

# chapter 1

---

## Propositions

proposition is a claim that can only be true or false

## propositional variables

propositional variables is a symbol that denote a case. For example, p denotes ground is wet

Each propositional variable has a truth value T or F

Proposition can be **Atomic** or **Compound**

Atomic: Cannot be splitted down

Compound: Can be splitted down

## Logical Operators

Belows are Connectives:

Negation  $\neg$

Conjunction  $\wedge$

Disjunction  $\vee$

XOR  $\oplus$

Implication  $\rightarrow$

Biconditional  $\Leftrightarrow$

## Truth table

A truth table is a enum of all posiable cases and results

Example:

p	q	p $\wedge$ q
T	T	T
T	F	F
F	T	F
F	F	F

This table shows all the posiable cases and result of the proposition p  $\wedge$  q

## Negation

Means opposite:

$p = T$ , then  $\neg p = F$

## AND

Only be true when both are true, otherwise false

$p = T, q = T, p \wedge q = T$

$p = T, q = F, p \wedge q = F$

## OR

Only be false when both are false, otherwise true

$p = F, q = F, p \vee q = F$

$p = T, q = F, p \vee q = T$

## XOR

Return the result of "p and q have different truth value". Negation of Biconditional( $\Leftrightarrow$ )

$p = T, q = F, p \oplus q = T$

$p = T, q = T, p \oplus q = F$

## Biconditional

Return the result of "p and q have same truth value". Negation of XOR( $\oplus$ ). Also known as iff (if and only if)

$p = F, q = F, p \Leftrightarrow q = T$

$p = T, q = F, p \Leftrightarrow q = F$

## Implication

Only return false when case 1 is true, case 2 is false. Also known as **if ... then ...**

$p = T, q = T, p \rightarrow q = T$

$p = T, q = F, p \rightarrow q = F$

$p = F, q = T, p \rightarrow q = T$

$p = F, q = F, p \rightarrow q = T$

### Vacuous Truth

when the case 1 is false, no matter what is the truth value of case 2, the output is true, because we cannot make any conclusion when case 1 is false, so we assume the claim is true

## Converse, Contrapositive, and Inverse

From  $p \rightarrow q$  we can form new conditional statements

$q \rightarrow p$  is the converse of  $p \rightarrow q$

$\neg p \rightarrow \neg q$  is the inverse of  $p \rightarrow q$

$\neg q \rightarrow \neg p$  is the contrapositive of  $p \rightarrow q$

## Precedence of Logical Operators

Operator	Precedence
$\neg$	1
$\wedge$	2
$\vee$	3
$\rightarrow$	4
$\Leftrightarrow$	5

## translate to english

Unless means if ... not ...

–Unless I work hard, I will fail the exam

If I do not work hard then I will fail the exam

## Tautologies, Contradictions, and Contingencies

Tautologies: something always true, example  $p \vee \neg p$

Contradictions: something always false, example  $p \wedge \neg p$

Contingency: something neither Tautologies or Contradictions, example  $p \Leftrightarrow q$

## Logical Equivalences

Means the propositions have the exactly same results, denote by  $\equiv$

Law	proposition
Identity Laws	$p \wedge T \equiv p, p \vee F \equiv p$
Domination Laws	$p \vee T \equiv T, p \wedge F \equiv F$
Idempotent laws	$p \wedge p \equiv p, p \vee p \equiv p$
Double Negation Law	$\neg(\neg p) \equiv p$
Negation Laws	$p \vee \neg p \equiv T, p \wedge \neg p \equiv F$
Commutative Laws	$p \vee q \equiv q \vee p, p \wedge q \equiv q \wedge p$

Law	proposition
Associative	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r), (p \vee q) \vee r \equiv p \vee (q \vee r)$
Distributive Laws	$(p \wedge q) \vee r \equiv (p \vee r) \wedge (q \vee r), (p \vee q) \wedge r \equiv (p \wedge r) \vee (q \wedge r)$
Absorption Laws	$p \vee (p \wedge q) \equiv p, p \wedge (p \vee q) \equiv p$
De Morgan's Laws	$\neg(p \vee q) \equiv \neg p \wedge \neg q, \neg(p \wedge q) \equiv \neg p \vee \neg q$
Nameless	$p \rightarrow q \equiv \neg p \vee q$
Nameless	$p \Leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

## Propositional Satisfiability

A compound proposition is satisfiable if there exist at least one set of variables that can make its truth value of this proposition True.

It is unsatisfiable if there does not exist any set of variables that can make its truth value True.

A compound proposition is unsatisfiable if and only if its negation is a tautology

## Arguments

An argument is a sequence of statements (premises) that end with a conclusion

(balabala...then balabala, so balabala)

an argument is valid if and only if it is impossible for all the premises to be true and the conclusion to be false

(it is valid, so it cannot have something like all premises are True but the conclusion is False, conclusion must be True)

However, we cannot say all premises are True if the conclusion is True

An argument which is not valid is called a fallacy

## Test for Argument Validity

P1: If John eats peanuts, he falls sick

P2: John did not eat peanuts

$\therefore$  John did not fall sick

Does  $((p \rightarrow q) \wedge \neg p) \rightarrow \neg q$ ?

<b>p</b>	<b>q</b>	<b><math>((p \rightarrow q) \wedge \neg p)</math></b>	<b><math>\neg q</math></b>
F	T	T	F

premises is True  $((p \rightarrow q) \wedge \neg p)$ , but conclusion is False  $(\neg q)$ , so this is a fallacy

## Argument Forms

P1: If John eats peanuts, he falls sick

P2: John did not fall sick

---

$\therefore$  John did not eat peanuts

P1: If you work hard, you will get a good raise

P2: You did not get a good raise

---

$\therefore$  You did not work hard

Even those two arguments are different, they have the same form which is  $((p \rightarrow q) \wedge \neg q) \rightarrow \neg p$

When two arguments have the same argument form, they:

1. Have the same symbolic representation
2. Belong to the same "family" of arguments

## Important Rules of Inference

Rules of Inference	Tautology	Name
<p>p</p> <p><math>p \rightarrow q</math></p> <p>-----</p> <p><math>\therefore q</math></p>	<p><math>(p \wedge (p \rightarrow q)) \rightarrow q</math></p>	Modus ponens
<p><math>\neg q</math></p> <p><math>p \rightarrow q</math></p> <p>-----</p> <p><math>\therefore \neg p</math></p>	<p><math>(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p</math></p>	Modus tollens
<p><math>p \rightarrow q</math></p> <p><math>q \rightarrow r</math></p> <p>-----</p> <p><math>\therefore r</math></p>	<p><math>((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)</math></p>	Hypothetical Syllogism
<p><math>p \vee q</math></p> <p><math>\neg p</math></p> <p>-----</p> <p><math>\therefore q</math></p>	<p><math>(\neg p \wedge (p \vee q)) \rightarrow q</math></p>	Disjunctive Syllogism
<p>p</p> <p>-----</p> <p><math>\therefore p \vee q</math></p>	<p><math>p \rightarrow (p \vee q)</math></p>	Addition

Rules of Inference	Tautology	Name
$p \wedge q$ $\text{-----}$ $\therefore p$	$(p \vee q) \rightarrow p$	Simplification
$p$ $q$ $\text{-----}$ $\therefore p \wedge q$	$((p) \wedge (q)) \rightarrow p \wedge q$	Conjunction
$p \vee q$ $\neg p \vee r$ $\text{-----}$ $\therefore q \vee r$	$((\neg p \vee r) \wedge (p \vee q)) \rightarrow q \vee r$	Resolution

## Comparing the Tautologies for Logical Equivalences and Rules of Inference

### Logical Equivalences

Logical\_Expression1  $\Leftrightarrow$  Logical\_Expresion2 is a tautology

### Rules of Inference

Conjunction\_of\_premises  $\rightarrow$  Conclusion is a tautology

## Fallacies

### 1. Fallacy of affirming the conclusion

$p \rightarrow q$

$q$

$\text{-----}$

$\therefore p$

cannot get premises from conclusion

### 2. Fallacy of denying the hypothesis

$p \rightarrow q$

$\neg p$

$\text{-----}$

$\therefore \neg q$

Vacuous truth

## Predicate Logic

Propositional calculus is inadequate to deal with arguments that deal with all cases, or with some case out of many cases

In such instances, instead of looking at propositions as a whole, we need to understand their inner structure by

- breaking propositions up into parts {predicates and variables}
- analyzing quantifiers like "all" or "some"

## What is Predicate Logic

1. Include both relationship (and, not, or...) and quantity (one, at least one, all...)
2. Studies the logical form INSIDE atomic propositions
3. Also known as predicate calculus and first-order logic

## Predicate

Describe the properties of an object or the relationship between objects

EX: In the proposition "17 is a prime number", "is a prime number" is the predicate

## Propositional Function

A proposition contains one or more predicates but does not focus on specific object

EX:

Proposition: James is a student at Bedford College

Predicate: is a student at

Predicate variables: x, y

Propositional Function: x is a student at y {represented as  $P(x, y)$ }

Propositional Function does not have truth value by it own

propositional function becomes a proposition when it is filled with variables

The set of all possible values is called the Domain, which denoted by U

The statement  $P(x)$  is said to be the value of the propositional function P at x  
( $P(x)$  is only one case out of all possible results)

## Compound Expressions

Connectives from propositional logic carry over to predicate logic

Expressions with variables are not propositions and therefore do not have truth values. For example:

$P(x) \rightarrow P(y)$

When used with quantifiers (at least one, all), these expressions (propositional functions) become propositions

Propositional functions can have multiple variables

## Quantifiers

The "degree" of "Being True" of a propositional function over a range of elements (from the pertinent domain)

There are two types of them:

1. Universal Quantifier, "For all,"  $\forall$
2. Existential Quantifier, "There exists,"  $\exists$

## Uniqueness Quantifier

$\exists! x P(x)$  means that  $P(x)$  is true for one and only one  $x$  in the universe of discourse

"There is a unique  $x$  such that..."

## Trailing Quantifiers

Additional information at the end of the sentence

Example:  $\forall x \in \mathbb{R}, x^2 \geq 0$  ( $x$  is a real number and  $x^2$  is not 0)

this will include all real number but 0

## Bound and Free Variables

when an variable has something to do with the quantifier, we say it is bound, otherwise, we say it is free

Example:  $\exists x (x + y = 1)$ , in this case,  $x$  is bound and  $y$  is free

## quantifier, propositional function, and proposition

Quantifiers provide an alternate way to convert propositional functions to propositions

Example: Let  $P(x)$  be the propositional function " $x + 1 > x$ ". Then  $\forall x P(x)$  is a proposition

## ways to make propositional function a proposition

In general, all the variables that occur in a propositional function must be bound or set equal to a particular value to turn it into a proposition

- universal quantifiers
- existential quantifiers
- value assignments

## Properties of Quantifiers

The truth value of  $\forall x P(x)$  and  $\exists x P(x)$  depend on both the propositional function  $P(x)$  and on the domain  $U$

If the domain is empty, for any propositional function:

- $\forall x P(x)$  is true
- $\exists x P(x)$  is false



## Truth Set

A truth set is a set contains all values of  $x$  that make the proposition true

Example,  $P(x)$ : "x is a factor of 8", Domain: all positive integers

Then the truth set of  $P(x)$  is  $\{1, 2, 4, 8\}$

## Quantifiers Over Finite Domains

$$\forall x P(x) \equiv P(x_1) \wedge P(x_2) \wedge P(x_3) \wedge \dots \wedge P(x_n)$$

Conjunction of propositions

$$\exists x P(x) \equiv P(x_1) \vee P(x_2) \vee P(x_3) \vee \dots \vee P(x_n)$$

Disjunction of propositions

## Quantifiers with Restricted Domains

Sometimes it is not feasible to enumerate the domain of a quantifier. Then we use the excluding method to exclude those we don't want out

In such instances, an abbreviated notation is often used

- a condition a variable must satisfy is included after the quantifier
- Such quantifiers are called restricted quantifiers

Example:

$$(\forall x)_{x < 0} (x^2 > 0)$$

means for all  $x$  such that  $x$  smaller than 0, we have  $x$  to the power of 2 is greater than 0

## Precedence of Quantifiers

The  $\forall$  and  $\exists$  have higher precedence than any logical operator

which means  $\forall x P(x) \vee Q(x)$  means  $(\forall x P(x)) \vee Q(x)$  instead of  $\forall x (P(x) \vee Q(x))$

## Equivalences in Predicate Logic

have the save value for every possible result

cover every domain

Example:

$$\forall x \neg \neg S(x) \equiv \forall x S(x)$$

$$\forall x (P(x) \wedge Q(x)) \equiv \forall x P(x) \wedge \forall x Q(x)$$

Assume  $\forall x(P(x) \wedge Q(x))$  is True, then  $P(x) \wedge Q(x)$  is true for every value of  $x$  in domain, so  $P(x)$  and  $Q(x)$  are True for every value in domain

Assume  $\forall x P(x) \wedge \forall x Q(x)$  is True, then  $P(x)$  and  $Q(x)$  are True for every value of  $x$  in domain

So they are equivalent

## Some equivalences and not equivalences

1.  $\forall x \neg \neg S(x) \equiv \forall x S(x)$
2.  $\neg \forall x P(x) \equiv \exists x \neg P(x)$
3.  $\neg \exists x P(x) \equiv \forall x \neg P(x)$
4.  $\forall x(P(x) \wedge Q(x)) \equiv \forall x P(x) \wedge \forall x Q(x)$
5.  $\exists x(P(x) \vee Q(x)) \equiv \exists x P(x) \vee \exists x Q(x)$
6.  $\forall x(P(x) \vee Q(x)) \not\equiv \forall x P(x) \vee \forall x Q(x)$
7.  $\exists x(P(x) \wedge Q(x)) \not\equiv \exists x P(x) \wedge \exists x Q(x)$

The Universal quantifier cannot be distributed over disjunction, and the existential quantifier cannot be distributed over conjunction

## De Morgan's Laws for Quantifiers

Negation	Equivalent Statement	When Is Negation True?	When Is False
$\neg \exists x P(x)$	$\forall x \neg P(x)$	For every $x$ , $P(x)$ is false	There is $x$ which $P(x)$ is true
$\neg \forall x P(x)$	$\exists x \neg P(x)$	There is $x$ which $P(x)$ is false	$P(x)$ is true for every $x$

## Restricted Quantifiers

Unrestricted quantifiers apply to the entire domain of discourse

A restricted quantifier has the same semantics as an unrestricted quantifier except that the variables in the domain must satisfy a certain condition in order for the quantification to apply

(Which means we only focus on the values in the domain which satisfies the restriction)

Example:

$$(\forall x)_{x < 0} (x^2 > 0)$$

for every  $x$  in real numbers, if  $x$  is smaller than 0, then we have  $x$  to the power of 2 is larger than 0

## Restricted quantifier -> Unrestricted quantifier

Sometimes we need to express a restricted quantifier as an unrestricted quantifier

$$\exists x_{P(x)} Q(x) \equiv \exists x (P(x) \wedge Q(x))$$

$$\forall x_{P(x)} Q(x) \equiv \forall x (P(x) \rightarrow Q(x))$$

## Nested Quantifiers

In nested quantifiers one quantifier is within the scope of another quantifier

Example: Every real number has an inverse

$$\forall x \exists y (x + y = 0)$$

## Order of Quantifiers

The order of the nested quantifiers is important, unless all the quantifiers are universal quantifiers or all are existential quantifiers

## Quantifications of Two Variables

Statement	When True?	When False?
$\forall x \forall y P(x, y)$	$P(x, y)$ is true for every pair of $x$ and $y$	when any pair of $x, y$ make $P(x, y)$ false
$\forall x \exists y P(x, y)$	For every $x$ , there is an $y$ such that $P(x, y)$ is true	when there is a $x$ for every $y$ $P(x, y)$ is false
$\exists x \forall y P(x, y)$	There is at least one $x$ such that for every $y$ , $P(x, y)$ is true	for every $x$ , there is a $y$ such that $P(x, y)$ is false
$\exists x \exists y P(x, y)$	There is a pair of $x$ and $y$ such that	for every pair of $x$ and $y$ , $P(x, y)$ is false

## Two Surprising Results

if  $\exists y \forall x P(x, y)$  is true, then  $\forall x \exists y P(x, y)$  must be also true

if  $\forall x \exists y P(x, y)$  is true, it is not necessary for  $\exists y \forall x P(x, y)$  to be true

## Additional Rules of Inference for Quantified Statements

Rule of Inference	Name
$\forall x P(x)$ ----- $\therefore P(c)$	Universal instantiation

Rule of Inference	Name
$P(c)$ for an arbitrary of $c$ ----- $\therefore \forall x P(x)$	Universal generalization
$\exists x P(x)$ ----- $\therefore P(c)$ for some element $c$	Existential instantiation
$P(c)$ for some element $c$ ----- $\therefore \exists x P(x)$	Existential generalization

$x$ : Unspecified member of the domain

$c$ : Specific member of the domain

## Using Rules of Inference

"All men has two legs", "John is a man", shows John has two legs

let  $M(x)$  represents "x is a man", and  $L(x)$  represents "x has two legs"

premises:  $\forall x (M(x) \rightarrow L(x))$ ,  $M(\text{John})$

$\forall x (M(x) \rightarrow L(x))$   
 -----  
 $\therefore M(\text{John}) \rightarrow L(\text{John})$

$M(\text{John}) \rightarrow L(\text{John})$   
 $M(\text{John})$   
 -----  
 $\therefore L(\text{John})$

## Universal Modus Ponens

Combines UI and Modus Ponens

$\forall x (P(x) \rightarrow Q(x))$   
 $P(c)$   
 -----  
 $\therefore Q(c)$

## Universal Modus Tollens

$\forall x (P(x) \rightarrow Q(x))$   
 $\neg Q(c)$   
 -----  
 $\therefore \neg P(c)$

## Proofs of Mathematical Statements

A proof is a valid argument that establishes the truth of a statement

A theorem is a statement that can be shown to be true using:

- definitions
- other theorems
- axioms (statements which are given as true, also known as postulates)
- rules of inference

A lemma is a 'helping theorem' or a result which is needed to prove a theorem

A corollary is a result which follows directly from a theorem. Less important theorems are sometimes called propositions

A conjecture is a statement that is being proposed to be true. Once a proof of a conjecture is found, it becomes a theorem. It may turn out to be false

## Direct Proof

A direct proof shows that a conditional statement  $p \rightarrow q$  is true by showing that if  $p$  is true, then  $q$  must also be true

We assume that  $p$  is true and use axioms, definitions, and previously proven theorems, together with rules of inference, to show that  $q$  must also be true

Example:

Proof the square of an odd number is also an odd number

any odd number :  $2n+1$

$$(2n + 1)^2 = 4n^2 + 4n + 1 = 2 * (n^2 + 2n) + 1$$

let  $r = (n^2 + 2n)$ , therefore  $(2n + 1)^2 = 2r + 1$ , and because of  $2r$  is an even number, so  $2r + 1$  is an odd number

## Proof By Contraposition

Sometimes it is easier to prove theorems using proof by contraposition

- the conditional statement  $p \rightarrow q$  is proved by showing that its contrapositive,  $\neg q \rightarrow \neg p$  is true

Example: Prove that if  $n$  is an integer and  $3n + 2$  is odd, then  $n$  is odd

let  $n = 2k$  for some integer  $k$

$$\text{the } 3n + 2 = 6k + 2 = 2(3k + 1)$$

let  $j = 3k + 1$  for some integer  $j$

$$\text{so } 3n + 2 = 6k + 2 = 2j$$

so  $3n + 2$  is even

Since we have shown  $\neg q \rightarrow \neg p$ ,  $p \rightarrow q$  must hold as well. We have proved by contraposition that if  $3n + 2$  is odd, then  $n$  is odd

## Vacuous & Trivial Proofs

- $p \rightarrow q \equiv \neg p \vee q$
- A proof that makes use of the fact that  $p \rightarrow q$  must be true when  $p$  is false is called a vacuous proof
- A proof that makes use of the fact that  $p \rightarrow q$  must be true when  $q$  is true is called a trivial proof
- These proofs are never treated as complete proofs but they are used in conjunction with other proof techniques (like proof by cases and mathematical induction) to establish that special cases of a theorem are not in violation of the generalized theorem

## Proof by Contradiction

We make a opposite proof of if  $p$  is true, then  $q$  is false for  $p \rightarrow q$ . Clearly, therefore, the assumption that if  $p$  is true then  $q$  is false is wrong. In other words, if  $p$  is true then  $q$  must be true

Example:

Prove that if you pick 22 days from the calendar, at least 4 must fall on the same day of the week

Assume that no more than 3 of the 22 days fall on the same day of the week. Because there are 7 days of the week, we could only have picked 21 days. This contradicts the assumption that we have picked 22 days

## Background Information

- Fundamental theorem of arithmetic (also called the unique factorization theorem)

Every number is a prime or a unique product of primes

## Theorems that are Biconditional Statements

To prove a theorem that is a biconditional statement, that is, a statement of the form  $p \Leftrightarrow q$ , we show that  $p \rightarrow q$  and  $q \rightarrow p$  are both true

Example:

Prove the theorem: "If  $n$  is an integer, then  $n$  is odd if and only if  $n^2$  is odd."

