Course Outline

- Logic and Proofs
- Sets and Functions
- Sequences and Series
- Relations
- Number Theory
- Proof techniques
- Induction and Recursion
- Counting and Recurrence Relations
- Graphs
- Trees

TEXT Book:

- Discrete Mathematics & its Applications, 8th edition. By Kenneth H. Rosen.

References Book:

- Invitation to Discrete Maths, 2nd edition. By Matousek and Nesetril.
- Discrete Mathematics. By Lovasz, Pelikan and Vesztergombi.

The Foundations: Logic and Proofs

Chapter 1, Part I: Propositional Logic

Mr. Shoaib Raza

Discrete Mathematics

Discrete mathematics is the part of mathematics devoted to the study of discrete objects (Kenneth H. Rosen, 8th edition).

- How to think (and argument) mathematically.
- Learn mathematical facts and their applications.
- **Discrete mathematics** is the mathematical study of properties, and relationships among discrete objects.
- **Discrete mathematics** is the study of mathematical **structures** that are fundamentally discrete rather than

Discrete Vs Continuous

- Continuous Data: A set of data is said to be continuous if the values belonging to the set can take on any value within a finite or infinite interval.
- Continuous data is information that can be measured on a continuum or scale. , e.g., [0, 70].
- Continuous data can have almost any numeric value and can be meaningfully subdivided into finer and finer increments, depending upon the precision of the measurement system.

- Discrete Data: A set of data is said to be discrete if the values belonging to the set are distinct and separate. It is counted e.g., {1,2,3,4,5,6}

Discrete vs Continuous

- Examples of discrete Data
  - Number of boys in the class.
  - Number of candies in a packet.
  - Number of suitcases lost by an airline.
- Examples of continuous Data
  - Height of a person.
  - Time in a race.
  - Distance traveled by a car.

Why DM?

- Digital computers are based on discrete "atoms" (bits).
- Therefore, both a computer's
  - structure (circuits) and
  - operations (execution of algorithms)

  can be described by discrete mathematics.

Applications

- How many ways are there to choose a valid password on a computer system?
- What is the probability of winning a lottery?
- Is there a link between two computers in a network?
- How can I identify spam e-mail messages?
- How can I encrypt a message so that no unintended recipient can read it?
- What is the shortest path between two cities using a transportation system?
- How can a list of integers be sorted so that the integers are in increasing order?
- How many steps are required to do such a sorting?
- How can it be proved that a sorting algorithm correctly sorts a list?
- How can a circuit that adds two integers be designed?
- How many valid Internet addresses are there?

Course Description:

- This class teaches students
  - how to think logically and mathematically.
  - It stresses mathematical reasoning and different ways to solve problems.

  It enhance your ability to formulate and solve applied problems, to analyze and interpret algorithms and functions and to use them effectively.

- Five important themes are interwoven in this class:
  - mathematical reasoning,
  - combinatorial analysis,
  - discrete structures,
  - algorithmic thinking,
  - applications and modeling

Applications: Logic

Hardware and software specifications

One-bit Full Adder with

Carry-In and Carry-Out

Formal: Input_wire_A value in {0, 1}

Example 1: Adder

4-bit full adder

Example 2: System Specification:

- The router can send packets to the edge system only if it supports the new address space.
- For the router to support the new address space it's necessary that the latest software release be installed.
- The router can send packets to the edge system if the latest software release is installed.
- The router does not support the new address space.

to write these specifications in a rigorous / formal way? **Use Logic**

Applications: Number Theory
RSA and Public-key Cryptography

Alice and Bob have never met but they would like exchange a message. Eve would like to eavesdrop

E.g. between you and the Bank of America.

They could come up with a good encryption algorithm and exchange the encryption key – but how to do it without
Eve getting it? (If Eve gets it, all security
is lost.)

CS folks found the solution: *public key encryption*. Quite remarkab that is feasible.

Applications: Coloring a Map

How to color this map so that no two adjacent regions have the same color?

Applications: Graph representation

Coloring the nodes of the graph:
What's the minimum number of colors such that any two nodes connected by an edge have different colors?

Logic

- Logic is fundamental because it allows us to understand meaning of statements, deduce information about mathematical structures and uncover further structures.

- The rules of logic specify the meaning of mathematical statements.

- These rules are used to distinguish between valid and invalid arguments.

