 166) = (g_3, t_3, s_3 - q_3 t_3), \text{ where } (q_3, r_3) = \mathsf{divMod}(248, 166) = (1, 82) \text{ and } (g_3, s_3, t_3) = \mathsf{extGCD}(166, 82).$$
$$\mathsf{extGCD}(166, 82) = (g_4, t_4, s_4 - q_4 t_4), \text{ where } (q_4, r_4) = \mathsf{divMod}(166, 82) = (2, 2) \text{ and } (g_4, s_4, t_4) = \mathsf{extGCD}(82, 2).$$
$$\mathsf{extGCD}(82, 2) = (g_5, t_5, s_5 - q_5 t_5), \text{ where } (q_5, r_5) = \mathsf{divMod}(82, 2) = (41, 0) \text{ and } (g_5, s_5, t_5) = \mathsf{extGCD}(2, 0) = (2, 1, 0).$$

So $g_5 = 2, s_5 = 1, t_5 = 0$. Thus $\mathsf{extGCD}(82, 2) = (g_5, t_5, s_5 - q_5 t_5) = (2, 0, 1 - 41 \times 0) = (2, 0, 1)$.
So $g_4 = 2, s_4 = 0, t_4 = 1$. Thus $\mathsf{extGCD}(166, 82) = (g_4, t_4, s_4 - q_4 t_4) = (2, 1, 0 - 2 \times 1) = (2, 1, -2)$.
So $g_3 = 2, s_3 = 1, t_3 = -2$. Thus $\mathsf{extGCD}(248, 166) = (g_3, t_3, s_3 - q_3 t_3) = (2, -2, 1 - 1 \times (-2)) = (2, -2, 3)$.
So $g_2 = 2, s_2 = -2, t_2 = 3$. Thus $\mathsf{extGCD}(414, 248) = (g_2, t_2, s_2 - q_2 t_2) = (2, 3, -2 - 1 \times 3) = (2, 3, -5)$.
So $g_1 = 2, s_1 = 3, t_1 = -5$. Thus $\mathsf{extGCD}(662, 414) = (g_1, t_1, s_1 - q_1 t_1) = (2, -5, 3 - 1 \times (-5)) = (2, -5, 8)$.
We can verify that $2 = -5 \times 662 + 8 \times 414$.

The following theorem shows that $\mathsf{extGCD}(a, b)$ is indeed calculating Bézout's coefficients.

Theorem 38. *Let $a, b \in \mathbb{N}$. If $\mathsf{extGCD}(a, b) = (g, s, t)$, 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 $\mathsf{extGCD}(a, 0) = (a, 1, 0)$. So $a = 1 \cdot a + 0 \cdot 0$ and $a = \mathsf{gcd}(a, 0)$.
- Step case. Suppose for any $0 \leq k < b$, if $\mathsf{extGCD}(a, k) = (g', s', t')$, then $g' = s' \cdot a + t' \cdot k$ and $\mathsf{gcd}(a, k) = g'$ for any $a \in \mathbb{N}$ (IH).

  Since $b > 0$, we have $\mathsf{extGCD}(a, b) = (g, t, s - q \times t)$, where $(q, r) = \mathsf{divMod}(a, b)$ and $\mathsf{extGCD}(b, r) = (g, s, t)$. We need to show $g = \mathsf{gcd}(a, b)$ and $t \cdot a + (s - q \times t) \cdot b = g$.

  Note that $a = qb + r$. Since $0 \leq r < b$, by IH, we have $g = s \cdot b + t \cdot r$ and $\mathsf{gcd}(b, r) = g$. So we have $\mathsf{gcd}(a, b) = \mathsf{gcd}(b, r) = g$ and $t \cdot a + (s - q \times t) \cdot b = (t \times (qb + r)) + (s - q \times t) \cdot b = tqb + tr + sb - qtb = tr + sb = g$.