Solution:

We have already shown (previous slides) that both  $p \rightarrow q$  and  $q \rightarrow p$ . Therefore we can conclude  $p \Leftrightarrow q$

## Proof By Counterexample

Statements of the form  $\forall x P(x)$  can be proved to be false by providing a counterexample  $\exists x (\neg P(x))$

Example: Show that the statement "Every positive integer is the sum of the squares of two integers" is false

The number 3 can be represented as the sum of two numbers in one of two ways (order being immaterial)

$$3 = 0 + 3 = 0^2 + 3$$

$$3 = 1 + 2 = 1^2 + 2$$

In neither case can be number 3 be represented as a sum of two squares. Therefore we have proved by counterexample that "Every positive integer is the sum of the squares of two integers" is false

## Mistakes in proofs

- performing a disallowed mathematical operation (divide by 0)
- given  $p \rightarrow q$  is true and  $q$  is true implying the conclusion that  $p$  is true ( $q$  can be True when  $p$  is False)
- given  $p \rightarrow q$  is true and  $p$  is false implying the conclusion that  $q$  is false ( $q$  is true)
- basing one or more steps of the proof on the truth of the statement being proved, or a statement equivalent to it (begging the question or circular reasoning)

## What is wrong with this

"Proof" that  $1 = 2$

1.  $a = b$
2.  $a^2 = a * b$
3.  $a^2 - b^2 = a * b - b^2$
4.  $(a - b)(a + b) = b(a - b)$
5.  $a + b = b$
6.  $2b = b$
7.  $2 = 1$

Solution: step 5, divide both side by  $(a - b)$ ,  $(a - b) = 0$ , it is dividing both side by 0 which is undefined

## If direct methods of proof do not work

- We may need a clever use of a proof by contraposition
- Or a proof by contradiction

## Proof by Exhaustion

Used in situations where theorems can be proved by examining a relatively small number of examples

(list all cases and proof they are true)

Example:

Prove that  $(n + 1)^3 \geq 3n$  if  $n$  is a positive integer with  $n \leq 4$

solution:

Case  $n = 1$ :  $(1 + 1)^3 = 2^3 = 8$ , which is  $\geq 3 \cdot 1$  i.e. 3

Case  $n = 2$ :  $(2 + 1)^3 = 3^3 = 27$ , which is  $\geq 3 \cdot 2$  i.e. 9

Case  $n = 3$ :  $(3 + 1)^3 = 4^3 = 64$ , which is  $\geq 3 \cdot 3$  i.e. 27

Case  $n = 4$ :  $(4 + 1)^3 = 5^3 = 125$ , which is  $\geq 3 \cdot 4$  i.e. 81

## Proof by Cases

Used in situations where:

1. it is impossible to list all the cases
2. need to consider different cases in different ways separately

To prove a conditional statement of the form

Example:

$$(p_1 \vee p_2 \vee p_3 \dots \vee p_n) \rightarrow q$$

$$(p_1 \vee p_2 \vee p_3 \dots \vee p_n) \rightarrow q \equiv \neg(p_1 \vee p_2 \vee p_3 \dots \vee p_n) \vee q$$

$$\neg(p_1 \vee p_2 \vee p_3 \dots \vee p_n) \vee q \equiv (\neg p_1 \wedge \neg p_2 \wedge \neg p_3 \dots \wedge \neg p_n) \vee q$$

$$(\neg p_1 \wedge \neg p_2 \wedge \neg p_3 \dots \wedge \neg p_n) \vee q \equiv (\neg p_1 \vee q) \wedge (\neg p_2 \vee q) \wedge (\neg p_3 \vee q) \dots \wedge (\neg p_n \vee q)$$

$$(\neg p_1 \vee q) \wedge (\neg p_2 \vee q) \wedge (\neg p_3 \vee q) \dots \wedge (\neg p_n \vee q) \equiv (p_1 \rightarrow q) \wedge (p_2 \rightarrow q) \wedge (p_3 \rightarrow q) \dots \wedge (p_n \rightarrow q)$$

Use the tautology

$$(p_1 \vee p_2 \vee p_3 \dots \vee p_n) \rightarrow q \Leftrightarrow (p_1 \rightarrow q) \wedge (p_2 \rightarrow q) \wedge (p_3 \rightarrow q) \dots \wedge (p_n \rightarrow q)$$

Each of the implications  $p_n \rightarrow q$  is a case

Example:

Prove that Prove that if  $n$  is an integer, then  $n^2 \geq n$

Case 1 :  $n = 0$ , then  $n^2 = 0$ , which  $n^2 \geq n$

Case 2 :  $n \leq 1$ , then  $n^2 \geq n$  or  $n^2 = n = 1$

Case 3:  $n \geq 1$ , then  $n^2 > n$

Since the inequality  $n^2 \geq n$  holds in all three cases, we can conclude that if  $n$  is an integer, then  $n^2 \geq n$

## Without Loss of Generality

The phrase "Without Loss of Generality" (WLOG) is used in proofs with cases to assert that by proving one case of a theorem, no additional argument is required to prove other specified cases



( Standard english: This is a common sense so I dont want to prove it)

## Existence Proofs

A proof of a proposition of the form  $\exists x P(x)$  is called an existence proof

there are two form of proof for existence proof

- Constructive: Finding an element  $a$ , which  $P(a)$  is True
- Nonconstructive: Proof there is  $\exists x P(x)$  is True without finding an element. For example, using proof by contradiction to shows that not all  $P(a)$  are False

Example 1:

Show that there is a positive integer that can be written as the sum of cubes of positive integers in two different ways

$$1729 = 10^3 + 9^3 = 12^3 + 1^3$$

Example 2:

Show that there exist irrational numbers  $x$  and  $y$  such that  $xy$  is rational

We know that  $\sqrt{2}$  is irrational. Let us consider the number  $\sqrt{2}^{\sqrt{2}}$ . There are two possibilities to consider

Possibility 1:  $\sqrt{2}^{\sqrt{2}}$  is rational

- In this case  $x = y = \sqrt{2}$  and  $x^y$  is rational

Possibility 2:  $\sqrt{2}^{\sqrt{2}}$  is irrational

- In this case let  $x = \sqrt{2}^{\sqrt{2}}$  and  $y = \sqrt{2}$ . Therefore  $x^y = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = (\sqrt{2})^2 = 2$ , a rational number

This is an example of Nonconstructive proof, we don't know the property of the  $\sqrt{2}^{\sqrt{2}}$ , but but we have proved that one of them does has the desired property

## Uniqueness Proofs

Some theorems asset the existence of a unique element with a particular property,  $\exists! x P(x)$ . The two parts of a uniqueness proof are

- Existence:  $x$  exist such that  $P(x)$  satisfies the properties
- Uniqueness: if  $y \neq x$ , then  $P(y) \neq P(x)$

Example:

Shows that if  $a, b$  are real numbers, and  $a \neq 0$ , there exits an real number  $r$  such that  $ar + b = 0$

Solution:

$$ar + b = 0$$

$$ar = -b$$

$$r = -\frac{b}{a}$$

Existence:

the real number  $r$  has a solution because  $ar + b = -b + b = 0$ , there for  $r = -\frac{b}{a}$

Uniqueness:

Suppose that  $s$  is a real number such that  $as + b = 0$

$$as + b = ar + b$$

$$as = ar$$

$$s = r$$

therefore,  $r$  is unique

QED

## Proof Strategies for proving $p \rightarrow q$

Choose a method

1. First try a direct method of proof
2. If this does not work, try an indirect method (e.g., try to prove the contrapositive)

For whichever method you are trying, choose a strategy

1. First try forward reasoning. Start with the axioms and known theorems and construct a sequence of steps that end in the conclusion. Start with  $p$  and prove  $q$ , or start with  $\neg q$  and prove  $\neg p$
2. If this doesn't work, try backward reasoning

## Backward Reasoning

1. we assume the conclusion is correct, then we try to get conditions from it
2. use the conditions to get backward, until we reach the start point
3. Start reversing, follow the opposite direction and make the forward proof

(we need to proof the reliability of those conditions before moves on)

Example:

Question:

Prove that for any two distinct positive real numbers  $x$  and  $y$ , their arithmetic mean [i.e.  $(x + y)/2$ ] is greater than their geometric mean [i.e.  $\sqrt{xy}$ ]

Solution:

We start by using backward reasoning to establish the starting point of our proof

Let us assume the conclusion is true

1.  $(x + y)/2 > \sqrt{xy}$
2.  $((x + y)^2)/4 > xy$  (sqr both sides)
3.  $(x + y)^2 > 4xy$
4.  $x^2 + 2xy + y^2 > 4xy$
5.  $x^2 - 2xy + y^2 > 0$
6.  $(x - y)^2 > 0$

Now that we have our secondary conclusion, let us see if we can prove it.

$(x - y)^2 > 0$  is true because  $x \neq y$ , so  $x - y \neq 0$ , and for any non-zero number, the square of it is always positive

Now we move to our formal proof

$$(x - y)^2 > 0$$

$$x^2 - 2xy + y^2 > 0$$

$$x^2 + 2xy + y^2 > 4xy$$

$$(x + y)^2 > 4xy$$

$$(x + y)^2/4 > xy$$

$$(x + y)/2 > \sqrt{xy}$$

## The Role of Open Problems

---

Unsolved problems have motivated much work in mathematics. Fermat's Last Theorem was conjectured more than 300 years ago. It has only recently been finally solved

Fermat's Last Theorem: The equation  $x^n + y^n = z^n$  has no solutions in integers  $x$ ,  $y$ , and  $z$ , with  $xyz \neq 0$  whenever  $n$  is an integer with  $n > 2$

A proof was found by Andrew Wiles in the 1990s

---

$3N + 1$  problem:

...

---

## Additional Proof Methods

Time permitting, we will see many other proof methods:

- Mathematical induction, which is a useful method for proving statements of the form  $\forall n P(n)$ , where the domain consists of all positive integers
- Structural induction, which can be used to prove such results about recursively defined sets
- Cantor diagonalization is used to prove results about the size of infinite sets
- Combinatorial proofs use counting arguments

## chapter 2

---

### set

A set is an unordered collection of objects

The objects in a set are called the elements, or members of the set. A set is said to contain its elements

The notation  $a \in A$  denotes that  $a$  is an element of the set  $A$ . If  $a$  is not a member of  $A$ , write  $a \notin A$

By convention, sets are denoted using uppercase letters while lowercase letters are used to denote elements of sets

elements in a set are unique

(just like Python dictionary with only keys)

### Describing a Set

#### Roster Method

---

$S = \{a, b, c, d\}$

(order does not matter)

Each distinct object is either a member or not; listing more than once does not change the set

$S = \{a, b, c, d\} = \{a, b, c, b, c, d\}$

Elipses (...) may be used to describe a set without listing all of the members when the pattern is clear

$S = \{a, b, c, d, \dots, z\}$

(iterator)

---

### Some Important Sets

name	meaning	Set
N	natural numbers	$\{0, 1, 2, 3, \dots\}$

---

name	meaning	Set
$\mathbb{Z}$	integers	$\{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$
$\mathbb{Z}^+$	positive integers	$\{1, 2, 3, \dots\}$
$\mathbb{R}$	set of real numbers	
$\mathbb{R}^+$	set of positive real numbers	
$\mathbb{C}$	set of complex numbers	
$\mathbb{Q}$	set of rational numbers	

## Set-Builder Notation

Specify the property or properties that all members must satisfy The general form of this notation is  $\{x \mid x \text{ has property } P\}$  and is read "the set of all  $x$  such that  $x$  has property  $P$ "

$S = \{x \mid x \text{ is a positive integer less than } 100\}$

A predicate may be used:  $S = \{x \mid P(x)\}$

Positive rational numbers:  $\mathbb{Q}^+ = \{x \in \mathbb{R} \mid x = p/q, \text{ for some positive integers } p, q\}$

(just like python list generator:  $[x \text{ for } x \text{ in range}(0, 100000)] \text{ if Prime}(x)$ )

## Interval Notation

"[" and "]" means including boundary, "(" and ")" means excluding boundary

closed interval  $[a, b]$

open interval  $(a, b)$

$[a, b] = \{x \mid a \leq x \leq b\}$

$(a, b) = \{x \mid a < x < b\}$

## Universal Set, Empty Set and Singleton Set

The universal set  $U$  is the set containing everything currently under consideration

The empty set (aka null set) is the set with no elements. Symbolized  $\emptyset$ , but  $\{\}$  also used

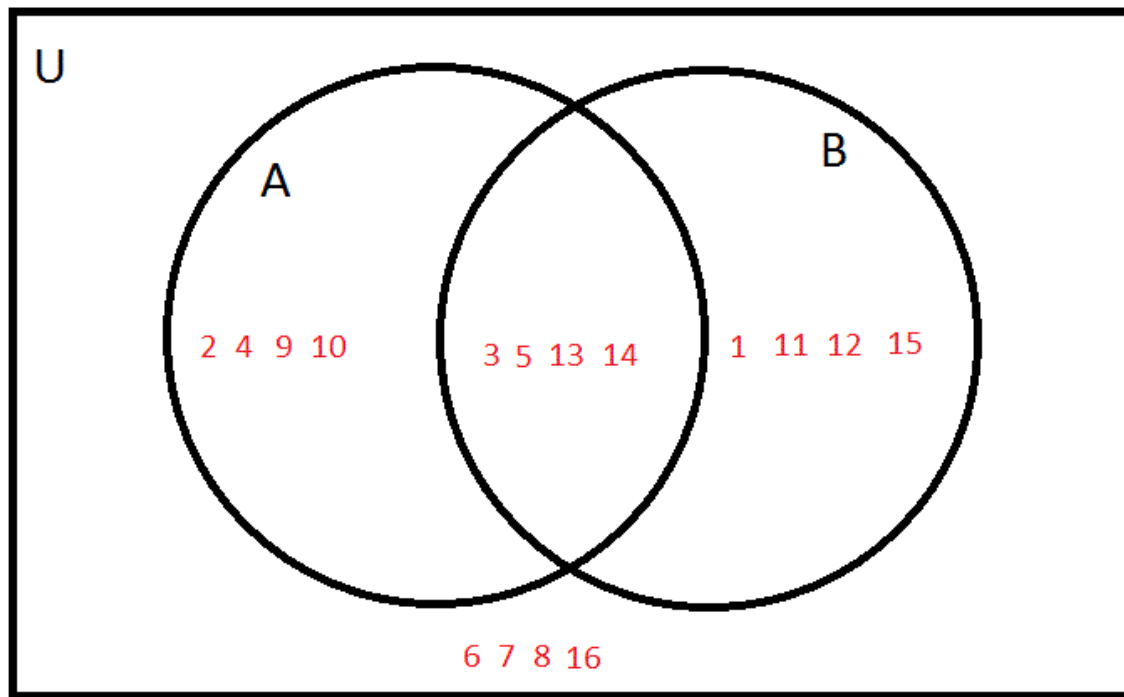
A set with one element is called a singleton set

## Venn Diagrams

In Venn diagrams the universal set  $U$  is represented by a rectangle

Inside the Venn Diagram for  $U$ , circles or other geometrical figures are used to represent sets

Sometimes points are used to represent the particular elements of the set



## Some things to remember

Sets can be elements of sets:  $\{\{1, 2, 3\}, a, \{b, c\}\}, \{N, Z, Q, R\}$

Let  $A = \{\{a\}, \{b\}, \{a, b\}\}$ , In this case  $\{a\} \in A$ , but  $a \notin A$

The empty set is different from a set containing the empty set

An empty set is a subset of any set

## Set Equality

Two sets are equal if and only if they have the same elements (order don't matter)

Therefore if A and B are sets, then A and B are equal if and only if  $\forall x(x \in A \Leftrightarrow x \in B)$

Remember:

- order is immaterial
- multiplicity is ignored

## Subsets

The set A is a subset of B, if and only if every element of A is also an element of B

The notation  $A \subseteq B$  is used to indicate that A is a subset of the set B

$A \subseteq B$  holds if and only if  $\forall x(x \in A \rightarrow x \in B)$  is true

1. Because  $x \in \emptyset$  is always false,  $\emptyset \subseteq S$ , for every set  $S$
2. Because  $x \in S \rightarrow x \in S$ ,  $S \subseteq S$ , for every set  $S$

"Every nonempty set  $S$  is guaranteed to have at least two subsets, the empty set and the set  $S$  itself"

(super class)

## Supersets

If set  $A$  is a subset of set  $B$ , then set  $B$  is a superset of set  $A$

The notation  $B \supseteq A$  is used to indicate that  $B$  is a superset of the set  $A$

$A \subseteq B$  and  $B \supseteq A$  are equivalent statements

(sub class)

## Showing a Set is or is not a Subset of Another Set

Showing that  $A$  is a Subset of  $B$ : To show that  $A \subseteq B$ , show that if  $x$  belongs to  $A$ , then  $x$  also belongs to  $B$

Showing that  $A$  is not a Subset of  $B$ : To show that  $A$  is not a subset of  $B$ ,  $A \not\subseteq B$ , find an element  $x \in A$  with  $x \notin B$ . (Such an  $x$  is a counterexample to the claim that  $x \in A$  implies  $x \in B$ )

(To show not a subset, just proof existence for  $x \in A$  and  $x \notin B$ )

## Another look at Equality of Sets

$\forall x(x \in A \Leftrightarrow x \in B)$  is equal to  $\forall x [(x \in A \rightarrow x \in B) \wedge (x \in B \rightarrow x \in A)]$  or  $(A \subseteq B)$  and  $(B \subseteq A)$

## Proper Subsets

If  $A \subseteq B$ , but  $A \neq B$ , then we say  $A$  is a proper subset of  $B$ , denoted by  $A \subset B$

If  $A \subset B$ , then:

$\forall x(x \in A \rightarrow x \in B) \wedge \exists x(x \in B \wedge x \notin A)$  must be True

## Set Cardinality

the size of the set (ignore the duplicate elements)

The cardinality of a finite set  $A$ , denoted by  $|A|$

1.  $|\emptyset| = 0$
2. Let  $S$  be the letters of the English alphabet. Then  $|S| = 26$
3.  $|\{1, 2, 3\}| = 3$
4.  $|\{\emptyset\}| = 1$
5. The set of integers is infinite

## Power Sets

The set of all subsets of a set  $A$ , denoted  $P(A)$ , is called the power set of  $A$

If  $A = \{a, b\}$  then  $P(A) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$

Q) What is the power set of  $\emptyset$ ?