Propositional Logic

Introduction

- A **proposition** is a declarative sentence (a sentence that declares a fact) that is either true or false, but not both.
- Are the following sentences propositions?
  - Islamabad is the capital of Pakistan. (Yes)
  - Read this carefully. (No)
  - 1+2=3 (Yes)
  - x+1=2 (No)
  - What time is it? (No)

Proposition

Definition

proposition (or statement):
a declarative sentence that is either true or false

- law of the excluded middle:
  a proposition cannot be partially true or partially false
- law of contradiction:
  a proposition cannot be both true and false

propositions

- The Moon revolves around the Earth.
- Elephants can fly.
- $3 + 8 = 11$

not propositions

- What time is it?
- Exterminate!
- $x < 43$

Examples: Propositions

Is the following sentence a proposition? If it is a proposition, determine whether it is true or false.

Islamabad is the capital of Pakistan.

This makes a declarative statement, and hence is a proposition. The proposition is TRUE (T).

Can Ali come with you?.

This is a question not the declarative sentence and hence not a proposition.

Take two aspirins.

This is an imperative sentence not the declarative sentence and therefore not a proposition.

$x + 4 > 9$.

Because this is true for certain values of x (such as $x = 6$) and false for other values of x (such as $x = 5$), it is not a proposition.

He is a college student.

Because truth or falsity of this proposition depend on the reference for the pronoun *he*. it is not a proposition.

Activity 1

Write down at least 5 examples of propositions and

Non-propositions.

Propositional Variable

- propositional variable:
  a name that represents the proposition

examples

- p_1: The Moon revolves around the Earth. (T)
- p_2: Elephants can fly. (F)
- p_3: $3 + 8 = 11$ (T)

Notations

- The small letters are commonly used to denote the propositional variables, that is, variables that represent propositions, such as, p, q, r, s, ...

- The truth value of a proposition is true, denoted by T or 1, if it is a true proposition and false, denoted by F or 0, if it is a false proposition.

Compound Propositions

Logical operators are used to form new propositions also called compound propositions from two or more existing propositions.

- compound propositions are obtained by applying logical operators

  The logical operators are also called connectives.

- truth table:
  a table that lists the truth value of the compound proposition for all possible values of its variables

Propositional Logic – the area of logic that deals with propositions

1. Negation:

DEFINITION 1:

Let p be a proposition. The negation of p, denoted by $\neg p$, is the statement "It is not the case that p."

The proposition $\neg p$ is read "not p." The truth value of the negation of p, $\neg p$ is the opposite of the truth value of p.

- Examples
  - Find the negation of the proposition "Today is Friday." and express this in simple English.

    Solution: The negation is "It is not the case that *today is Friday*."

    In simple English, "Today is not Friday." or "It is not Friday today."
  - Find the negation of the proposition "At least 10 mm of rain fell today in Karachi." and express this in simple English.

    Solution: The negation is "It is not the case that *at least 10 mm of rain fell today in Karachi*."

    In simple English, "Less than 10 mm of rain fell today in Karachi."

Negation:

- Note: Always assume fixed times, fixed places, and particular people unless otherwise noted.
- Truth table:

The Truth Table for the Negation of a Proposition.	
p	$\neg p$
T	F
F	T

examples

- $\neg p_1$: The Moon does not revolve around the Earth.
  $\neg T : F$
- $\neg p_2$: Elephants cannot fly.
  $\neg F : T$

2. Conjunction:

DEFINITION 2

Let p and q be propositions. The *conjunction* of p and q, denoted by $p \wedge q$, is the proposition "p and q". The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

Examples

- Find the conjunction of the propositions p and q where p is the proposition "Today is Friday." and q is the proposition "It is raining today.", and the truth value of the conjunction.

Solution: The conjunction is the proposition "Today is Friday and it is raining today." The proposition is true on

$p \wedge q$

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

examples

- $p_1 \wedge p_2$: The Moon revolves around the Earth and elephants can fly.
  $T \wedge F : F$
- $p_1 \wedge p_3$: The Moon revolves around the Earth and $3 + 8 = 11$.
  $T \wedge T : T$

3. Disjunction:

DEFINITION 3

Let p and q be propositions. The *disjunction* of p and q, denoted by $p \vee q$, is the proposition "p or q". The *disjunction* $p \vee q$ is false when both p and q are false and is true otherwise.