□

6.5 Solving congruences

Consider the following equivalences:

$$x \equiv 4 \pmod 7$$
$$x \equiv 1 \pmod 6$$

Does a solution for $x \in \mathbb{Z}_{6\times 7}$ exist? In another word, can we find an element $c \in \mathbb{Z}_{6\times 7}$ such that $c \equiv 4 \pmod 7$ and $c \equiv 1 \pmod 6$? The answer for this question is yes, and the method that we use to obtain such c is called *Chinese remainder theorem*.

Theorem 39 (Chinese remainder theorem)**.** *Let $a, b, m, n \in \mathbb{N}$, $m, n > 0$ and $\gcd(m, n) = 1$. There exists a unique $x \in \mathbb{Z}_{mn}$ such that the following holds.*

$$x \equiv a \pmod{m}$$
$$x \equiv b \pmod{n}$$

Proof. Since $\gcd(m, n) = 1$, by Bézout's identity, we have $s, t \in \mathbb{Z}$ such that $sm + tn = 1$. Thus $tn \equiv 1 \pmod{m}$ and $sm \equiv 1 \pmod{n}$. Let $x = \text{mod}(smb + tna, mn) \in \mathbb{Z}_{mn}$. By division theorem, we have $smb+tna = qmn+x$ for some quotient $q \in \mathbb{Z}$. Thus $x = smb+tna-qmn$. Observe that $x \equiv tna \equiv a \pmod{m}$ and $x \equiv smb \equiv b \pmod{m}$. Therefore we find the solution for x.

We leave proving the uniquenss of x as an exercise. □

Once again, we note that computing Bézout coefficients play a crucial role in the Chinese remainder theorem. Now let us look back at the following equivalences.

$$x \equiv 4 \pmod{7}$$
$$x \equiv 1 \pmod{6}$$

Since $1 \times 7 + (-1) \times 6 = 1 = \gcd(6, 7)$, let $x = \text{mod}(1 \times 7 \times 1 + (-1) \times 6 \times 4, 42) = \text{mod}(-17, 42) = 25$. We can verify that 25 indeed satisfies the above modular equivalences.

We also have the following more general form of Chinese remainder theorem.

Theorem 40. *Let $a_1, ..., a_k, m_1, ..., m_k \in \mathbb{N}$, $m_i > 0$ for all i and $\gcd(m_i, m_j) = 1$ when $i \neq j$. There exists a unique $x \in \mathbb{Z}_{m_1...m_k}$ such that the following holds.*

$$x \equiv a_1 \pmod{m_1}$$
$$...$$
$$x \equiv a_k \pmod{m_k}$$

Proof. Here we only show the existence of x. Let $M_i = m_1 \times ... \times m_k/m_i$, i.e., M_i is a product of $m_1, ...m_k$ that excludes m_i. Therefore $\gcd(M_i, m_i) = 1$. So M_i has an modular inverse b_i such that $b_i M_i \equiv 1 \pmod{m_i}$.

Let $x = \text{mod}(b_1 M_1 a_1 + b_2 M_2 a_2 + ... + b_k M_k a_k, m_1...m_k) \in \mathbb{Z}_{m_1...m_k}$. Thus by division theorem, we have $x = b_1 M_1 a_1 + b_2 M_2 a_2 + ... + b_k M_k a_k - qm_1...m_k$. We can show that $x \equiv a_i \pmod{m_i}$ for all i. □

As an example, let us look at the following equivalences.

$$x \equiv 2 \pmod{3}$$
$$x \equiv 1 \pmod{5}$$
$$x \equiv 1 \pmod{7}$$

Let $M_1 = 5 \times 7 = 35, M_2 = 3 \times 7 = 21, M_3 = 3 \times 5 = 15$. We calculate the inverse for M_1 modulo 3, which is 2. Similarly, the inverse for M_2 modulo 5 is 1 and the inverse for M_3 modulo 7 is 1. Therefore we let $x = \text{mod}(2 \times M_1 \times 2 + 1 \times M_2 \times 1 + 1 \times M_3 \times 1, 3 \times 5 \times 7) = 71$. We can verify that this is indeed the solution for x.