A)  $P(\emptyset) = \{\emptyset\}$

Q) What is the power set of  $\{\emptyset\}$ ?

A)  $P(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}$

## Tuples

Because sets are unordered, a different structure is needed to represent ordered collections. This is provided by ordered n-tuples

The ordered n-tuple  $(a_1, a_2, \dots, a_n)$  is the ordered collection that has  $a_1$  as its first element and  $a_2$  as its second element and so on until  $a_n$  as its last element

Two n-tuples are equal if and only if their corresponding elements are equal

2-tuples are called ordered pairs

The ordered pairs  $(a, b)$  and  $(c, d)$  are equal if and only if  $a = c$  and  $b = d$

Order does matter

## Cartesian Product

The Cartesian Product of two sets  $A$  and  $B$ , denoted by  $A \times B$  is the set of ordered pairs  $(a, b)$  where  $a \in A$  and  $b \in B$

$$A * B = \{(a, b) \mid a \in A \wedge b \in B\}$$

Example:

$$A = \{a, b\} \text{ and } B = \{1, 2, 3\}$$

$$A \times B = \{(a, 1), (a, 2), (a, 3), (b, 1), (b, 2), (b, 3)\}$$

### relation

A subset  $R$  of the Cartesian product  $A \times B$  is called a relation from the set  $A$  to the set  $B$

A relation from a set  $A$  to itself is called a relation on  $A$

$$A * \emptyset = \emptyset \text{ for any set } A$$

We use the notation  $A^2$  to denote  $A \times A$

Similarly  $A^3 = A \times A \times A$  and so on...

## Truth Sets of Quantifiers



Given a predicate  $P$  and a domain  $D$ , we define the truth set of  $P$  to be the set of elements in  $D$  for which  $P(x)$  is true

The truth set of  $P(x)$  is denoted by  $\{x \in D \mid P(x)\}$

## Boolean Algebra

Propositional calculus and set theory are both instances of an algebraic system called a Boolean Algebra

As always there must be a universal set  $U$ . All sets are assumed to be subsets of  $U$

## Set Operations

methods you can apply on sets to get new set

Most commonly used set theory operations are:

- Union
- Intersection
- Difference
- Complementation
- Symmetric Difference

### Union

Let  $A$  and  $B$  be sets. The union of the sets  $A$  and  $B$ , denoted by  $A \cup B$ , is the set

$$A \cup B = \{x \mid x \in A \vee x \in B\}$$

Set of all unique elements from  $A$  and  $B$

### Intersection

The intersection of sets  $A$  and  $B$ , denoted by  $A \cap B$ , is

$$A \cap B = \{x \mid x \in A \wedge x \in B\}$$

Set of all unique elements in both  $A$  and  $B$

### Difference

The difference of  $A$  and  $B$ , denoted by  $A - B$ , is the set containing the elements of  $A$  that are not in  $B$

$$A - B = \{x \mid x \in A \wedge x \notin B\}$$

Set of all unique element that only in  $A$

### Complement

If  $A$  is a set, then the complement of the  $A$  (with respect to  $U$ ), denoted by  $\overline{A}$  is the set  $U - A$

which means  $A + \overline{A} = U$ , or Those elements in the universal set  $U$  but not in  $A$

$$\overline{A} = \{x \mid x \in U \mid x \notin A\}$$

## Another Interpretation of Set Difference

$$A - B = \{x \mid x \in A \wedge x \notin B\} = A \cap \overline{B}$$

## The Cardinality of the Union of Two Sets

$$|A \cup B| = |A| + |B| - |A \cap B|$$

## Symmetric Difference

The symmetric difference of A and B, denoted by  $A \oplus B$  is the set  $(A - B) \cup (B - A)$

The set of all unique elements from A or B (in A or in B, not both)

Symmetric Difference is different from Complement of Intersection

## Set Identities

---

### Identity laws

- $A \cup \emptyset = A$
- $A \cap U = A$

### Domination laws

- $A \cup U = U$
- $A \cap \emptyset = \emptyset$

### Idempotent laws

- $A \cup A = A$
- $A \cap A = A$

### Complementation law

- The complement of  $\overline{A}$  is A (markdown cannot display double overline)
- 

### Commutative laws

$$A \cup B = B \cup A$$

$$A \cap B = B \cap A$$

### Associative laws

$$A \cup (B \cup C) = (A \cup B) \cup C$$

$$A \cap (B \cap C) = (A \cap B) \cap C$$

### Distributive laws

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

De Morgan's laws

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

Absorption laws

$$A \cup (A \cap B) = A$$

$$A \cap (A \cup B) = A$$

Complement laws

$$A \cup \overline{A} = U$$

$$A \cap \overline{A} = \emptyset$$

## Proving Set Identities

Venn diagrams "proofs" are considered informal

Different ways to formally prove set identities:

1. prove both sets are subset of the other
2. Use set builder notation and propositional logic to transform one side of the identity to the other
3. Membership Tables: Verify that elements in the same combination of sets always either belong or do not belong to the same side of the identity. Use 1 to indicate it is in the set and a 0 to indicate that it is not

## Membership Table

similar with truth table, proof by giving all possible values

1 indicate the element belong to the table, 0 indicate not

A	B	$A \cap B$	$A \cup B$
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	0

## Generalized Unions and Intersections

Let  $A_1, A_2, \dots, A_n$  be an indexed collection of sets.

$$\bigcup_{i=1}^n A_i = A_1 \cup A_2 \cup \dots \cup A_n$$

$$\bigcap_{i=1}^n A_i = A_1 \cap A_2 \cap \dots \cap A_n$$

These are well defined, since union and intersection are associative.

For  $i = 1, 2, \dots$ , let  $A_i = \{i, i + 1, i + 2, \dots\}$ . Then,

$$\bigcup_{i=1}^n A_i = A_1$$

$$\bigcap_{i=1}^n A_i = A_n$$

## Multisets

A multiset (short for multiple-membership set) is an unordered collection of elements where an element can occur as a member more than once

(a set with some duplicated elements)

Multisets use a similar notation as sets but for each element we also list its multiplicity i.e. the number of times it occurs

e.g.  $\{4 \cdot a, 1 \cdot b, 3 \cdot c\}$  (there are 4 a, 1 b and 3 c in this set)

The cardinality of a multiset is defined to be the sum of the multiplicities of its elements

(sum of the all elements, including duplicated)

## Multiset Operations

- The union of the multisets  $P$  and  $Q$  ( $P \cup Q$ ) is the multiset in which the multiplicity of an element is the maximum of its multiplicities in  $P$  and  $Q$
- The intersection of  $P$  and  $Q$  ( $P \cap Q$ ) is the multiset in which the multiplicity of an element is the minimum of its multiplicities in  $P$  and  $Q$
- The difference of  $P$  and  $Q$  ( $P - Q$ ) is the multiset in which the multiplicity of an element is the multiplicity of the element in  $P$  less its multiplicity in  $Q$  unless this difference is negative, in which case the multiplicity is 0
- The sum of  $P$  and  $Q$  ( $P + Q$ ) is the multiset in which the multiplicity of an element is the sum of multiplicities in  $P$  and  $Q$

Example:

Suppose that  $P$  and  $Q$  are the multisets  $\{4 \cdot a, 1 \cdot b, 3 \cdot c\}$  and  $\{3 \cdot a, 4 \cdot b, 2 \cdot d\}$ , respectively. Find  $P \cup Q$ ,  $P \cap Q$ ,  $P - Q$ , and  $P + Q$ .

$$P \cup Q = \{\max(4, 3) \cdot a, \max(1, 4) \cdot b, \max(3, 0) \cdot c, \max(0, 2) \cdot d\} = \{4 \cdot a, 4 \cdot b, 3 \cdot c, 2 \cdot d\}$$

$$P \cap Q = \{\min(4, 3) \cdot a, \min(1, 4) \cdot b, \min(3, 0) \cdot c, \min(0, 2) \cdot d\} = \{3 \cdot a, 1 \cdot b, 0 \cdot c, 0 \cdot d\} = \{3 \cdot a, 1 \cdot b\}$$

$$P - Q = \{\max(4 - 3, 0) \cdot a, \max(1 - 4, 0) \cdot b, \max(3 - 0, 0) \cdot c, \max(0 - 2, 0) \cdot d\} = \{1 \cdot a, 0 \cdot b, 3 \cdot c, 0 \cdot d\} = \{1 \cdot a, 3 \cdot c\}$$

$$P + Q = \{(4 + 3) \cdot a, (1 + 4) \cdot b, (3 + 0) \cdot c, (0 + 2) \cdot d\} = \{7 \cdot a, 5 \cdot b, 3 \cdot c, 2 \cdot d\}$$

## Functions

Let  $A$  and  $B$  be nonempty sets. A function  $f$  from  $A$  to  $B$ , denoted  $f: A \rightarrow B$  is an assignment of each element of  $A$  to exactly one element of  $B$ . We write  $f(a) = b$  if  $b$  is the unique element of  $B$  assigned by the function  $f$  to the element  $a$  of  $A$ .

Functions are sometimes called mappings or transformations.

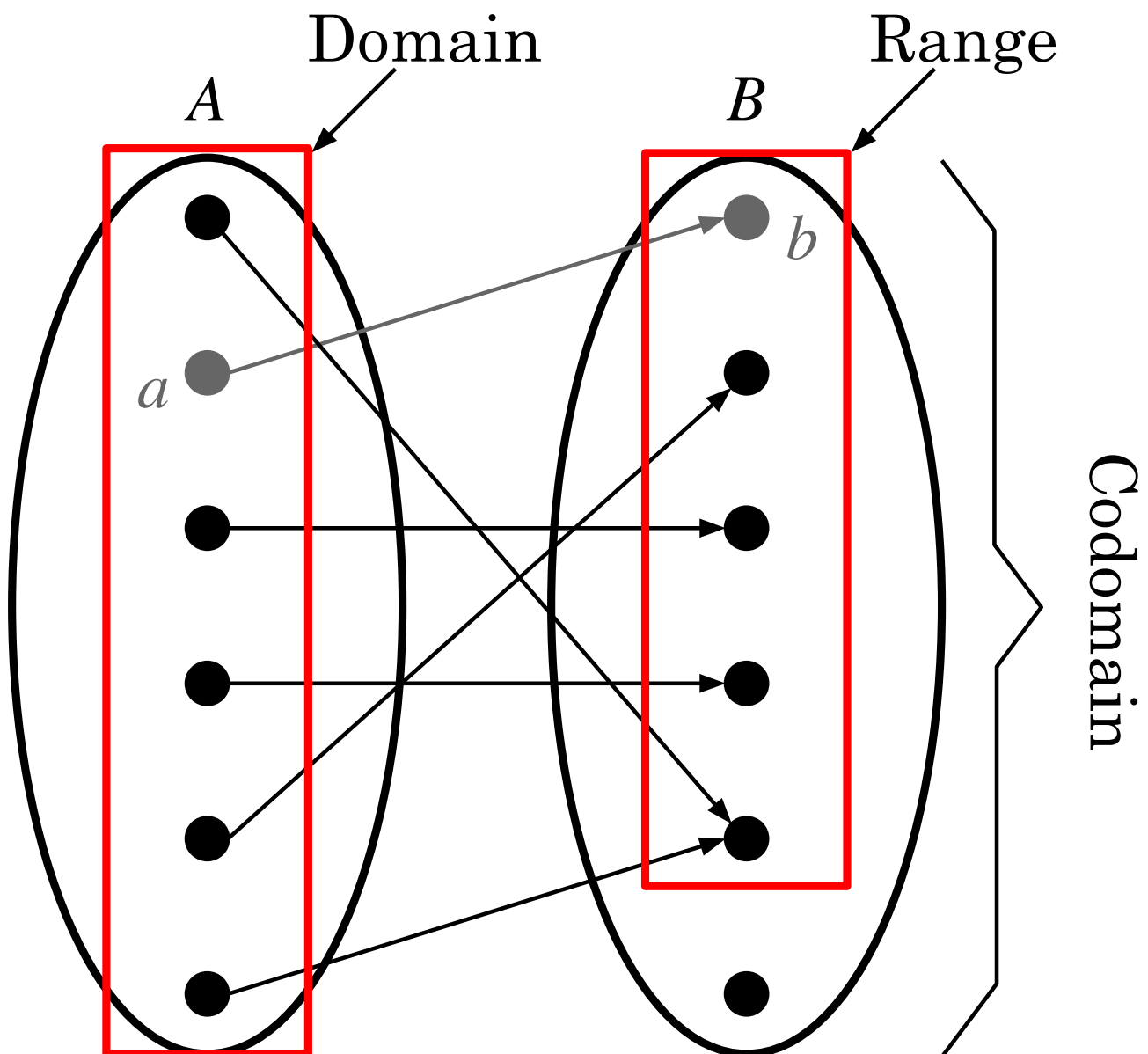
A function  $f: A \rightarrow B$  can also be defined as a subset of  $A \times B$  (a relation). This subset is restricted to be a relation where no two elements of the relation have the same first element.

Specifically, a function  $f$  from  $A$  to  $B$  contains one, and only one ordered pair  $(a, b)$  for every element  $a \in A$ .

(function is an one to one mapping relationship)

Given a function  $f: A \rightarrow B$ :

- We say  $f$  maps  $A$  to  $B$  or  $f$  is a mapping from  $A$  to  $B$
- $A$  is called the domain of  $f$
- $B$  is called the codomain of  $f$
- If  $f(a) = b$ ,
  - then  $b$  is called the image of  $a$  under  $f$
  - $a$  is called the preimage of  $b$
- The range of  $f$  is the set of all images of points in  $A$  under  $f$ . We denote it by  $f(A)$



Two functions are equal when they have the same domain, the same codomain and map each element of the domain to the same element of the codomain

## Domain, Codomain and Range

domain: what may comes in

codomain: what may comes out

range: what actually comes out

## Representing Functions

Functions may be specified in different ways

1. An explicit statement of the assignment. Students and grades example
2. A formula  $f(x) = x + 1$

### 3. A computer program

## image

image of  $a$  is  $b$  means  $f(a) = b$ , where  $b$  is image and  $a$  is pre-image

income is image, outcome is pre-image

## Question on Functions and Sets

if  $S$  is a subset of  $A$ ,  $s$  belongs to  $S$ ,  $B$  is codomain of  $A$ , then  $F(s)$  is a subset of  $B$

because  $s$  must fall in  $A$ , so  $F(x)$  must fall in  $B$

## Increasing and Decreasing Functions

increasing function: increasing if

$$\forall x_1 \forall x_2 (x_1 < x_2 \rightarrow f(x_1) \leq f(x_2))$$

this may have somewhere horizontal

strictly increasing function: increasing if  $\forall x_1 \forall x_2 (x_1 < x_2 \rightarrow f(x_1) < f(x_2))$

no horizontal line, if  $x_1 > x_2$ , then  $f(x_1)$  is always larger than  $f(x_2)$

decreasing function and strictly decreasing function are the reverse

## Injection

Injection is also called one-to-one, which means for each value in domain will always has it unique image in range. (if  $f(a) = f(b)$ , then  $a = b$ )

## Surjection

Surjections means a function's codomain is equal to it's range, it is also called onto

## Bijection

A function that is both Injection and Surjection

## showing function is injection or surjection

Example:

Suppose that  $f: A \rightarrow B$

injective: show that if  $f(x) = f(y)$ , then  $x = y$

surjective: show that for any  $x$  in codomin, there is a  $y$  in domain such that  $f(y) = x$

$f(x) = 4x - 1$  show one-to-one and onto

one-to-one:

$$4a - 1 = 4b - 1$$

$$4a = 4b$$

$$a = b$$

onto:

assume there is a  $y$  such that  $y = 4x - 1$

$$y - 1 = 4x$$

$$x = (y - 1) / 4$$

There exist a function to project  $y$  to  $x$ , so it is surjective

book example:

Solution:

Part 1 (scratch work)

Let us pick an arbitrary number  $y \in \mathbb{R}$  in the codomain. If such a number exists,

$$2x - 3 = y \quad 2x = y + 3 \quad x = (y + 3)/2$$

Part 2 (actual solution) Let  $y \in \mathbb{R}$  be an arbitrary element from the codomain

$$\text{Let } x = (y + 3)/2$$

$f(x) = 2((y + 3)/2) - 3 = y + 3 - 3 = y$  An arbitrary element from the codomain has a preimage in the domain  $\therefore f(x) = 2x - 3$ , from  $\mathbb{R}$  to  $\mathbb{R}$ , is surjective