- Note:

  inclusive or : The disjunction is true when at least one of the two propositions is true.

  - E.g. "Students who have taken calculus or computer science can take this class." – those who take one or both classes.

  exclusive or : The disjunction is true only when one of the proposition is true.

  - E.g. "Students who have taken calculus or computer science, but not both, can take this class." – only those who take one of them.
- Definition 3 uses *inclusive or*.

Disjunction (OR)

$p \vee q$

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

example

- $p_1 \vee p_2$: The Moon revolves around the Earth or elephants can fly.
  $T \vee F : T$

4. Exclusive OR:

DEFINITION 4

Let p and q be propositions. The *exclusive or* of p and q, denoted by $p \oplus q$, is the proposition that is true when exactly one of p and q is true and is false otherwise.

The Truth Table for the Conjunction of Two Propositions.

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

The Truth Table for the Disjunction of Two Propositions.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

The Truth Table for the Exclusive *Or* (*XOR*) of Two Propositions.

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Examples

1. Find the *exclusive or* of the propositions p and q, where

p : Atif will pass the course CSC102.

q : Atif will fail the course CSC102.

The *exclusive or* is

$p \oplus q$: Atif will pass or fail the course CSC102.

The following proposition uses the (English) connective "or". Determine from the context whether "or" is intended to be used in the inclusive or exclusive sense.

1. "Nabeel has one or two brothers".

A person cannot have both one and two brothers. Therefore, "or" is used in the exclusive sense.

Examples (OR vs XOR)

2. To register for BSC you must have passed the qualifying exam or be listed as an Math major.

Presumably, if you have passed the qualifying exam and are also listed as an Math major, you can still register for BCS. Therefore, “or” is inclusive.

5. Implication / Conditional Statements:

DEFINITION 5

Let p and q be propositions. The *conditional statement* $p \to q$, is the proposition "if p, then q." The conditional statement $p \to q$ is false when p is true and q is false, and true otherwise. In the conditional statement $p \to q$, p is called the *hypothesis* (or *antecedent* or *premise*) and q is called the *conclusion* (or *consequence*).

- A conditional statement is also called an implication.
- Example: "If I am elected, then I will lower taxes." $p \to q$

implication:

elected, lower taxes.	T	T	T
not elected, lower taxes.	F	T	T
not elected, not lower taxes.	F	F	T
elected, not lower taxes.	T	F	F

Example: Conditional Statements

- Example:
  - Let p be the statement "Maria learns discrete mathematics." and q the statement "Maria will find a good job." Express the statement $p \rightarrow q$ as a statement in English.

    Solution: Any of the following -

    "If Maria learns discrete mathematics, then she will find a good job.

    "Maria will find a good job when she learns discrete mathematics."

    "For Maria to get a good job, it is sufficient for her to learn discrete mathematics."

    "Maria will find a good job unless she does not learn discrete mathematics."

Examples: Implication

Examples of implications:

If you stand in the rain, then you'll get wet.
If you got an A in this class, I gave you \$5.

An implication $P \Longrightarrow Q$ is false only when P is true and Q is false. For example, the first statement would be false only if you stood in the rain but didn't get wet. The second statement above would be false only if you got an "A," yet I didn't give you \$5.

Here is the truth table for $P \Longrightarrow Q$:

P	Q	$P \Longrightarrow Q$	$\neg P \vee Q$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

[1] P is also called the *antecedent* and Q the *consequent*.

Examples: Implication

Note that $P \Longrightarrow Q$ is always true when P is false. This means that many statements that sound nonsensical in English are true, mathematically speaking. Examples are statements like: "If pigs can fly, then horses can read" or "If 14 is odd then $1+2=18$." When an implication is stupidly true because the hypothesis is false, we say that it is **vacuously true**. Note also that $P \Longrightarrow Q$ is logically equivalent to $\neg P \vee Q$, as can be seen in the above truth table.

$P \Longrightarrow Q$ is the most common form mathematical theorems take. Here are some of the different ways of saying it:

(1) If P, then Q.
(2) Q if P.
(3) P only if Q.
(4) P is sufficient for Q.
(5) Q is necessary for P.