6.6 Fermat's little theorem

Consider the task of calculating $\text{mod}(2^{1001}, 5)$. If we first calculate the exponentiation 2^{1001}, and then calculate the remainder, it would be too slow and inefficient. Fermat's little theorem solve this problem for us. The following is the theorem.

Theorem 41 (Fermat's little theorem)**.** *Let $a \in \mathbb{N}$, p be prime and $p \nmid a$. We have $a^{p-1} \equiv 1 \pmod{p}$.*

So how do we use Fermat's little theorem to calculate $\mathrm{mod}(2^{1001}, 5)$? Since by Fermat's little theorem, we have $2^4 \equiv 1 \pmod{p}$. Moreover, recall we have $\mathrm{mod}(a \times b, m) = \mathrm{mod}(\mathrm{mod}(a, m) \times \mathrm{mod}(b, m), m)$. By division theorem, we know that $1001 = 250 \times 4 + 1$, so $\mathrm{mod}(2^{1001}, 5) = \mathrm{mod}(2^{250\times4} \times 2, 5) = \mathrm{mod}(\mathrm{mod}(2^{250\times4}, 5) \times \mathrm{mod}(2, 5), 5) = \mathrm{mod}(2, 5) = 2$.

In general, whenever we need to calculate the exponentiation of a^b modulo some prime p, where $p \nmid a$, then we can use Fermat's little theorem to drastically simplify the calculation. Now let us prove Fermat's little theorem. it requires the following two lemmas.

Lemma 1. *Recall for the combinatorics function $C(p, i)$, where p is a prime and $0 < i < p$, and $C(p, i) = \frac{p!}{i! \times (p-i)!}$. We have $C(p, i) \equiv 0 \pmod{p}$.*

Proof. Exercise. □

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

Proof. By Binomial theorem, we have $(x+y)^p = C(p,0)x^p + C(p,1)x^{p-1}y + ... + C(p,p-1)xy^{p-1} + C(p,p)y^p$. Since $p|C(p,1), ..., p|C(p,p-1)$ (by Lemma 1), and $C(p,0) = C(p,p) = 1$, we have $(x+y)^p \equiv x^p + y^p \pmod{p}$. □

Theorem 42. *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 need to prove $(a+1)^p \equiv a + 1 \pmod{p}$. We have $(a+1)^p \equiv a^p + 1 \equiv a + 1 \pmod{p}$. Note that the first modular equivalence is by Lemma 2, and the second one is by induction.

□

Theorem 43 (Fermat's little theorem)**.** *Let $a \in \mathbb{N}$, p be prime and $p \nmid a$. We have $a^{p-1} \equiv 1 \pmod{p}$.*

Proof. By Theorem 42, we have $a^p \equiv a \pmod{p}$. Since p is a prime and $p \nmid a$, we know that $\gcd(p, a) = 1$. So a has a modular inverse, let's denote it by a^{-1}. Multiply both sides of $a^p \equiv a \pmod{p}$ with a^{-1}, we have $a^{p-1} \equiv 1 \pmod{p}$. □

6.7 A quick tour of RSA

Definition 43 (RSA)**.**

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

Encryption*: Let m be a number represents plaintext. We can calculate the ciphertext $c = (m^e \bmod n)$.*
Decryption*: Calculate $\mathsf{mod}(c^d, n)$.*
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 $\mathsf{mod}(c^d, n)$.

Why RSA is secure? What are the obvious attacks? One way to find out m is to factor n into a product of two primes, but this is a hard problem. The other way is to use a chosen plaintext m and try to calculate the modular logarithm (or discrete logarithm) d via $(m^e)^d \equiv m \pmod{n}$. But this is again hard because there is no known efficient way to solve discrete logarithm problem.

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

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

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