## Inverse function

All inverse functions are bijective

Let function be  $f$ , then the inverse function of it is  $f^{-1}$

$$f(x) = y \text{ iff } f^{-1}(y) = x$$

inverse function is the projection from image to pre-image

All bijective functions are invertible

## Composition

let  $f$  from  $B$  to  $C$ ,  $g$  from  $A$  to  $B$ , then the composition of  $f$  with  $g$  denote as  $f \circ g$  is the function from  $A$  to  $C$  defined by  $f \circ g = f(g(x))$

the inner function will be calculated first



## Floor and Cell function

Floor function return the largest integer that less than or equal to x

Celling function return the smallest integer that larger than or equal to x

$$\lfloor 4.5 \rfloor = 4, \lfloor -4.5 \rfloor = -5$$

$$\lceil 4.5 \rceil = 5, \lceil -4.5 \rceil = -4$$

## properties of flooring and ceiling functions

n is a integer, and x is a real number

1.  $\lfloor x \rfloor = n$  iff  $n \leq x < n + 1$
2.  $\lceil x \rceil = n$  iff  $n - 1 < x \leq n$
3.  $\lfloor x \rfloor = n$  iff  $x - 1 < n \leq x$
4.  $\lceil x \rceil = n$  iff  $x \leq n < x + 1$
5.  $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$
6.  $\lfloor -x \rfloor = -\lceil x \rceil$
7.  $\lceil -x \rceil = -\lfloor x \rfloor$
8.  $\lfloor x + n \rfloor = \lfloor x \rfloor + n$
9.  $\lceil x + n \rceil = \lceil x \rceil + n$

## Proving properties of functions

prove if x is a real number, then  $\lfloor 2x \rfloor = \lfloor x \rfloor + \lfloor x + 1/2 \rfloor$

let  $x = n + \epsilon$ , where n is an integer and  $0 \leq \epsilon < 1$ .

case 1:  $\epsilon < 1/2$

$$\lfloor 2x \rfloor = \lfloor 2n + 2\epsilon \rfloor = 2n \text{ since } \epsilon < 1/2$$

$$\lfloor x \rfloor + \lfloor x + 1/2 \rfloor = \lfloor n + \epsilon \rfloor + \lfloor n + \epsilon + 1/2 \rfloor = n + n \text{ since } \epsilon < 1/2 \text{ and correspondingly } \epsilon + 1/2 < 1$$

case 2:  $\epsilon \geq 1/2$

$$\lfloor 2x \rfloor = \lfloor 2n + 2\epsilon \rfloor = 2n + 1 \text{ since } \epsilon \geq 1/2$$

$$\lfloor x \rfloor + \lfloor x + 1/2 \rfloor = \lfloor n + \epsilon \rfloor + \lfloor n + \epsilon + 1/2 \rfloor = n + (n + 1) \text{ since } \epsilon > 1/2 \text{ and correspondingly } \epsilon + 1/2 > 1$$

## Factorial function

$f: \mathbb{N} \rightarrow \mathbb{Z}^+$ , denoted by  $f(n) = n!$  is the product of the first n positive integers when n is a nonnegative integer.

$$f(n) = 1 * 2 * 3 \dots * n$$

$$f(0) = 0! = 1$$

$$f(1) = 1! = 1$$

## Partial function

A partial function from  $A \rightarrow B$  is an assignment of each element in subset of  $A$  to  $B$ , which means not every element in domain is used.

for example,  $f: \mathbb{Z} \rightarrow \mathbb{R}, f(x) = \sqrt{x}$  is a partial function because  $f(x)$  is undefined when  $x$  is negative

## Sequences and Summations

Sequences are ordered lists of elements

Example:

1, 2, 3, 5, 8...

1, 3, 9, 27, 81...

The notion of position is important in sequences.

It is the index at which a certain value appears in the sequence

## Formal definition of sequences

A sequence is a function from a subset of the integers (usually  $\{1, 2, 3, \dots\}$  or  $\{0, 1, 2, 3, \dots\}$ ) to a set  $S$

The notation  $a_n$  is used to denote the image of the integer  $n$ .

## Types of sequences

Geometric Progression:

A geometric progression is a sequence of the form  $a, ar, ar^2, \dots, ar^n, \dots$

where the initial term  $a$  and the common ratio  $r$  are real numbers

Arithmetic Progression

A arithmetic progression is a sequence of the form  $a, a + d, a + 2d, \dots, a + nd, \dots$

where the initial term  $a$  and the common difference  $d$  are real numbers

## Summations

Sum of the terms  $a_m, a_{m+1}, \dots, a_n$ ,

Notation:  $\sum_{i=m}^n a_i$

(starting from  $a_m$  stop at  $a_n$ )

## Geometric Series

Sum of terms of geometric progressions:

$\sum_{j=0}^n ar^j = \frac{ar^{n+1} - a}{r - 1}$  when  $r \neq 1$

$$\sum_{j=0}^n ar^j = \frac{a(r^{n+1} - 1)}{r - 1} \text{ when } r \neq 1$$

## Some Useful Summation formulae

Sum	Closed Form
$\sum_{j=0}^n ar^j \ (r \neq 1)$	$\frac{ar^{n+1} - a}{r - 1}, r \neq 1$
$\sum_{j=0}^n j$	$\frac{n(n+1)}{2}$
$\sum_{j=0}^n j^2$	$\frac{n(n+1)(n+2)}{6}$
$\sum_{j=0}^n j^3$	$\frac{n^2(n+1)^2}{4}$
$\sum_{j=0}^{\infty} x^j,  x  < 1$	$\frac{1}{1-x}$
$\sum_{j=0}^{\infty} jx^{j-1},  x  < 1$	$\frac{1}{(1-x)^2}$

## chapter 3

### Problems and Algorithms

We then solve the general problem by specifying the steps of a procedure that takes a valid input and produces the desired output. This procedure is called an algorithm.

Definition:

An algorithm is a finite set of precise instructions for

performing a computation or for solving a problem.

Human understandable words:

The steps of a general question to get desired output from any valid input

Example:

describe an algorithm for finding the maximum value in a finite sequence of integers

1. Set the temporary maximum equal to the first integer in the

sequence.

2. Compare the next integer in the sequence to the temporary

maximum.

- If it is larger than the temporary maximum, set the temporary maximum equal to this integer.

3. Repeat the previous step if there are more integers. If not, stop.

4. When the algorithm terminates, the temporary maximum is the

largest integer in the sequence.

### Specifying Algorithms

Algorithms can be specified in different ways.

- Human language
- Flowchart
- Pseudocode

## flowchart

graphic representation of an algorithm

operations, instructions and series of instructions are represented by boxes of different shapes

The flow of control is represented by directed lines connecting the boxes

Unwieldy for algorithms of even moderate complexity

## Properties of Algorithms

Input: have input values from a specified set

Output: performs an action or produces the output

Correctness: produce the correct output values for each set of input values

Finiteness: should produce the output after a finite number of steps for any input

Effectiveness: it must be possible to perform each step of the algorithm correctly and in a finite amount of time

Generality: should work for all problems of the desired form

## Search problem

---

Definition: The general searching problem is to locate an element  $x$  in the list of distinct elements  $a_1, a_2, \dots, a_n$ , or determine that it is not in the list.

The solution to searching problem is the location of the term in the list that equals  $x$  or 0 if  $x$  is not in the list

### Linear Search

Search from beginning of the list to the end of the list

### Binary Search

Assume the input is a list of items in increasing order

The algorithm begins by comparing the element to be found with the middle element.

- if the middle element is lower, the search proceeds with the upper half of the list
- otherwise, the search proceeds with the lower half of the list

Repeat this process until we have a list of size 1.

- if the element remain equal to the element in the list, the position is returned.
- otherwise, return 0

## Sorting

To sort the elements of a list is to put them in increasing order

### Bubble Sort

Bubble sort makes multiple passes through a list. Every pair of elements that are found to be out of order are interchanged

1. compare two elements sit together each time, if the first is larger, exchange position with second element.
2. repeat from beging to the end of the list
3. repeat (1-2) for number of elements in list times
4. each time step 3 is done, the largest value will be pushed to the end of the list

Although bubble sort is considered to be an extremely inefficient algorithm, it is possible to slightly tweak the algorithm so as to give bubble sort the ability to handle already sorted lists very efficiently

### Selection Sort

The selection sort begins by finding the least element in the list. This element is moved to the front. Then the least element among the remaining elements is found and put into the second position. This procedure is repeated until the entire list has been sorted.

1. find max from index 1 to end
2. exchange max and index 1 element
3. find max from index 2 to end
4. exchange max and index 2 element
5. repeat for number of elements times...

### Insertion Sort

An optimized version of the algorithm uses fewer swaps but instead uses a temporary location to store the element that needs to be assigned the correct location

1. Get index 2, consider index 1 as another list, call in ret list
2. loop through ret, find the correcct index for index 2
3. insert index2 to ret in the correct position
4. repeat for the remain part of the list
5. ret is the sorted list

## Greedy Algorithms

Optimization problems minimize or maximize some parameter over all

Takes the best one each step instead of over all best one

It not always comes with the best over all result even it comes out with best each step results.

After specifying what the “best choice” at each step is, we try to prove that this approach always produces an optimal solution, or find a counterexample

This is a very important step because in some cases greedy algorithms yield a very poor solution

## Usage of greedy Algorithms

Try to prove the US coins produces change using the fewest coins possible

1. Assume there is a positive integer  $n$  such that change can be made for  $n$  cents using quarters, dimes, nickels, and pennies, with a fewer total number of coins than given by the algorithm.
2. Then,  $\dot{q} \leq q$  where  $\dot{q}$  is the number of quarters used in this optimal way and  $q$  is the number of quarters in the greedy algorithm's solution. But this is not possible by Lemma 1, since the value of the coins other than quarters can not be greater than 24 cents.
3. Similarly, by Lemma 1, the two algorithms must have the same number of dimes, nickels, and quarters.

- The value of the coins other than quarters and dimes can not be greater than 9 cents.

- The value of the coins other than quarters, dimes and nickels can not be greater than 4 cents

## Unsolvable Problems

There is a class of problems for which it can be shown that no algorithm exists

Typically these problems require a yes/no answer, but where there cannot possibly be any algorithm that always gives the correct answer

The most famous problem in this category is The Halting Problem

Example: barber problem

barber will cut hair for everyone who don't cut hair themselves, do the barber cut hair himself?

## Growth of functions

In computer science, we want to understand how quickly an algorithm can solve a problem as the size of the input grows.

## Factors that affect the running time of a program

1. Size of the input to the program
2. The time complexity of the algorithm underlying the program
3. The quality of code generated by the compiler used to create the object program
4. The nature and speed of the instructions on the machine used to execute the program

In most of the case, 1 and 2 are the dominant factors

3 and 4 are largely independent from 1 and 2

## Big-O notation

Definition: Let  $f$  and  $g$  be functions from the set of integers or the set of real numbers to the set of real numbers. We say that  $f(x)$  is  $O(g(x))$  if there are constants  $C$  and  $k$  such that

$$|f(x)| \leq C|g(x)|$$

whenever  $x > k$

This is read as " $f(x)$  is big-O of  $g(x)$ " or " $g$  asymptotically dominates  $f$ "

The constants  $C$  and  $k$  are called witnesses to the relationship " $f(x)$  is  $O(g(x))$ ". Only one pair of witnesses is needed

## Some important points about Big-O Notation

If one pair of witnesses is found, then there are infinity many pairs. We can always make the  $k$  or the  $C$  larger and still maintain the inequality

$$|f(x)| \leq C|g(x)|$$

Any pair of  $C'$  and  $k'$  where  $C < C'$  and  $k < k'$  is also a pair of witnesses since whenever  $x > k' > k$ .

$$|f(x)| \leq C|g(x)| \leq C'|g(x)|$$

Sometimes in the literature you may see " $f(x) = O(g(x))$ " instead of " $f(x)$  is  $O(g(x))$ "

Strictly speaking this is an abuse of the equals sign since this notation does not represent a genuine equality.

It is ok to write  $f(x) \in O(g(x))$ , because  $O(g(x))$  represents the set of functions that are  $O(g(x))$ .

Usually, we will drop the absolute value sign since we will mostly deal with functions that take on positive values.

## Analogy for reasoning often used in Big-O problems

- Consider a weighing scale where on the one side you have an apple, a pear and a banana. On the other side you replace the apple by a bigger apple, the banana by a bigger banana and the pear by a bigger pear
- Obviously the first side of the scale will be lighter than the second side of the scale
- In our case we will look at an expression and replace smaller terms by larger terms. Obviously the original expression will be less than the resultant expression

Exmple:

Show that  $f(x) = x^2 + 2x + 1$  is  $O(x^2)$

Solution:

Consider  $x > 1$ ,  $x < x^2$  and  $1 < x^2$

$$x^2 + 2x + 1 \leq x^2 + 2x^2 + x^2 = 4x^2$$

take  $C = 4$  and  $k = 1$  as witnesses to show that

human understandable version

$$x^2 + 2x + 1 = 4x^2 \rightarrow x = 1 \rightarrow k = 1$$

$$C = 4k^2 = 4$$

So the pair is  $k = 1$  and  $C = 4$  (don't write answer like this, this is informal)

## Order

if two functions have the same Big-O notation, we say they have the same order

For example,  $f(x) = x^2 + 2x + 1$  and  $x^2$  are  $O(x^2)$

we say they are in the same order

## using definition of Big-O notation

Example: show that  $7x^2$  is  $O(x^3)$

solution: when  $x < 7$ ,  $7x^2 < x^3$ . Take  $C=1$  and  $k=7$  as witnesses to establish that  $7x^2$  is  $O(x^3)$

Example: show that  $n^2$  is not  $O(n)$

Solution:

Suppose there are constants  $C$  and  $k$  for which  $n^2 \leq Cn$ , whenever  $n > k$

- We can always choose  $C$  to be greater than  $k$  because if that is not so we can always choose another constant  $D > k$ . Now  $\forall n > k$

$n^2 \leq Cn$ , divide both sides by  $n$ , we get  $n \leq C$ , and  $n \leq C$  must hold all  $n > k$ , a contradiction.

For  $n > k$ ,  $n \leq C$

$\therefore k \leq C$

But we chose  $C$  to be greater than  $k$

## Big-O estimates for Polynomials

A mathematical expression of the following type is known as polynomial of degree  $n$

- $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$  In a polynomial of degree  $n$ , the leading term dominates its growth

We will prove the result that a polynomial of degree  $n$  is  $O(x^n)$

We will use the triangle inequality as the lemma for this proof

if  $x$  and  $y$  are real numbers, then  $|x| + |y| \leq |x + y|$

$$|f(x)| = |a_n x^n + a_{n-1} x^{n-1} + \dots|$$



$$\leq |a_n|x^n + |a_{n-1}|x^{n-1} + \dots$$

$$= x^n(|a_n| + |\frac{a_{n-1}}{x}| + \dots)$$

$$\leq x^n(|a_n| + |a_{n-1}| + \dots)$$

take  $C = |a_n| + |a_{n-1}| + \dots$ , and  $k=1$ . then  $f(x)$  is  $O(x^n)$

## important functions

$$f(x) = \sum_{i=0}^x 1 \text{ is } O(x^2)$$

$$f(x) = x! \text{ is } O(x^x)$$

$$f(x) = \log(x!) \text{ is } O(x \log(x))$$

## Sum and Production functions

$$(f_1 + f_2)(x) = f_1(x) + f_2(x)$$

$$(f_1 * f_2)(x) = f_1(x) * f_2(x)$$

example

$$f_1(x) = x^2, f_2(x) = -x^2 - x$$

$$(f_1 + f_2)(x) = x^2 - x^2 - x = -x$$

$$(f_1 * f_2)(x) = x^2(-x^2 - x) = -x^4 - x^3$$

## combination of functions

The number of steps used by a computer to solve a problem with input of a specified size, using an algorithm that is composed of subprocedures is the sum of number of steps used by the subprocedures

The big-O estimate of the algorithm is the sum of the big-O estimates of the subprocedures of the algorithm

if  $f_1(x)$  is  $O(g_1(x))$  and  $f_2(x)$  is  $O(g_2(x))$

then:

$$(f_1 + f_2)(x) \text{ is } O(\max(|g_1(x)|, |g_2(x)|))$$

$$(f_1 * f_2)(x) \text{ is } O(|g_1(x)| |g_2(x)|)$$

## Big-Omega notation

Definition: Let  $f$  and  $g$  be functions from the set of integers or the set of real numbers to the set of real numbers. We say that  $f(x)$  is  $\Omega(g(x))$  if there are constants  $C$  and  $k$  such that

$$|f(x)| \geq C|g(x)| \text{ when } x > k$$

We say that " $f(x)$  is big-Omega of  $g(x)$ "

Big-O gives an upper bound on the growth of a function, while Big-Omega gives the lower bound. Big-Omega tells us that a function grows at least as fast as another

Human word:

Big-O estimate the worst case, Big-Omega estimate the best case

Example:

Show that  $f(x) = 8x^3 + 5x^2 + 7$  is  $\Omega(g(x))$  where  $g(x) = x^3$

Solution:  $f(x) = 8x^3 + 5x^2 + 7 \geq 8x^3$  for all positive real numbers  $x$

## Big-Theta notation

Definition: Let  $f$  and  $g$  be functions from the set of integers or the set of real numbers to the set of real numbers. The function

$f(x)$  is  $\Theta(g(x))$  if  $f(x)$  is  $O(g(x))$  and  $f(x)$  is  $\Omega(g(x))$

We say that " $f$  is big-Theta of  $g(x)$ " and also that " $f(x)$  is of order  $g(x)$ " and also that " $f(x)$  and  $g(x)$  are of the same order."

$f(x)$  is  $\Theta(g(x))$  if and only if there exists constants  $C_1, C_2$  and  $k$  such that  $C_1g(x) < f(x) < C_2g(x)$  if  $x > k$ . This follows from the definitions of the big-O and big-Omega

Example:

Show that the sum of the first  $n$  positive integers is  $\Theta(n^2)$

Solution:

We have already shown that  $f(n)$  is  $O(n^2)$

to show that  $f(n)$  is  $\Omega(n^2)$ , we need a positive constant  $C$  such that  $f(n) > Cn^2$  for sufficiently large  $n$ . Summing only the terms greater than  $n/2$  we obtain the inequality

$$1+2+3+\dots+n \geq \lceil n/2 \rceil + \lceil n/2 \rceil + 1 + \dots + n$$

$$\geq \lceil n/2 \rceil + \lceil n/2 \rceil + \dots + \lceil n/2 \rceil$$

$$= (n - \lceil n/2 \rceil + 1) \lceil n/2 \rceil$$

$$\geq (n/2)(n/2) = n^2/4$$

$1+2+3+\dots+n$  is  $\Omega(n^2)$  because constant are abandoned

$1+2+3+\dots+n$  is  $\Theta(n^2)$

## Extra Example

Show that  $f(x) = 3x^2 + 8x \log x$  is  $\Theta(x^2)$

solution:

part 1:

for  $x > 1$ ,  $\log x$  is positive and  $\log x < x$

$$8x \log x < 8x^2$$

$$3x^2 + 8x \log x < 3x^2 + 8x^2$$

$$3x^2 + 8x \log x < 11x^2$$

$$3x^2 + 8x \log x \text{ is } O(x^2)$$

part 2:

$$3x^2 + 8x \log x > x^2$$

$$3x^2 + 8x \log x \text{ is } \Omega(x^2)$$

hencely,  $3x^2 + 8x \log x$  is  $\Theta(x^2)$

## Relating $O$ , $\Omega$ and $\Theta$

- If  $f(x)$  is  $O(g(x))$  then  $f$  is of order at most  $g$

–  $g$  is the upper-bound for  $f$

- If  $f(x)$  is  $\Omega(g(x))$  then  $f$  is of order at least  $g$

–  $g$  is the lower-bound for  $f$

- If  $f(x)$  is  $\Theta(g(x))$  then  $f$  is of order  $g$

–  $f$  is bounded both above and below by some multiple of  $g$

## Additional Big-Theta

when  $f(x)$  is  $\Theta(g(x))$ , then it must also be the case that  $g(x)$  is  $\Theta(f(x))$

## Big-Theta estimates for polynomials

Theorem: let  $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$

where  $a_0, a_1, \dots, a_n$  are real numbers with  $a_n \neq 0$

Then  $f(x)$  is of order  $x^n$  (or  $\Theta(x^n)$ )

Example:

The polynomial  $f(x) = 8x^5 + 5x^2 + 10$  is order of  $x^5$  (or  $\Theta(x^5)$ )

## Complexity of Algorithms

Given an algorithm, how efficient is this algorithm for solving a problem given input of a particular size?