Some other cases of implications:

"if p, then q"	"p implies q"
"if p, q"	"p only if q"
"p is sufficient for q"	"a sufficient condition for q is p"
"q if p"	"q whenever p"
"q when p"	"q is necessary for p"
"a necessary condition for p is q"	"q follows from p"
"q unless $\neg p$"	

Implication Example

- "If I weigh over 70 kg, then I will exercise."

Implication Example

- "If I weigh over 70 kg, then I will exercise."

- p: I weigh over 70 kg.
- q: I exercise.
- when is this claim false?

$p \to q$

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Implication Examples

- p_4: $3 < 8$, p_5: $3 < 14$, p_6: $3 < 2$, p_7: $8 < 6$

- $p_4 \to p_5$:
  if $3 < 8$, then $3 < 14$
  $T \to T : T$
- $p_4 \to p_6$:
  if $3 < 8$, then $3 < 2$
  $T \to F : F$
- $p_6 \to p_4$:
  if $3 < 2$, then $3 < 8$
  $F \to T : T$
- $p_6 \to p_7$:
  if $3 < 2$, then $8 < 6$
  $F \to F : T$

- Other conditional statements:
  - Converse of $p \rightarrow q$: $q \rightarrow p$
  - Contrapositive of $p \rightarrow q$: $\neg q \rightarrow \neg p$
  - Inverse of $p \rightarrow q$: $\neg p \rightarrow \neg q$

The contrapositive of "If you got an A in this class, I gave you \$5," is "If I did not give you \$5, you didn't get an A in this class." The converse is "If I gave you \$5 you must have received an A in this class." Does the contrapositive say the same thing as the original statement? Does the converse?

Let's look at the truth table:

P	Q	$\neg P$	$\neg Q$	$P \Longrightarrow Q$	$Q \Longrightarrow P$	$\neg Q \Longrightarrow \neg P$	$P \Longleftrightarrow Q$
T	T	F	F	T	T	T	T
T	F	F	T	F	T	F	F
F	T	T	F	T	F	T	F
F	F	T	T	T	T	T	T

Note that the contrapositive of $P \Longrightarrow Q$ has the same truth values, while the converse does not. Many students unreasonably assume that the converse is true, but the above truth table shows that it is not necessarily the case. When two propositional forms have the same truth values, they are said to be **logically equivalent** – they mean the same thing. We'll see next time how useful this can be for proving theorems.

Converse, Contrapositive, and Inverse

- From $p \to q$ we can form new conditional statements .
  - $q \to p$ is the **converse** of $p \to q$
  - $\neg q \to \neg p$ is the **contrapositive** of $p \to q$
  - $\neg p \to \neg q$ is the **inverse** of $p \to q$

Example: Find the converse, inverse, and contrapositive of "It is raining is a sufficient condition for my not going to town."

Solution:

converse: If I do not go to town, then it is raining.

inverse: If it is not raining, then I will go to town.

contrapositive: If I go to town, then it is not raining.

Contrapositive:

- Contrapositive of $p \rightarrow q$ is $\neg q \rightarrow \neg p$
- Any proposition and its contrapositive are logically equivalent (have the same truth table values) – Check with the truth table.
- E.g. The contrapositive of “If you get 100% in this course, you will get an A+” is “If you do not get an A+ in this course, you did not get 100%”.

Converse:

- Converse of $p \rightarrow q$ is $q \rightarrow p$
- Both are not logically equivalent.
- Ex 1: “If you get 100% in this course, you will get an A+” and “If you get an A+ in this course, you scored 100%” are not equivalent.
- Ex 2: If you won the lottery, you are rich.

Inverse:

- Inverse of $p \rightarrow q$ is $\neg p \rightarrow \neg q$
- Both are not logically equivalent.
- Ex1 : “If you get 100% in this course, you will get an A+” and “If you didn’t 100%, then won’t have an A+ in this course.” are not equivalent.
- Ex2: You can not ride the roller coaster if you are under 4 feet. What is its inverse statement?

Example of converse (1/2)

- Find the converse of the following statement:
- R: ‘Raining tomorrow is a sufficient condition for my not going to town.’
- Step 1: Assign propositional variables to
- component propositions
- P: It will rain tomorrow
- Q: I will not go to town

Example of converse (2/2)

- Step 2: Symbolize the assertion R: $P \rightarrow Q$
- Step 3: Symbolize the converse $Q \rightarrow P$
- Step 4: Convert the symbols back into words

‘If I don’t go to town then it will rain tomorrow’

or

‘Raining tomorrow is a necessary condition for my not going to town.’

or

‘My not going to town is a sufficient condition for it raining tomorrow.’