We ask:

How much time does this algorithm use to solve a problem

How much computer memory does this algorithm use to solve a problem?

When we analyze the time the algorithm uses to solve the problem given input of a particular size, we are studying the time complexity of the algorithm.

When we analyze the computer memory the algorithm uses to solve the problem given input of a particular size, we are studying the space complexity of the algorithm.

## Space Complexity

To analyze the space complexity we often consider the "extra" memory needed i.e. not counting the memory needed to store the input itself

A big focus in space complexity is analyzing the memory requirements for data structures employed for processing the data. For recursive algorithms we also need to analyze the recursion stack space.

## Time Complexity

To analyze the time complexity of algorithms, we determine the number of operations, such as comparisons and arithmetic operations (addition, multiplication, etc.). We can estimate the time a computer may actually use to solve a problem using the amount of time required to do basic operations.

We focus on the worst-case time complexity of an algorithm. This provides an upper bound on the number of operations an algorithm uses to solve a problem with input of a particular size.

It is usually much more difficult to determine the average case time complexity of an algorithm. This is the average number of operations an algorithm uses to solve a problem over all inputs of a particular size.

## Complexity Analysis of Algorithms

Example: find the complexity of the algorithm for finding the max

```
procedure: max($a_1, a_2, ..., a_n$)
max:= $a_1$
for i:=2 to n
    if max < $a_i$ then max := $a_i$
return max
```

Solution:

- The  $\text{max} < a_i$  comparisons is made  $n - 1$  times
- Each time  $i$  is incremented, a test is made to see if  $i \leq n$
- One last comparison determines that  $i > n$
- Exactly  $2(n-1) + 1 = 2n - 1$  comparisons are made

Hence, the time complexity of the algorithm is  $\Theta(n)$

## worst case complexity of each algorithms

Linear Search:  $\Theta(n)$

Binary Search:  $\Theta(\log n)$

Bubble Sort:  $\Theta(n^2)$

Selection Sort:  $\Theta(n^2)$

Insertion Sort:  $\Theta(n^2)$

Matrix Multiplication  $(n \times n)$  :  $\Theta(n^3)$

## Algorithmic Paradigms

An algorithmic paradigm is a general approach based on a particular concept for constructing algorithms to solve a variety of problems

### Brute-force algorithms

- naive approach for solving problems; does not take advantage of any special structure of the problem or clever ideas
- sequential search, bubble sort, selection sort, insertion sort

### Greedy algorithms

- select the best choice at each step, instead of considering all sequences of steps; focus on local optimization and not overall optimization.

### Divide and conquer algorithms

- divide a problem into one or more instances of the same problem of smaller size and conquer the problem by using the solutions of the smaller problems to find a solution of the original problem
- quick sort, merge sort, fast matrix multiplication

### Dynamic programming

- recursively breaks down a problem into simpler overlapping subproblems, and stores the results of the subproblems in a table; computes the solution of the problem using the solutions of the subproblems
- cutting stock, matrix-chain multiplication, longest common subsequence

### Backtracking

- performs an exhaustive search of all possible solutions; once it is known that no solution can result from any further sequence of decisions, backtrack to a known point and work towards a solution with another series of decisions
- n-Queens, graph coloring, sum of subsets

## Probabilistic algorithms

- make random choices at one or more steps
- Monte Carlo algorithms, Las Vegas algorithms, Sherwood algorithms

## Understanding the Complexity of Algorithms

complexity	terminology
$\Theta(1)$	Constant complexity
$\Theta(\log n)$	Logarithmic complexity
$\Theta(n)$	Linear complexity
$\Theta(n \log n)$	Linearithmic complexity
$\Theta(n^b)$	Polynomial complexity
$\Theta(b^n)$ , where $b > 1$	Exponential complexity
$\Theta(n!)$	Factorial complexity

## Complexity size rank

Usually, the size of them follow this case

$$\log n < n < n \log n < n^2 < 2^n < n!$$

## Complexity of problems

Trackable problem: there exists a polynomial time algorithm to solve this problem. These problems are said to belong to the Class P.

Intrackable problem: there does not exist a polynomial time algorithm to solve this problem

Unsolvable problem: No algorithm exists to solve this problem

Class NP: Solution can be checked in polynomial time. But no polynomial time algorithm has been found for finding a solution to problems in this class.

NP Complete Class: If you find a polynomial time algorithm for one member of the class, it can be used to solve all the problems in the class.

## Number Theory

---

Number theory is the part of mathematics devoted to study of the integers and their properties

## Divisibility and Modular Arithmetic

Division: if  $a$  and  $b$  are integers with  $a \neq 0$ , then  $a$  divides  $b$  if there exist an integer  $c$  such that  $b = ac$

- The notation  $a \mid b$  denotes that  $a$  divides  $b$ .
- If  $a$  does not divide  $b$ , we write  $a \nmid b$ .

for example:  $3 \nmid 7$  and  $3 \mid 12$

different from divide by, the number being divided at back

$3 \mid 12$  is the same as  $\frac{12}{3}$  or  $12/3$  or  $12 \div 3$

## Properties of Divisibility

Theorem 1: let  $a$ ,  $b$  and  $c$  be integers, where  $a \neq 0$

1. if  $a \mid b$  and  $a \mid c$ , then  $a \mid (b + c)$
2. if  $a \mid b$ , then  $a \mid bc$  for all integer  $c$
3. if  $a \mid b$  and  $b \mid c$ , then  $a \mid c$

## Division Algorithm

when an integer is divided by a positive integer, there is a quotient and a remainder. This is traditionally called the "division algorithm", but is really a theorem.

Division Algorithm:

if  $a$  is an integer and  $d$  is a positive integer, then there are unique integers  $q$  and  $r$ , with  $0 \leq r < d$ , such that  $a = dq + r$

$d$  is called the divisor

$a$  is called the dividend

$q$  is called the quotient

$r$  is called the remainder

## +ve and -ve Remainders

$18/5 = 3$  remainder 3

- 3 more than the current bucket boundary

$18/5 = 4$  remainder -2

- 2 less than the next bucket boundary

From this perspective division will either yield a remainder of 0 or it will yield a +ve and a -ve remainder

We will only consider the +ve remainder

What are the quotient and remainder when -11 is divided by 3?

Quotient =  $-11 \div 3 = -4$

Remainder =  $-11 \bmod 3 = 1$

Only consider the positive remainder, so push Quotient from -3 to -4

## Procedure for +ve Remainder

Given dividend  $a$  and divisor  $d$  – Step 1: First find the quotient •  $q = \lfloor \frac{a}{d} \rfloor$

– Remember that  $\lfloor -x \rfloor = -\lceil x \rceil$

– Step 2: Find the +ve remainder

- $r = a - d * q$

$$q = \lfloor \frac{-11}{3} \rfloor = -\lceil \frac{11}{3} \rceil = -4$$

$$r = -11 - 3(-4) = 1$$

## Congruence Relation

if  $a$  and  $b$  are integers and  $m$  is a positive integer, then  $a$  is congruent to  $b$  modulo  $m$  if  $m$  divides  $a - b$

The notation  $a \equiv b \pmod{m}$  says that  $a$  is congruent to  $b$  modulo  $m$ .

We say that  $a \equiv b \pmod{m}$  is a congruence and that  $m$  is its modulus.

Two integers are congruent mod  $m$  if and only if they have the same remainder when divided by  $m$ .

If  $a$  is not congruent to  $b$  modulo  $m$ , we write  $a \not\equiv b \pmod{m}$

Human words: if two numbers divide by another number will come up with the same remainder, then they are Congruence

Example:

17 is congruent to 5 modulo 6 and 24 and 14 are not congruent modulo 6

$$17 - 5 = 12, 6 \mid 12$$

$$24 - 14 = 10, 6 \nmid 10$$

## More on Congruences

Let  $m$  be a positive integer. The integers  $a$  and  $b$  are congruent modulo  $m$  if and only if there is an integer  $k$  such that  $a = b + km$

## Note on proofs

when  $a$  = dividend,  $m$  = divisor,  $q$  = quotient,  $r$  = remainder

- $a = q * m + r$
- $m \mid (a-r)$
- given  $m \mid x$  then



- $x = k * m$

given  $a \equiv b \pmod{m}$  then

- $a \bmod m = b \bmod m$
- $m \mid (a-b)$
- $a = b + km$

## Congruence Class

The set of all integers congruent to  $a \bmod m$  is called the congruence class of  $a$  modulo  $m$

for example the congruence class for  $3 \bmod 4$  is  $\{\dots, -5, -1, 3, 7, 11, \dots\}$

explain:

$$q = -5 / 4 = \lfloor \frac{-5}{4} \rfloor = -\lceil \frac{5}{4} \rceil = -2$$

$$r = -5 - (-2 * 4) = 3$$

$$\text{also, } 3 \bmod 4 = 3$$

$$\text{so } -5 \bmod 4 = 3 \bmod 4$$

## Congruences of Sums and Products

Let  $m$  be a positive integer. If  $a \equiv b \pmod{m}$  and  $c \equiv d \pmod{m}$ , then  $a + c \equiv b + d \pmod{m}$  and  $ac \equiv bd \pmod{m}$

example:  $7 \equiv 2 \pmod{5}$ , and  $11 \equiv 1 \pmod{5}$

$$7 + 11 \equiv 2 + 1 \pmod{5}$$

$$18 / 5 = 3 \dots 3$$

$$3 / 5 = 0 \dots 3$$

## Algebraic Manipulation of Congruences

Multiplying both sides of a valid congruence by an integer preserves validity.

if  $a \equiv b \pmod{m}$  then  $a * c \equiv b * c \pmod{m}$  where  $c$  is any integer

if  $a \equiv b \pmod{m}$  then  $a + c \equiv b + c \pmod{m}$  where  $c$  is any integer

Division not always produce a valid congruence

## Arithmetic Modulo $m$

Definition: let  $\mathbb{Z}_m$  be the set of nonnegative integers less than  $m$ :  $\{0, 1, \dots, m-1\}$

the operation  $+_m$  is defined as  $a +_m b = (a+b) \bmod m$ , this is addition modulo  $m$

the operation  $a \cdot b \pmod m$  is defined as  $a \cdot b \pmod m = (ab) \pmod m$ , this is multiplication modulo  $m$

## Applications of Congruences

Hashing Functions:

A hashing function  $h$  assigns memory location  $h(k)$  to a record that has  $k$  as its key

A common hashing function is  $h(k) = k \pmod m$ , where  $m$  is the number of memory locations.

Because this hashing function is onto, all memory locations are possible.

Example: Let  $h(k) = k \pmod{111}$ .

This hashing function assigns the records of customers with social security numbers as keys to memory locations in the following manner:

$h(064212848) = 064212848 \pmod{111} = 14$

$h(037149212) = 037149212 \pmod{111} = 65$

$h(107405723) = 107405723 \pmod{111} = 14$ ,

but since location 14 is already

occupied, the record is assigned to the next available position, which is 15.

Pseudorandom Numbers:

Randomly chosen numbers are needed for many purposes, including computer simulations.

Pseudorandom numbers are not truly random since they are generated by systematic methods but they strive for some desirable properties of random numbers like uniformity and independence

The linear congruential method is one commonly used procedure for generating pseudorandom numbers.

Four integers are needed:

1. The modulus  $m$
2. the multiplier  $a$
3. the increment  $c$
4. the seed  $x_0$

where  $2 \leq a < m$ ,  $0 \leq c < m$ ,  $0 \leq x_0 < m$

We generate a sequence of pseudorandom numbers  $\{x_n\}$ , with  $0 \leq x_n < m$  for all  $n$ , by successively using the recursively defined function

$$x_{n+1} = (ax_n + c) \pmod m$$

If pseudo-random numbers between 0 and 1 are needed, then the

generated numbers are divided by the modulus,  $x_n / m$

Linear congruential generators provide a very efficient way to

generate pseudo-random numbers which are suitable for many applications

Unfortunately long pseudo-random number sequences do not

share some important statistical properties that true random numbers have. Because of this, it is not advisable to use them for some tasks, such as large simulations

### Check Digits

A common method of detecting errors in strings of digits is to add an extra at the end, which is evaluated using a function. If the final digit is not correct, then the string is assumed not to be correct

odd even check:

last bit = (number of bits in byte) mod 2

Also used in Universal Product Codes (UPCs) and International Standard Book Number (ISBN-10)

UPCs:

$$(3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \pmod{10})$$

ISBN:

$$x_{10} = \sum_{i=1}^9 ix_i \pmod{11}$$

## Integer Representations

usually, integers are represented in decimal

for example:  $965 = 9 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0$

We can represent numbers using any base  $b$ , where  $b$  is a positive integer greater than 1

The bases  $b = 2$  (binary),  $b = 8$  (octal), and  $b = 16$  (hexadecimal) are important for computing and communications

The ancient Mayans used base 20 and the ancient Babylonians used base 60

## Base $b$ representations

Let  $b$  be a positive integer greater than 1. Then if  $n$  is a positive integer, it can be expressed uniquely in the form:

$$n = a_k b^k + a_{k-1} b^{k-1} + \dots + a_1 b + a_0$$

(this is called base  $b$  expansion of  $n$ )

where  $k$  is a nonnegative integer,  $a_0, a_1, \dots, a_k$  are nonnegative integers less than  $b$ , and  $a_k \neq 0$ . The  $a_j, j = 0, \dots, k$  are called the base  $b$  digits of the representation.

We usually omit the subscript 10 for base 10 expansions.

## Base Conversion

To construct the base  $b$  expansion of an integer  $n$ :

- Divide  $n$  by  $b$  to obtain a quotient and remainder.

$$n = bq_0 + a_0 \quad 0 \leq a_0 \leq b$$

- The remainder,  $a_0$ , is the rightmost digit in the base  $b$  expansion of  $n$ . Next, divide  $q_0$  by  $b$ .

$$q_0 = bq_1 + a_1 \quad 0 \leq a_1 \leq b$$

- The remainder,  $a_1$ , is the second digit from the right in the base  $b$  expansion of  $n$ .
- Continue by successively dividing the quotients by  $b$ , obtaining the additional base  $b$  digits as the remainder. The process terminates when the quotient is 0.

Example:

Find the octal expansion of  $(12345)_{10}$

Solution: Successively dividing by 8 gives

$$12345 = 8 * 1543 + 1$$

$$1543 = 8 * 192 + 7$$

$$192 = 8 * 24 + 0$$

$$24 = 8 * 3 + 0$$

$$3 = 8 * 0 + 3$$

The Octal expansion of  $(12345)_{10}$  is  $(30071)_8$  (read from bottom to top)

## Conversion Between Binary, Octal, and Hexadecimal Expansions

Conversion between binary and octal and between binary and hexadecimal expansions is extremely easy because each octal digit corresponds to a block of three binary digits and each hexadecimal digit corresponds to a block of four binary digits

Table 2.1 Binary hexadecimal, and octal representation

Binary	Hexadecimal	Binary	Octal
0000	0	000	0
0001	1	001	1
0010	2	010	2
0011	3	011	3
0100	4	100	4
0101	5	101	5
0110	6	110	6
0111	7	111	7
1000	8		
1001	9		
1010	A		
1011	B		
1100	C		
1101	D		
1110	E		
1111	F		

Example:

convert \$(11111010111100)\_2\$ to oct ad hex

011 111 010 111 100 = 3 7 2 7 4 = \$(37274)\_8\$

0011 1110 1011 1100 = 3 E B C

## Induction and recursion

---

### Mathematical Induction

A powerful proof technique that is used to check conjectures about the outcomes of processes that occur repeatedly and according to definite patterns

Based on the rule of inference that if  $P(1)$  and  $\forall k(P(k) \rightarrow P(k+1))$  are true for the domain of positive integers, then  $\forall nP(n)$  is true

### Climbing an Infinite Ladder

Suppose we have an infinite ladder:

1. We can reach the first rung of the ladder.
2. If we can reach a particular rung of the ladder, then we can reach the next rung.

by 1, we can reach the first rung, then by 2, we can reach the second rung, again by 2, we can reach the third rung... We can apply w any number of ties to reach any paticular rung.

This example illustrates the reasoning employed for proof by mathematical induction.

### Principle of Mathematical Induction

Principle of Mathematical Induction: To prove that the propositional function  $P(n)$  is true for all positive integers  $n$ , we complete these steps:

- basis step: show that  $P(1)$  is true
- inductive step: show that  $P(k) \rightarrow P(k+1)$  is true for all integers  $k$

Climbing an Infinite Ladder Example:

BASIS STEP: By (1), we can reach rung 1

INDUCTIVE STEP: Assume the inductive hypothesis that we can reach rung  $k$ . Then by (2), we can reach rung  $k + 1$

Hence,  $P(k) \rightarrow P(k + 1)$  is true for all positive integers  $k$ . We can reach every rung on the ladder

Note:- To prove that  $P(n)$  is true for all positive integers  $n \geq a$ , in the Basis Step we show that  $P(a)$  is true

## Important Points About Using Mathematical Induction

Mathematical induction can be expressed as the rule of inference

$$(P(1) \wedge \forall k(P(k) \rightarrow P(k+1))) \rightarrow \forall n P(n)$$

where the domain is the set of all positive integers

In a proof by mathematical induction, we don't assume that  $P(k)$  is true for all positive integers! We show that if we assume that  $P(k)$  is true, then  $P(k + 1)$  must also be true. It is therefore not a case of begging the question, or circular reasoning

## Proving a Summation Formula by Mathematical Induction

show that  $\sum_{i=1}^n i = \frac{n(n+1)}{2}$

Basis step:

$$P(1) \text{ is true since } 1(1+1)/2 = 1$$

Inductive step:

Assume true for  $P(k)$ , the inductive hypothesis is  $\sum_{i=1}^k i = \frac{k(k+1)}{2}$

$$1 + 2 + 3 \dots + k + (k + 1) = \frac{k(k+1)}{2} + (k+1)$$

$$= \frac{k(k+1) + 2(k+1)}{2}$$

$$= \frac{(k+1)(k+2)}{2}$$

$$= \frac{(k+1)((k+1)+1)}{2}$$

Hence, we have shown that  $P(k+1)$  follows from  $P(k)$ . Therefore the sum of the first  $n$  integers is  $\frac{n(n+1)}{2}$

## Summation Formula

Conjecture and prove correct a formula for the sum of the first  $n$  positive odd integers. Then prove your conjecture.

$$1 = 1 = 1^2, 1 + 3 = 4 = 2^2, 1 + 3 + 5 = 9 = 3^2 \dots$$

We can conjecture that the sum of the first  $n$  positive odd integers is  $n^2$

$$1 + 3 + 5 + \dots + (2n-1) = n^2$$

Basis step:

$$P(1) \text{ is true since } 1 = 1^2$$

So,  $P(k) \rightarrow P(k+1)$  for every positive integer  $k$

Assume  $P(k)$ , then

$$1 + 3 + 5 + \dots + (2k-1) + (2k+1) = k^2$$

$$= [1 + 3 + 5 + \dots + (2k-1)] + (2k+1)$$

$$= k^2 + (2k+1) \text{ (from } P(k) \text{ is true)}$$

$$= (k+1)^2$$

Hence, we have shown that  $P(k+1)$  follows from  $P(k)$ . Therefore the sum of the first  $n$  positive odd integers is  $n^2$

## Proving inequalities

Example: Use mathematical induction to prove that  $n < 2^n$  for all positive integers  $n$

Let  $P(n)$  be the proposition that  $n < 2^n$

Basis step:

$$P(1) \text{ is true because } 1 < 2^1 = 2$$

Inductive step:

if  $P(k)$  is true, then  $P(k+1)$  is also true

(first part below from  $P(k)$ :  $k < 2^k$ )

$$k + 1 < (2^k) + 1 \leq 2^k + 2^k = 2 * 2^k = 2^{k+1}$$

Example:

prove that for every  $n$  which  $n \geq 4$ , there is  $2^n < n!$

Basis step:

$$P(4) \text{ is true because } 2^4 = 16 < 4! = 24$$

Inductive step:

Assume  $P(k)$  is true, then  $2^k < k!$

then  $P(k+1)$ :

$$2^{k+1} = 2 \cdot 2^k < 2 \cdot k! \text{ (from } P(k)) < (k+1)k! = (k+1)!$$

## Proving divisibility result

Use mathematical induction to prove that  $n^3 - n$  is divisible by 3 for every possible integer  $n$

Basis step:

$$1^3 - 1 = 0 \text{ which is divisible by 3}$$

Inductive step:

Assume  $P(k)$  is true, then  $k^3 - k$  is divisible by 3

$$P(k+1) = (k+1)^3 - (k+1)$$

$$= (k^3 + 3k^2 + 3k + 1) - (k + 1)$$

$$= (k^3 - k) + 3(k^2 + k)$$

$(k^3 - k)$  is divisible by 3 provide by  $P(k)$

$3(k^2 + k)$  is divisible by 3

so  $P(k+1)$  is divisible by 3

## Number of Subsets of a Finite Set

a set with  $n$  elements will have  $2^n$  subsets

## Guidelines: Mathematical Induction Proofs

Template for Proofs by Mathematical Induction

1. Express the statement that is to be proved in the form "for all  $n \geq b$ ,  $P(n)$ " for a fixed

integer  $b$ .

2. Write out the words "Basis Step." Then show that  $P(b)$  is true, taking care that the

correct value of  $b$  is used. This completes the first part of the proof.

3. Write out the words "Inductive Step".

4. State, and clearly identify, the inductive hypothesis, in the form "assume that  $P(k)$  is

true for an arbitrary fixed integer  $k \geq b$ ."

5. State what needs to be proved under the assumption that the inductive hypothesis is

true. That is, write out what  $P(k + 1)$  says.



6. Prove the statement  $P(k + 1)$  making use the assumption  $P(k)$ . Be sure that your proof

is valid for all integers  $k$  with  $k \geq b$ , taking care that the proof works for small values of  $k$ , including  $k = b$ .

7. Clearly identify the conclusion of the inductive step, such as by saying "this completes

the inductive step."

8. After completing the basis step and the inductive step, state the conclusion, namely,

by mathematical induction,  $P(n)$  is true for all integers  $n$  with  $n \geq b$ .

## Strong Induction

Strong mathematical induction is similar to ordinary mathematical induction in that there is a basis step and an inductive step

However, the inductive step shows that if  $P(j)$  is true for all positive integers  $j$  not exceeding  $k$ , then  $P(k + 1)$  is true.

Taking the dominoes analogy one step further, imagine that the dominoes are arranged in increasing order of weight and a given domino requires the combined weight of all the previous dominoes toppling over before it topples over as well.

Basis Step: Verify that the proposition  $P(1)$  is true.

Inductive Step: Show the conditional statement

$$[P(1) \wedge P(2) \wedge P(3) \dots \wedge P(k)] \rightarrow P(k+1)$$

holds for all positive integers  $k$

(rather than just  $P(k)$ , you need  $P(0), P(1), P(2) \dots P(k)$ )

## Strong Induction and the Infinite Ladder

Strong induction tells us that we can reach all rungs if:

1. We can reach the first rung of the ladder.
2. For every integer  $k$ , if we can reach the first  $k$  rungs, then we can reach the  $(k + 1)$ st rung.

To conclude that we can reach every rung by strong induction:

- BASIS STEP:  $P(1)$  holds
- INDUCTIVE STEP: Assume  $P(1) \wedge P(2) \wedge \dots \wedge P(k)$  holds for an arbitrary integer  $k$ , and show that  $P(k + 1)$  must also hold.

We will have then shown by strong induction that for every positive integer  $n$ ,  $P(n)$  holds, i.e., we can reach the  $n$ th rung of the ladder.

## Difference between mathematical induction and strong induction

Example: Suppose we can reach the first and second rungs of an infinite ladder, and we know that if we can reach a rung, then we can reach two rungs higher. Prove that we can reach every rung

Solution: Prove the result using mathematical induction.

BASIS STEP:

We can reach the first step.

ATTEMPTED INDUCTIVE STEP:

The inductive hypothesis is that we can reach the  $k^{\text{th}}$  rung.

However, assuming that we have reached the  $k^{\text{th}}$  rung, we know how to reach the  $(k+2)^{\text{nd}}$  rung, but we don't know how to reach the  $(k+1)^{\text{st}}$  rung.

Hence we cannot conclude that if we have reached the  $k^{\text{th}}$  rung we can reach the  $(k+1)^{\text{st}}$  rung as well.

Solution: Prove the result using strong induction.

Basis step:

we can reach the first step

inductive step:

The inductive hypothesis is that we can reach the first  $k$  rungs, for any  $k \geq 2$

Once we have reached the  $(k-1)^{\text{st}}$  rung, we can reach two rungs higher i.e. we can reach the  $(k+1)^{\text{st}}$  rung.

Hence, we can reach all rungs of the ladder.

## Which Form of Induction Should Be Used

- We can always use strong induction instead of mathematical induction. But there is no reason to use it if it is simpler to use mathematical induction

you should use mathematical induction when it is straightforward to prove that  $P(k) \rightarrow P(k+1)$  is true for all positive integers  $k$

- Use strong induction, and not mathematical induction, when you see how to prove that  $P(k+1)$  is true from the assumption that  $P(j)$  is true for all positive integers  $j$  not exceeding  $k$ , but you cannot see how to prove that  $P(k+1)$  follows from just  $P(k)$
- However, as already mentioned, in theory any result that can be proved by mathematical induction can be proved by strong induction and vice versa

## Completion of the proof of the Fundamental Theorem of Arithmetic

Example: Show that if  $n$  is an integer greater than 1, then  $n$  can be written as the product of primes.

Solution:

let  $P(n)$  be the proposition that  $n$  can be written as a product of primees

Basis step:

$P(2)$  is true since 2 itself is a prime

Inductive step:

The inductive hypothesis  $P(j)$  is true for all integers  $j$  with  $2 \leq j \leq k$ . To show that  $P(k+1)$  must be true under this assumption, two cases need to be considered

- if  $k+1$  is prime, then  $P(k+1)$  is true
- Otherwise,  $k+1$  is composite and can be written as product of two positive integers  $a$  and  $b$  with  $2 \leq a \leq b < k+1$ . By the inductive hypothesis  $a$  and  $b$  can be written as the product of primes and therefore  $k+1$  can also be written as the product of those primes.

Hence, it has been shown that every integer greater than 1 can be written as the product of primes.

## Proving a result using Strong Induction

Example: prove that every amount of postage of 12 cents or more can be formed using just 4-cent and 5-cent stamps

Basis step:

$P(12)$ ,  $P(13)$ ,  $P(14)$ ,  $P(15)$  are hold  $P(12)$  can be formed by three 4-cent  $P(13)$  can be formed by two 4-cent and one 5-cent  $P(14)$  can be formed by one 4-cent and two 5-cent  $P(15)$  can be formed by three 5-cent

Inductive step:

Assume that  $P(j)$  is true for  $12 \leq j \leq k$ , where  $k \geq 15$

$k - 3 \geq 12$

so we can form postage of  $k - 3$  cents using just 4-cent and 5-cent stamps

Now if we add postage of  $k - 3$  cents a 4-cents postage, we can form postage for  $k + 1$  cents

we have shown that if the inductive hypothesis is true, then  $P(k+1)$  is also true. This complete the inductive step

Because we have completed the basis step and the inductive step, we conclude that every postage of  $n$  cents, where  $n$  is at least 12, can be formed using 4-cent and 5-cent stamps

## Proving the same result using Mathematical Induction

Basis step:

$P(12)$  is true because 12 cents can be form by three 4 cents stamps

Inductive step:

Assume  $P(k)$  is ture for any  $k \geq 12$

we will consider two cases

1. The postage requires at least one 4-cent stamp

In this case, we can replace one 4-cent with a 5-cent stamp, then we get  $k+1$  cents

2. The postage does not require any 4-cent stamp Since  $k \geq 12$ , the postage will require at least three 5-cent stamps. In this case, we can replace three 5-cent stamps with four 4-cent stamps, we will get  $k+1$  cents

Because we have completed the basis step and the inductive step, we conclude that every postage of  $n$  cents, where  $n$  is at least 12, can be formed using 4-cent and 5-cent stamps.

## Recursive Algorithms

An algorithm is called recursive if it solves a problem by reducing it to an instance of the same problem with smaller input.

For the algorithm to terminate, the instance of the problem must eventually be reduced to some initial case for which the solution is known.

A recursive function consists of two parts:

1. A base case that is processed without recursion; and
2. A recursive case that reduces a particular case to one or more of the smaller cases, thereby making progress toward eventually reducing the problem all the way to the base case

Each successive invocation of the recursive function, other than the base case, is a slightly simpler or smaller version of the previous invocation

(function calling itself and only exit at specific cases)

## Recursive Factorial Algorithm

Give a recursive algorithm for computing  $n!$ , where  $n$  is a nonnegative integer.

```
Procedure factorial(n:nonnegative integer)
if n = 0 then
    return 1
else
    return n * factorial(n-1)
//out put is n!
```

## Background: Fibonacci Series

Fibonacci series are the numbers in the following sequence

- 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, ...
- the first two numbers are 0 and 1 and each subsequent number in the series is equal to the sum of the previous two numbers

```

procedure fibonacci(n: nonnegative integer)
if n ≤ 1 then
    return n
else
    return fibonacci(n-1) + fibonacci(n-2)
//output is nth Fibonacci number

```

## Recursive Binary Search Algorithm

```

procedure binary search(i, j, x : integers, 1 ≤ i ≤ j ≤ n)
m := ⌊(i + j)/2⌋
if x = am then
    return m
else if (x < am and i < m) then
    return binary search(i, m-1, x)
else if (x > am and j > m) then
    return binary search(m+1, j, x)
else
    return 0
//output is location of x in a1, a2, ..., an if it appears, otherwise 0

```

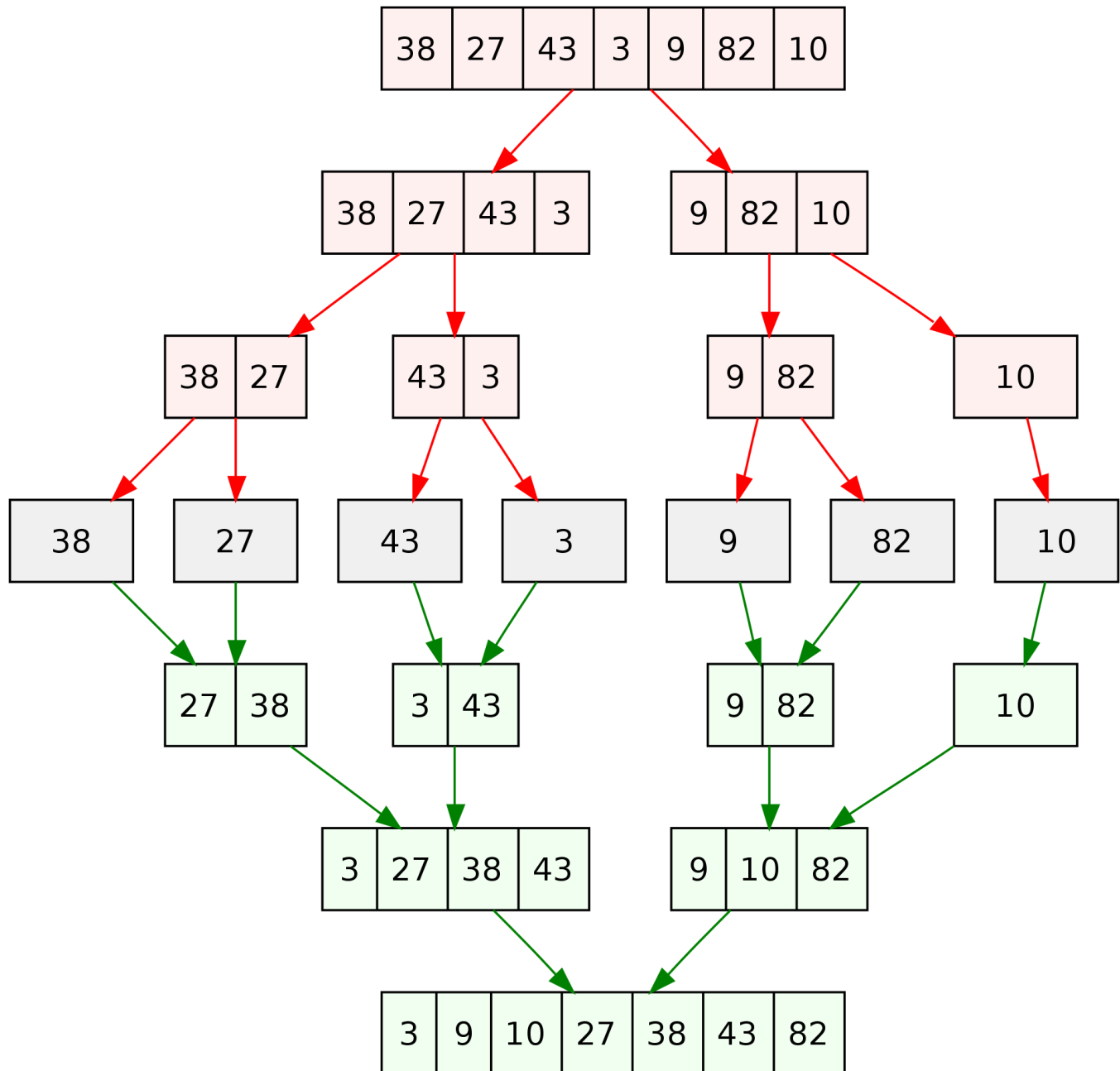
## Merge Sort

Merge Sort works by iteratively splitting a list into two sublists until each sublist has one element

Each sublist is represented by a binary tree

At each step a pair of sublists is successively merged into a list with the elements in increasing order. The process ends when all the sublists have been merged

The succession of merged lists is represented by a binary tree



## Recursive Merge Sort

Begin with the list of  $n$  elements  $L$ .

```

procedure mergesort($L = a_1, a_2, ..., a_n$)
  if $n > 1$ then
    $m := \lfloor n/2 \rfloor$
    $L_1 := a_1, a_2, ..., a_m$
    $L_2 := a_{m+1}, a_{m+2}, ..., a_n$
    $L := \text{merge}(\text{mergesort}(L_1), \text{mergesort}(L_2))$
  //L is now sorted into elements in increasing order

```

```

procedure merge(L1, L2 :sorted lists)

```

```

L := empty list

while L1 and L2 are both nonempty
    remove smaller of first elements of L1 and L2 from its list;
    put at the right end of L
    if this removal makes one list empty
        then remove all elements from the other list and append them to L

return L {L is the merged list with the elements in increasing order}

```

Complexity of Merge Sort: It can be proved (beyond the scope of this course) that the complexity of Merge Sort is  $O(n \log n)$ . In more advanced classes you will learn that the fastest comparison based sorting algorithms have  $O(n \log n)$  time complexity. So Merge Sort achieves the best possible big-O estimate of time complexity for a comparison based sorting algorithm

## Recursion and Iteration

Recursion	Iteration
Function calls itself, until the base case is executed, and the recursion terminates	Set of instructions is executed repeatedly, while a given condition is satisfied
More overhead; slower - Context switching for each recursive call	Less overhead; faster
Higher space complexity – A new stack frame for each invocation of the recursive call	Lower space complexity
Generally a recursive function is very simple – e.g. the recursive algorithm for the Towers of Hanoi problem is very simple and intuitive	Sometimes it is hard to follow the change in execution context from iteration to iteration – e.g. the iterative algorithm for the Towers of Hanoi problem is very complex

- In theory any problem that can be solved by a recursive algorithm can be solved by an iterative algorithm, and vice-versa
- Some problems can be better understood and resolved using recursion
- The easiest recursive algorithms to convert to iterative are those involving tail recursion i.e. recursive algorithms in which no statements are executed after the return from the recursive call

## The Basics of Counting

Combinatorics, the study of arrangements of objects, is an important part of discrete mathematics

This subject had its origin in the systematic study of gambling games

Combinatorics deals with ordered or unordered arrangements of the objects of a set with or without repetitions. These arrangements, called permutations and combinations

Combinatorics has many practical applications

(combination is order does not matter, XY and YX count as the same element)

(permutations is order does matter, XY and YX count as two different elements)

Generally, permutation will come out larger result

## Basic Counting Principles: The Product Rule

The Product Rule:

Suppose that a procedure can be broken down into a sequence of two tasks. If there are  $n_1$  ways to do the first task and for each of these ways of doing the first task, there are  $n_2$  ways to do the second task, then there are  $n_1 n_2$  ways to do the procedure

Example:

Assume that a procedure consists of three steps  $s_1, s_2, s_3$

There are 5 ways to perform  $s_1$ : A, B, C, D, E

There are 4 ways to perform  $s_2$ : 1, 2, 3, 4

There are 3 ways to perform  $s_3$ :  $\alpha, \beta, \gamma$

By the Product Rule there are  $5 \times 4 \times 3 = 60$  ways to perform the procedure, which are enumerated below

A-1- $\alpha$ , A-1- $\beta$ , A-1- $\gamma$ , A-2- $\alpha$ , A-2- $\beta$ , A-2- $\gamma$ , A-3- $\alpha$ , A-3- $\beta$ , A-3- $\gamma$ , A-4- $\alpha$ , A-4- $\beta$ , A-4- $\gamma$ , B-1- $\alpha$ , B-1- $\beta$ , B-1- $\gamma$ , B-2- $\alpha$ , B-2- $\beta$ , B-2- $\gamma$ , B-3- $\alpha$ , B-3- $\beta$ , B-3- $\gamma$ , B-4- $\alpha$ , B-4- $\beta$ , B-4- $\gamma$ , C-1- $\alpha$ , C-1- $\beta$ , C-1- $\gamma$ , C-2- $\alpha$ , C-2- $\beta$ , C-2- $\gamma$ , C-3- $\alpha$ , C-3- $\beta$ , C-3- $\gamma$ , C-4- $\alpha$ , C-4- $\beta$ , C-4- $\gamma$ , D-1- $\alpha$ , D-1- $\beta$ , D-1- $\gamma$ , D-2- $\alpha$ , D-2- $\beta$ , D-2- $\gamma$ , D-3- $\alpha$ , D-3- $\beta$ , D-3- $\gamma$ , D-4- $\alpha$ , D-4- $\beta$ , D-4- $\gamma$ , E-1- $\alpha$ , E-1- $\beta$ , E-1- $\gamma$ , E-2- $\alpha$ , E-2- $\beta$ , E-2- $\gamma$ , E-3- $\alpha$ , E-3- $\beta$ , E-3- $\gamma$ , E-4- $\alpha$ , E-4- $\beta$ , E-4- $\gamma$

Example:

How many bit strings of length seven are there?