6. Bi-implications:

DEFINITION 6

Let p and q be propositions. The *biconditional statement* $p \leftrightarrow q$ is the proposition "p if and only if q." The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called *bi-implications*.

- $p \leftrightarrow q$ has the same truth value as $(p \rightarrow q) \wedge (q \rightarrow p)$
- "*if and only if*" can be expressed by "*iff*"
- Example:
  - Let p be the statement "You can take the flight" and let q be the statement "You buy a ticket." Then $p \leftrightarrow q$ is the statement

    "You can take the flight if and only if you buy a ticket."

    Implication:

    If you buy a ticket you can take the flight.

    If you don't buy a ticket you cannot take the flight.

Bi-implications:

If p denotes "I am at home." and q denotes "It is raining." then $p \leftrightarrow q$ denotes "I am at home if and only if it is raining."

If both $P \Longrightarrow Q$ and $Q \Longrightarrow P$ are true, then we say "P if and only if Q" (abbreviated P iff Q). Formally, we write $P \Longleftrightarrow Q$. P if and only if Q is true only when P and Q have the same truth values.

For example, if we let P be "3 is odd," Q be "4 is odd," and R be "6 is even," then $P \Longrightarrow R$, $Q \Longrightarrow P$ (vacuously), and $R \Longrightarrow P$. Because $P \Longrightarrow R$ and $R \Longrightarrow P$, P if and only if R.

Given an implication $P \Longrightarrow Q$, we can also define its

(a) **Contrapositive**: $\neg Q \Longrightarrow \neg P$
(b) **Converse**: $Q \Longrightarrow P$

The contrapositive of "If you got an A in this class, I gave you \$5," is "If I did not give you \$5, you didn't get an A in this class." The converse is "If I gave you \$5 you must have received an A in this class." Does the contrapositive say the same thing as the original statement? Does the converse?

The Truth Table for the Biconditional $p \leftrightarrow q$.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Expressing the Biconditional

- Some alternative ways "*p* if and only if *q*" is expressed in English:
  - *p* **is necessary and sufficient for** *q*
  - **if** *p* **then** *q* , **and conversely**
  - *p* **iff** *q*

Without changing their meanings, convert each of the following sentences into a sentence having the form "p iff q"

For a matrix to be invertible, it is necessary and sufficient that its determinant is not zero.

Answer: A matrix is invertible if and only if its determinant is not zero.

If $xy = 0$ then $x = 0$ or $y = 0$, and conversely.

Answer: $xy = 0$ if and only if $x = 0$ or $y = 0$

For an occurrence to become an adventure, it is necessary and sufficient for one to recount it.

Answer: An occurrence becomes an adventure if and only if one recounts it.

Truth Tables of Compound Propositions

- We can use connectives to build up complicated compound propositions involving any number of propositional variables, then use truth tables to determine the truth value of these compound propositions.
- Example: Construct the truth table of the compound proposition

$$(p \vee \neg q) \rightarrow (p \wedge q).$$

The Truth Table of $(p \vee \neg q) \rightarrow (p \wedge q)$.

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F

Example Truth Table

- Construct a truth table for $p \vee q \rightarrow \neg r$

p	q	r	$\neg r$	$p \vee q$	$p \vee q \rightarrow \neg r$
T	T	T	F	T	F
T	T	F	T	T	T
T	F	T	F	T	F
T	F	F	T	T	T
F	T	T	F	T	F
F	T	F	T	T	T
F	F	T	F	F	T
F	F	F	T	F	T

Problem:

- How many rows are there in a truth table with *n* propositional variables?

  Solution: 2^n We will see how to do this in Chapter 6.

- Note that this means that with n propositional variables, we can construct 2^n distinct (i.e., not equivalent) propositions.

Precedence of Logical Operators

- We can use parentheses to specify the order in which logical operators in a compound proposition are to be applied.
- To reduce the number of parentheses, the precedence order is defined for logical operators.