Solution:

Since each of the seven bits is either a 0 or a 1 (i.e. 2 choices), the answer is  $2^7 = 128$ .

Example:

A new company with just two employees, Sanchez and Patel, rents a floor of a building with 12 offices. How many ways are there to assign different offices to these two employees?

Solution:

Assigning an office to Sanchez can be done in 12 ways, then assigning an office to Patel different from the office assigned to Sanchez can be done in 11 ways. So there are  $12 \cdot 11 = 132$  ways to assign offices to these two employees.

Example:



How many different license plates can be made if each plate contains a sequence of three uppercase English letters followed by three digits?

Solution:

By the product rule, there are  $26 \cdot 26 \cdot 26 \cdot 10 \cdot 10 \cdot 10 = 17,576,000$  different possible license plates.

## Refresher on Functions

Definition: Let  $A$  and  $B$  be nonempty sets. A function  $f$  from  $A$  to  $B$ , denoted  $f: A \rightarrow B$  is an assignment of each element of  $A$  to exactly one element of  $B$ . We write  $f(a)=b$  if  $b$  is unique element of  $B$  assigned by the function  $f$  to the element  $a$  of  $A$

## Counting Functions

How many functions are there from a set with  $m$  elements to a set with  $n$  elements?

Solutions:

Suppose the elements in the domain are  $a_1, a_2, \dots, a_m$ . There are  $n$  ways to choose the value of  $a_1$ ,  $n$  ways to choose the value of  $a_2$ , etc. The product rule tells us that there are  $n \cdot n \cdot \dots \cdot n = n^m$  such functions.

## Counting One-to-One functions

How many one-to-one functions are there from a set with  $m$  elements to a set with  $n$  elements

Solutions:

Suppose the elements in the domain are  $a_1, a_2, \dots, a_m$ , there are  $n$  ways to choose for value  $a_1$ , there are  $n-1$  ways to choose from  $a_2$ , etc.

The product rule tells us that there are  $n(n-1)(n-2)\dots(n-m+1)$  such functions

## Counting Subsets of a Finite Set

Use the product rule to show that the number of different subsets of a finite set  $S$  is  $2^{|S|}$

Solutions:

Let us assume that the elements of set  $S$  are  $a_1, a_2,$

$a_3, \dots, a_n$ . Let us represent the subsets of  $S$  by bit strings of length  $n$  where the  $i$ th bit of a bit string is 1 if  $a_i$  belongs to the corresponding subset and 0 if  $a_i$  does not belong to the corresponding subset

By the product rule, there are  $2^n$  such bit strings, and therefore  $2^n$  subsets

Since  $n = |S|$ , there are  $2^{|S|}$  subsets

## Basic Counting Principles: The Sum Rule

The Sum Rule:

If a task can be done either in one of  $n_1$  ways or in one of  $n_2$  ways, where none of the set of  $n_1$  ways is the same as any of the  $n_2$  ways, then there are  $n_1 + n_2$  ways to do the task.

If a task can be done in one of  $n_1$  ways or one of  $n_2$  ways, or one of  $n_m$  ways, and those ways have no overlapping, there are total of  $n_1 + n_2 + n_3 + \dots + n_m$  ways to do this task

## Sum Rule: An Illustrative Example

Alternative 1

– There are 5 ways to perform a task: designated as A, B, C, D, E

Alternative 2

– There are 4 ways to perform the task: designated as 1, 2, 3, 4

Alternative 3

– There are 3 ways to perform the task: designated as  $\alpha$ ,  $\beta$ ,  $\gamma$

By the Sum Rule the number of ways to perform the task is  $5 + 4 + 3 = 12$

Example:

The mathematics department must choose either a student or a faculty member as a representative for a university committee. How many choices are there for this representative if there are 37 members of the mathematics faculty and 83 mathematics majors and no

one is both a faculty member and a student.

Solution:

By the sum rule it follows that there are

$37 + 83 = 120$  possible ways to pick a representative

## Rule of Thumb for Applying the Product Rule or the Sum Rule

if we must make one choice and then another choice, the product rule applies

if we must make one choice or another choice, the sum rule applies

## Combining the sum and product rule

Example:

Suppose statement labels in a programming language can be either a single letter or a letter followed by a digit. Find the number of possible labels.

Solutions:

Number of choices when label is single letter: 26

Number of choice when label is single letter followed by a digit:  $10 * 26 = 260$  (product rule applied here)

total number of possible labels:  $26 + 260 = 286$  (sum rule applied here)

Example:

How many strings are there of lowercase letters of length four or less, not counting the empty string

Solution:

Number of string of length 1 =  $26$

Number of string of length 2 =  $26 * 26 = 26^2 = 676$

Number of string of length 3 =  $26 * 26 * 26 = 26^3 = 17576$

Number of string of length 4 =  $26 * 26 * 26 * 26 = 26^4 = 456976$

Total number of strings:  $26 + 676 + 17576 + 456976 = 475254$

There are 475254 strings of lowercase letters of length four or less, not counting the empty string

## Extra Example

Counting password:

Each user on a computer system has a password, which is six to eight characters long, and each character is an uppercase letter or a digit. Each password must have at least one digit. How many possible passwords are there?

Solution:

Let  $P$  be the total number of passwords, and let  $P_6$ ,  $P_7$ , and  $P_8$  be the passwords of length of 6, 7, and 8.

for each  $P$ , we find it by the total number of passwords with letter or number minus the total number of passwords with only letters

$P_6 = (10+26)^6 - 26^6 = 2176782336 - 308915776 = 1867866560$

$P_7 = (10+26)^7 - 26^7 = 78364164096 - 8031810176 = 70332353920$

$P_8 = (10+26)^8 - 26^8 = 2821109907456 - 208827064576 = 2612282842880$

Consequently,  $P = P_6 + P_7 + P_8 = 2684483063360$

## Basic Counting Principles: Subtraction Rule

If a task can be done either in one of  $n_1$  ways or in one of  $n_2$  ways, then the total number of ways to do the task is  $n_1 + n_2$  minus the number of ways to do the task that are common to the two different ways.

$$|A \cup B| = |A| + |B| - |A \cap B|$$

Example:

How many bit strings of length eight either start with a 1 bit or end with the two bits 00?

Solution:

A byte is 8 bit

If first bit is 1, then there are 7 free bits, and each bit can either be 0 or 1 which is 2 options:

$$2^7 = 128$$

If last two bits are 00, there are 6 free bits, and each bits can be either be 0 or 1 which is 2 options:

$$2^6 = 64$$

if first bit is 1 and last two bits are 00, there are 5 free bits, and each bits can either be 0 or 1 which is 2 options:

$$2^5 = 32$$

first bit 1 and last two bits 00 was counted twice in union of first bit a and last two bits 00

$$2^7 + 2^6 - 2^5 = 128 + 64 - 32 = 160$$

There are total of 160 strings either start with a 1 or end with 00

## Counting Patterns

How many seven digit numbers do not contain the substring 1234?

Solution:

number of seven digit numbers:  $10^7$

Number of substring that contains 1234

1234 \_ \_ \_ :  $1 * 10^3$  choices

\_ 1234 \_ \_ :  $1 * 10^3$  choices

\_ \_ 1234 \_ :  $1 * 10^3$  choices

\_ \_ \_ 1234 :  $1 * 10^3$  choices

Total of substrings that contains 1234:  $4 * 10^4$

Total of substrings that does not contains 1234:

$$10^7 - (4 * 10^3) = 9996000$$

## Basic Counting Principles: Division Rule

There are  $n/d$  ways to do a task if it can be done using a procedure that can be carried out in  $n$  ways, and for every way  $w$ , there are  $d$  equivalent ways of doing  $w$

Restated in terms of sets: If the finite set  $A$  is the union of  $n$  pairwise disjoint subsets each with  $d$  elements, then  $n = |A|/d$ . In terms of functions: If  $f$  is a function from  $A$  to  $B$ , where both are finite sets, and for every value  $y \in B$  there are exactly  $d$  values  $x \in A$  such that  $f(x) = y$ , then  $|B| = |A|/d$ .

The Division Rule is commonly used for counting arrangements when some of the objects are indistinguishable or when order doesn't matter.

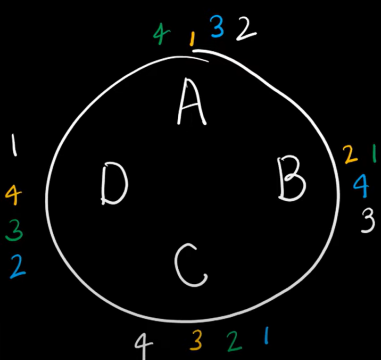
## Division Rule: An Illustrative Example

How many way there are to assign 4 people to 4 seats if the pattern are considered the same if a person neighbors are the same

Discrete Math II - 6.1.3 The Subtraction and Division Rules

Reminder: There are  $\frac{n}{d}$  ways to do a task if it can be done using a procedure that can be carried out in  $n$  ways, where there are  $d$  corresponding outcomes per group.

How many ways are there to seat four people around a circular table, where two "seatings" are considered the same when each person has the same left and right neighbor?



is your solution which is of course six

$4 \cdot 3 \cdot 2 \cdot 1 = 24$

$\frac{24}{4} = 6$

13:41 / 13:56 • The Division Rule Example >

$$4 \cdot 3 \cdot 2 \cdot 1 = 24 \text{ ways}$$

However, assigning person 1 at seat 1, 2, 3, 4 may has same pattern

$$24 \text{ ways} / 4 \text{ seats} = 6 \text{ ways (total of 6 different pattern)}$$

result = number of possibilities / number of elements in each subset

## Six poeple's seats

Example:

How many ways are there to seat six people around a circular table where two seatings are considered the same when everyone has the same two neighbors without regard to whether they are right or left neighbors?

Solutions:

There are  $6! = 720$  ways to order 6 people. this overcounts the seating arrangements by a multiple of 6, corresponding to rotating the table. Furthermore, we are overcounting by a multiple of 2 because seating arrangements that reverse the right and the left neighbor are equivalent.

$$720/6/2 = 60 \text{ ways}$$

## Tree diagrams

We can solve many counting problems through the use of tree diagrams, where a branch represents a possible choice and the leaves represent possible outcomes

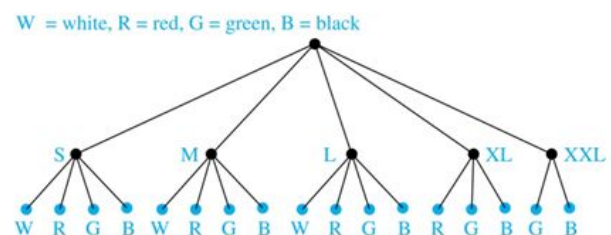
Example:

Suppose that a T-shirt come in five different size : S, M, L, XL, and XXL. Each size comes with four colors(white, red, green, and black) except XL comes with three (red, green, and black) and XXL which comes with two (green and black).

What is the minimum number of shirts that the campus books store needs to stock to have one of each size and color available?

## Tree Diagrams

- **Tree Diagrams:** We can solve many counting problems through the use of *tree diagrams*, where a branch represents a possible choice and the leaves represent possible outcomes.
- **Example:** Suppose that “I Love Discrete Math” T-shirts come in five different sizes: S,M,L,XL, and XXL. Each size comes in four colors (white, red, green, and black), except XL, which comes only in red, green, and black, and XXL, which comes only in green and black. What is the minimum number of stores that the campus book store needs to stock to have one of each size and color available?
- **Solution:** Draw the tree diagram.



- The store must stock 17 T-shirts.

Alternative solution:

$$4S + 4M + 4L + 3XL + 2XXL = 17 \text{ types of shirts}$$

Example:

How many bit strings of length four do not have two consecutive 1s?

**Solution:**

The tree diagram shows that there are 8 bit strings of length four that do not have two consecutive 1s

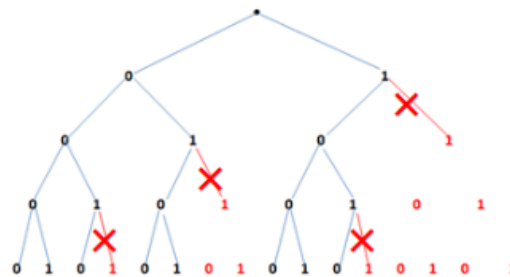
- 0000
- 0001
- 0010
- 0100
- 0101
- 1000
- 1001
- 1010

# Tree Diagrams<sub>2</sub>

**Example:** How many bit strings of length four do not have two consecutive 1s?

**Solution:** The tree diagram shows that there are 8 bit strings of length four that do not have two consecutive 1s

- 0000
- 0001
- 0010
- 0100
- 0101
- 1000
- 1001
- 1010



**Note:** Tree diagrams are most feasible when the number of leaves in the tree is small.

# The Pigeonhole Principle

If a flock of 20 pigeons roosts in a set of 19 pigeonholes, one of the pigeonholes must have more than 1 pigeon.

Also known as the Dirichlet Drawer Principle

Defination:

Pigeonhole Principle: If  $k$  is a positive integer and  $k + 1$  or more objects are placed into  $k$  boxes, then at least one box contains two or more objects.

## Corollary 1:

A function  $f$  from a set with  $k + 1$  or more elements to a set with  $k$  elements is not one-to-one

## Example:

Among any group of 367 people, there must be at least two with the same birthday, because there are only 366 possible birthdays.

## The Generalized Pigeonhole Principle

If  $N$  objects are placed into  $k$  boxes, then there is at least one box containing at least  $\lceil N/k \rceil$  objects

## Example:

Among 100 people there are at least  $\lceil 100/12 \rceil = 9$  who were born in the same month.

Typically with the Generalized Pigeonhole Principle we look for the smallest number that will guarantee an outcome

## Problem:

What is the minimum number of students required in a

discrete mathematics class to be sure that at least six will receive the same grade, if there are five possible grades, A, B, C, D, and F?

## Solution:

The minimum number of students needed to ensure that

at least six students receive the same grade is the smallest integer  $N$  such that  $\lceil N/5 \rceil = 6$ . The smallest such integer is  $N = [(6 - 1) * 5] + 1 = 26$ .

## Draw cards

How many cards must be selected from a standard deck of 52 cards to guarantee that at least three cards of the same suit are chosen?

We assume four boxes; one for each suit. Using the Generalized Pigeonhole Principle, at least one box contains at least  $\lceil N/4 \rceil$  cards. At least three cards of one suit are selected if  $\lceil N/4 \rceil \geq 3$ . The smallest integer  $N$  such that  $\lceil N/4 \rceil \geq 3$  is  $N = [(3 - 1) * 4] + 1 = 9$

9 cards at least need to be drawn to have three cards in same suit

How many must be selected to guarantee that at least three hearts are selected?

A deck has 13 hearts and 39 non-hearts cards. So, if we select  $39 + 3$  cards, we must have at least three hearts cards in it.

## Phone number



What is the least number of area codes needed to guarantee that the 25 million phones in a state can be assigned distinct 10-digit telephone numbers? Assume that telephone numbers are of the form NXX-NXX-XXXX, where the first three digits form the area code, N represents a digit from 2 to 9 inclusive, and X represents any digit.

Solution:

Total numbers of phone numbers without area code =  $8 \times 10^6$

Since we have to provide 25 million phone numbers, the state will need a minimum of  $\lceil 25 \times 10^6 / 8 \times 10^6 \rceil = 4$  area codes

## Permutations and Combinations

Many counting problems deal with finding the number of ways to arrange a specified number of elements of a set, without actually listing them

There are two distinct methods that can be used to select  $r$  objects from a set of  $n$  elements: ordered and unordered

in an ordered selection, it is not only what elements are chosen but also the order in which they are chosen that matters

- an ordered selection of  $r$  elements from a set of  $n$  elements is called an  $r$ -permutation of the set

in an unordered selection it is only the identify of the chosen elements that matters while order is immaterial

- an unordered selection of  $r$  elements from a set of  $n$  elements in called the  $r$ -combination of the set

The product rule is a generic rule that allows us to count the number of ways to complete any procedure that is composed of multiple tasks

Permutations and combinations make heavy use of the product rule to count the number of ordered and unordered selections, of a desired size, that can be made from a given set

## Permutations

defination:

A permutation of a set of distinct objects is an ordered arrangement of these objects. An ordered arrangement of  $r$  elements of a set is called an  $r$ -permutation.

Human word:

the number of ways to select  $m$  elements from  $n$  elements while the order does matter

Example:

Let  $S = \{1, 2, 3\}$

The number of  $r$ -permutations of a set with  $n$  elements is denoted by  $P(n, r)$

$P(3, 2)$  = number of ways to pick 2 elements from 3 elements while the order does matter

In this case,  $P(3, 2)$  for  $S$  is  $\{1, 2\}, \{1, 3\}, \{2, 1\}, \{2, 3\}, \{3, 1\}, \{3, 2\}$ , which is 6 ways. So  $P(3, 2) = 6$

## Formula for the number of permutations

Theorem 1: If  $n$  is a positive integer and  $r$  is an integer with  $1 \leq r \leq n$ , then there are

$$P(n, r) = n(n-1)(n-2)\dots(n-r+1)$$

$r$ -permutations of a set with  $n$  distinct elements

If  $n$  and  $r$  are integers with  $1 \leq r \leq n$ , then

$$P(n, r) = \frac{n!}{(n-r)!}$$

$$P(n, n) = n!$$

$$P(n, 0) = 1$$

$$P(n, 1) = n$$

## Solving Counting Problems by Counting Permutations

Example:

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

Solution:

$$P(100, 3) = 100 \cdot 99 \cdot 98 = 970200 \text{ ways}$$

Example:

How many permutations of the letters ABCDEFGH contain the string ABC?

Solution:

Because the letters ABC must occur as a block, we can consider them to be a single object. The problem now reduces to finding the number of permutations of six objects viz. ABC, D, E, F, G, and H. So the answer is  $P(6, 6) = 6! = 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 720$

## Combinations

Definition: An  $r$ -combination of elements of a set is an unordered selection of  $r$  elements from the set. Thus, an  $r$ -combination is simply a subset of the set with  $r$  elements.

The number of  $r$ -combinations of a set with  $n$  distinct elements is denoted by  $C(n, r)$ . The notation  $\binom{n}{r}$  is also called a binomial coefficient

Example:

Let  $S$  be the set  $\{a, b, c, d\}$ . Then  $\{a, c, d\}$  is a 3-combination from  $S$ . It is the same as  $\{d, c, a\}$  since the order listed does not matter.

$C(4, 2) = 6$  because the 2-combinations of  $\{a, b, c, d\}$  are the six subsets  $\{a, b\}$ ,  $\{a, c\}$ ,  $\{a, d\}$ ,  $\{b, c\}$ ,  $\{b, d\}$ , and  $\{c, d\}$ .

Theorem 2: The number of  $r$ -combinations of a set with  $n$  elements, where  $n \geq r \geq 0$ , equals

$$C(n, r) = \frac{n!}{(n-r)!r!}$$

Corollary 2:

Let  $n$  and  $r$  be nonnegative integers with  $r \leq n$ . Then  $C(n, r) = C(n, n - r)$ .

$$C(n, r) = \frac{n!}{(n-r)!r!}$$

$$C(n, n) = 1$$

$$C(n, 0) = 1$$

$$C(n, 1) = n$$

Example:

How many poker hands of five cards can be dealt from a standard deck of 52 cards?

Also, how many ways are there to select 47 cards from a deck of 52 cards?

Solution:

Since the order in which the cards are dealt does not matter, the number of five card hands is:

$$C(52, 5) = C(52, 47) = 2598960$$

## Combination problems

Problem:

How many bit strings of length 10 contain

- a) exactly four 1s?
- b) at most four 1s?
- c) at least four 1s?
- d) an equal number of 0s and 1s?

Solution:

$$\text{a) } C(10, 4) = 210$$

$$\text{b) } C(10, 0) + C(10, 1) + C(10, 2) + C(10, 3) + C(10, 4) = 1 + 10 + 45 + 120 + 210 = 386$$

$$\text{c) } C(10, 4) + C(10, 5) + C(10, 6) + C(10, 7) + C(10, 8) + C(10, 9) + C(10, 10) = 210 + 252 + 210 + 120 + 45 + 10 + 1 = 848$$

$$\text{d) } C(10, 5) = 252$$

Problem:

Suppose that there are 9 faculty members in the mathematics department and 11 in the computer science department. How many ways are there to select a committee to develop a discrete mathematics course at a school if the committee is to consist of three faculty members from the mathematics department and four from the computer science department?

Solution:

Number of ways to select 3 faculty members from the mathematics department =  $C(9, 3)$

Number of ways to select 4 faculty members from the computer science department =  $C(11, 4)$

By product rule the total ways to select the committee is  $C(9, 3) * C(11, 4) = 84 * 330 = 27720$

## Relations and Their Properties

### Binary Relations

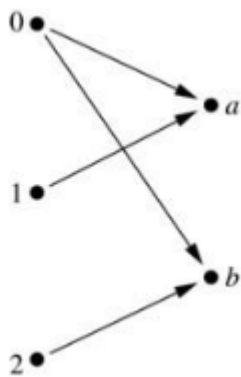
Definition: A binary relation  $R$  from a set  $A$  to a set  $B$  is a subset of the Cartesian product of  $A$  and  $B$

$$R \subseteq A \times B$$

We use the notation  $a R b$  to denote that  $(a, b) \in R$  and  $a \not R b$  to denote that  $(a, b) \notin R$

Example:

Let  $A = \{0, 1, 2\}$  and  $B = \{a, b\}$

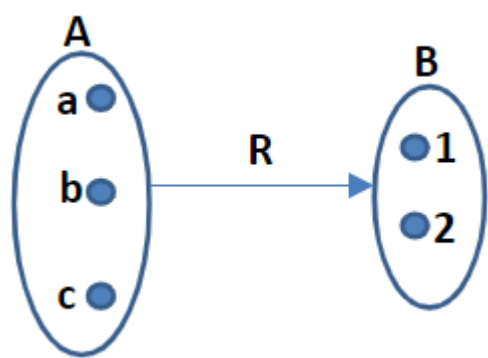


$\{(0, a), (0, b), (1, a), (2, b)\}$  is the relation from  $A$  to  $B$

Instead of drawing arrows, we listed each arrangement and combined them, call the total set as the relation

---

Assume we have an relationship from  $A$  to  $B$



In this case, there are  $3 \times 2$  elements in  $A \times B$ , and  $2^6=64$  subsets of  $A \times B$

In this instance there are 64 relations between  $A$  and  $B$

## Binary Relations Example

Example:

Let  $A$  be the set of cities in the U. S. A., and

let  $B$  be the set of 50 states in the U. S. A. Define relation  $R$  as follows

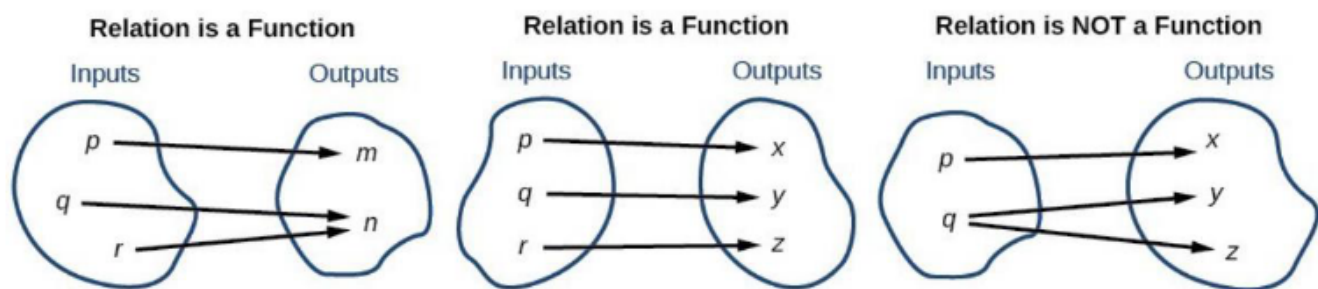
$$R = \{(a, b) \mid a \in A, b \in B \wedge a \text{ is in state } b\}$$

Then (Boulder, Colorado), (Bangor, Maine), (Ann Arbor, Michigan), (Middletown, New Jersey), (Middletown, New York), (Cupertino, California), and (Red Bank, New Jersey) are in  $R$

## Functions as Relations

A function  $f$  from a set  $A$  to a set  $B$  assigns exactly one element of  $B$  to each element of  $A$

Relations are generalizations of functions and they can be used to express a much wider class of relationships between sets

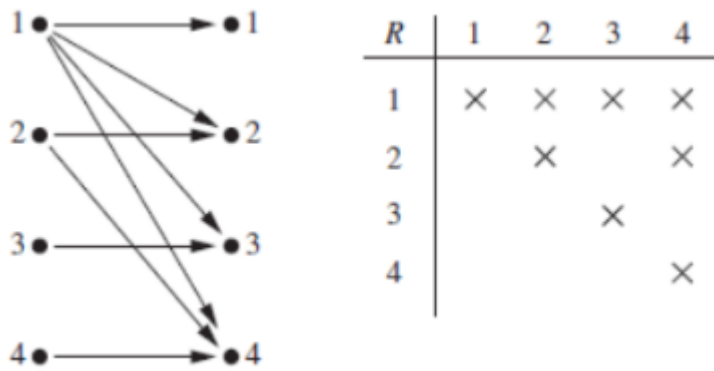


## Binary Relations on a Set

A binary relation  $R$  on a set  $A$  is a subset of  $A \times A$  or a relation from  $A$  to  $A$

Suppose that  $A = \{a, b, c\}$ . Then  $R = \{(a, a), (a, b), (a, c)\}$  is a relation on  $A$ .

Let  $A = \{1, 2, 3, 4\}$ . The ordered pairs in the relation  $R = \{(a, b) \mid a \text{ divides } b\}$  are  $(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 4), (3, 3)$ , and  $(4, 4)$



## Number of Binary Relations on a Set

Question:

How many relations are there on a set  $A$

Solution:

Because a relation on  $A$  is a subset of  $A \times A$ , we count the subsets of  $A \times A$ . Since  $A \times A$  has  $n^2$  elements when  $A$  has  $n$  elements, and a set with  $m$  elements has  $2^m$  subsets, therefore there are  $2^{n^2}$  relations on a set  $A$ .

## Types of relations

Let  $R$  be a relation on a set  $A$ . There are several properties that can be used to classify  $R$

$R$  can be classified as:

- reflexive
- symmetric
- antisymmetric
- transitive

When classifying a relation based on a property keep in mind that:

- there should be no violation of the pertinent property
- there may be additional elements in the relation that do not satisfy the pertinent property
- e.g. there may be additional elements in a reflexive relation that do not satisfy the reflexive property

## Reflexive Relations

A relation  $R$  on a set  $A$  is reflexive if  $(a, a) \in R$  for every element  $a \in A$ . In terms of quantifiers,  $\forall a((a, a) \in R)$ , where the universe of discourse is the set of all elements in  $A$

- Each element in set  $A$  is related to itself

Human word: you will find all elements  $x$  in  $A$  in form of  $(x, x)$  in this relation set

Example:

$\{(1, 1), (1, 2), (1, 4), (2, 1), (2, 2), (3, 3), (4, 1), (4, 4)\}$

(this contains all pairs of  $(a, a)$  in set  $\{1, 2, 3, 4\}$ )

Divides relation is reflexive because  $a \mid a$  for every  $a$  except 0

## Symmetric Relations

A relation  $R$  on a set  $A$  is called symmetric if  $(b, a) \in R$  whenever  $(a, b) \in R$ , for all  $a, b \in A$ . In terms of quantifiers:  $\forall a \forall b ((a, b) \in R \rightarrow (b, a) \in R)$

- An element related to a second element implies the second element is also related to the first element.

Human word: for every  $(x, y)$  in this relation set, you will always being able to find a  $(y, x)$ , or you will find if  $x == y$

Example:

$\{(1, 1), (1, 2), (2, 1)\}$

Divides relation is not Symmetric Relations because  $a \mid b$  does not implies  $b \mid a$

## Antisymmetric Relations

A relation  $R$  on a set  $A$  such that for all  $a, b \in A$ , if  $(a, b) \in R$  and  $(b, a) \in R$ , then  $a = b$  is called antisymmetric. In terms of quantifiers,  $\forall a \forall b (((a, b) \in R \wedge (b, a) \in R) \rightarrow (a = b))$

- there are no pairs of distinct elements  $a$  and  $b$  with  $a$  related to  $b$  and  $b$  related to  $a$
- for example,  $a \leq b$  and  $b \leq a$  implies that  $a = b$

Human word: this is just like combination, if  $(a, b)$  appeared then there should no  $(b, a)$  exist in relation set

Example:

$\{(2, 1), (3, 1), (3, 2), (4, 1), (4, 2), (4, 3)\}$

Divides relation is antisymmetric relations

## Transitive Relations

A relation  $R$  on a set  $A$  is called transitive if whenever  $(a, b) \in R$  and  $(b, c) \in R$ , then  $(a, c) \in R$ , for all  $a, b, c \in A$ . In terms of quantifiers:  $\forall a \forall b \forall c (((a, b) \in R \wedge (b, c) \in R) \rightarrow (a, c) \in R)$ .

- If the first element is related to the second element, and the second element is related to the third element, then the first element must be related to the third element

Human word: if you have  $(a, b)$  and  $(b, c)$ , then you must also have  $(a, c)$  in relation set

Example:

$\{(2, 1), (3, 1), (3, 2), (4, 1), (4, 2), (4, 3)\}$

Explanation:

$(3, 2), (2, 1) \rightarrow (3, 1)$

$(4, 2), (2, 1) \rightarrow (4, 1)$

$(4, 3), (3, 2) \rightarrow (4, 2)$

Divides relation is transitive relations

## Example question

Let  $R$  be the following relation defined on the set  $\{a, b, c, d\}$

$R = \{(a, a), (a, c), (a, d), (b, a), (b, b), (b, c), (b, d), (c, b), (c, c), (d, b), (d, d)\}$

Determine whether  $R$  is: (a) reflexive. (b) symmetric (c) antisymmetric (d) transitive

Solution:

- $R$  is reflexive because  $R$  contains  $(a, a), (b, b), (c, c)$  and  $(d, d)$
- $R$  is not symmetric because  $(a, c) \in R$  but  $(c, a) \notin R$
- $R$  is not antisymmetric because both  $(b, c) \in R$  and  $(c, b) \in R$  but  $b \neq c$
- $R$  is not transitive because, for example,  $(a, c) \in R$  and  $(c, b) \in R$  but  $(a, b) \notin R$

## Equivalence Relations

A relation on a set  $A$  is called an equivalence relation if it is reflexive, symmetric, and transitive

Two elements  $a$ , and  $b$  that are related by an equivalence relation are called equivalent. The notation  $a \sim b$  is often used to denote that  $a$  and  $b$  are equivalent elements with respect to a particular equivalence relation

## Congruence Modulo $m$

$R = \{(a, b) \mid a \equiv b \pmod{m}\}$  for  $m > 1$

- Reflexivity:  $a \equiv a \pmod{m}$  since  $a - a = 0$  is divisible by  $m$  since  $0 = 0 \cdot m$ .
- Symmetry: Suppose that  $a \equiv b \pmod{m}$ . Then  $a - b$  is divisible by  $m$ , and so  $a - b = km$ , where  $k$  is an integer. It follows that  $b - a = (-k)m$ , or  $b = a + (-k)m$  so  $b \equiv a \pmod{m}$ .
- Transitivity: Suppose that  $a \equiv b \pmod{m}$  and  $b \equiv c \pmod{m}$ . Then  $m$  divides both  $a - b$  and  $b - c$ . Hence, there are integers  $k$  and  $l$  with  $a - b = km$  and  $b - c = lm$ . We obtain by adding the equations:  $a - c = (a - b) + (b - c) = km + lm = (k + l)m$ .

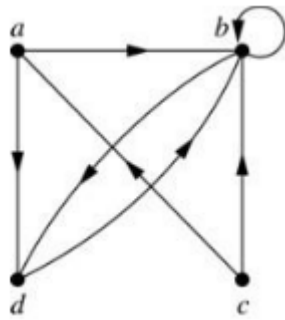
## Divides

Divides is not equivalent because it is not symmetry

## Representing Relations Using Digraphs

- Relations with a manageable number of elements can be represented pictorially, using digraphs





## Determining which Properties a Relation has from its Digraph

**Reflexivity:** A loop must be present at all vertices in the graph.



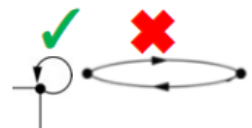
**Symmetry:** If  $(x,y)$  is an edge, then so is  $(y,x)$ .

- All edges either loops or “anti-parallel”



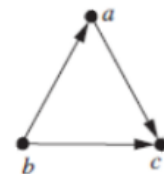
**Antisymmetry:** If  $(x,y)$  with  $x \neq y$  is an edge, then  $(y,x)$  is not an edge.

- If no “anti-parallel” edges then no violation of anti-symmetry



**Transitivity:** If  $(x,y)$  and  $(y,z)$  are edges, then so is  $(x,z)$ .

- All path of length 2 (e.g. b-a-c) accompanied by a corresponding path of length 1 (e.g. b-c)



## Combining Relations

Given two relations  $R_1$  and  $R_2$ , we can combine them using basic set operations to form new relations such as  $R_1 \cup R_2$ ,  $R_1 \cap R_2$ ,  $R_1 - R_2$ , and  $R_2 - R_1$

Example:

Let  $A = \{1, 2, 3\}$  and  $B = \{1, 2, 3, 4\}$ . The relations  $R_1 = \{(1, 1), (2, 2), (3, 3)\}$  and  $R_2 = \{(1, 1), (1, 2), (1, 3), (1, 4)\}$  can be combined using basic set operations to form new relations

$$R_1 \cup R_2 = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (3, 3)\}$$

$$R_1 \cap R_2 = \{(1, 1)\}$$

$$R_1 - R_2 = \{(2, 2), (3, 3)\}$$

$$R_2 - R_1 = \{(1, 2), (1, 3), (1, 4)\}$$

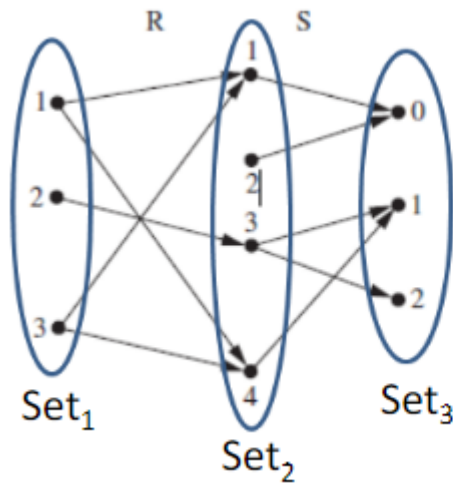
## Composition

Suppose

- $R_1$  is a relation from a set A to a set B.
- $R_2$  is a relation from B to a set C.

Then the composition (or composite) of  $R_2$  with  $R_1$ , is a relation from A to C where

if  $(x, y)$  is a member of  $R_1$  and  $(y, z)$  is a member of  $R_2$ , then  $(x, z)$  is a member of  $R_2 \circ R_1$



$1 \rightarrow 1 \rightarrow 0$	$(1, 0)$
$1 \rightarrow 4 \rightarrow 1$	$(1, 1)$
$2 \rightarrow 3 \rightarrow 1$	$(2, 1)$
$2 \rightarrow 3 \rightarrow 2$	$(2, 2)$
$3 \rightarrow 1 \rightarrow 0$	$(3, 0)$
$3 \rightarrow 4 \rightarrow 1$	$(3, 1)$