Precedence of Logical Operators.	
Operator	Precedence
$\neg$	1
$\wedge$ $\vee$	2 3
$\rightarrow$ $\leftrightarrow$	4 5

E.g. $\neg p \wedge q = (\neg p) \wedge q$

$p \wedge q \vee r = (p \wedge q) \vee r$

$p \vee q \wedge r = p \vee (q \wedge r)$

Translating English Sentences

(1) If P, then Q.
(2) Q if P.
(3) P only if Q.
(4) P is sufficient for Q.
(5) Q is necessary for P.

- English (and every other human language) is often ambiguous. Translating sentences into compound statements removes the ambiguity.
- Example: How can this English sentence be translated into a logical expression?

"You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old."

Solution: Let q, r, and s represent "You can ride the roller coaster,"

"You are under 4 feet tall," and "You are older than 16 years old." The sentence can be translated into:

$$(r \wedge \neg s) \rightarrow \neg q.$$

You cannot ride the coaster if You are under 4 feet tall and you are not older than 16 Years old.

Translating English Sentenc

(1) If P, then Q.
(2) Q if P.
(3) P only if Q.
(4) P is sufficient for Q.
(5) Q is necessary for P.

- Steps to convert an English sentence to a statement in propositional logic
  - Identify atomic propositions and represent using propositional variables.
  - Determine appropriate logical connectives
- "If I go to Harry's or to the country, I will not go shopping."
  - *p*: I go to Harry's
  - q: I go to the country.
  - *r*: I will go shopping.

If *p* or *q* then not *r*. $(p \vee q) \rightarrow \neg r$

Translating English Sentences

- Example: How can this English sentence be translated into a logical expression?

  "You can access the Internet from campus only if you are a computer science major or you are not a freshman."

Solution: Let a, c, and f represent "You can access the Internet from campus," "You are a computer science major," and "You are a freshman." The sentence can be translated into:

$$a \rightarrow (c \vee \neg f).$$

(1) If P, then Q.
(2) Q if P.
(3) P only if Q.
(4) P is sufficient for Q.
(5) Q is necessary for P.

System Specifications

- System and Software engineers take requirements in English and express them in a precise specification language based on logic.

  Example: Express in propositional logic:

  "The automated reply cannot be sent when the file system is full"

  Solution: One possible solution: Let *p* denote "The automated reply can be sent" and *q* denote "The file system is full."

$$q \rightarrow \neg p$$

1.1 Propositional Logic

Logic and Bit Operations

- Computers represent information using bits.
- A **bit** is a symbol with two possible values, 0 and 1.
- By convention, 1 represents T (true) and 0 represents F (false).
- A variable is called a Boolean variable if its value is either true or false.
- Bit operation – replace true by 1 and false by 0 in logical operations.

Table for the Bit Operators *OR*, *AND*, and *XOR*.

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

1.1 Propositional Logic

DEFINITION 7

A *bit string* is a sequence of zero or more bits. The *length* of this string is the number of bits in the string.

- Example: Find the bitwise *OR*, bitwise *AND*, and bitwise *XOR* of the bit string 01 1011 0110 and 11 0001 1101.

Solution:

```
01 1011 0110
11 0001 1101
      -------------------
     11 1011 1111     bitwise OR
01 0001 0100    bitwise AND
10 1010 1011    bitwise XOR
```

Propositional Equivalences

DEFINITION 1

A compound proposition that is always true, no matter what the truth values of the propositions that occurs in it, is called a *tautology*.

A compound proposition that is always false is called a *contradiction*.

A compound proposition that is neither a tautology or a contradiction is called a *contingency*.

Examples of a Tautology and a Contradiction.

p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

Tautologies and Contradictions

- Tautology is a statement that is always true regardless of the truth values of the individual logical variables

- Examples:
- $R \vee (\neg R)$
- $\neg(P \wedge Q) \leftrightarrow (\neg P) \vee (\neg Q)$

Tautologies and Contradictions

- A Contradiction is a statement that is always false regardless of the truth values of the individual logical variables

Examples

- $R \wedge (\neg R)$
- $\neg(\neg(P \wedge Q) \leftrightarrow (\neg P) \vee (\neg Q))$
- The negation of any tautology is a contradiction, and the negation of any contradiction is a tautology.

Propositional Equivalences

DEFINITION 2

The compound propositions p and q are called *logically equivalent* if $p \leftrightarrow q$ is a tautology. The notation $p \equiv q$ denotes that p and q are logically equivalent.

- Compound propositions that have the same truth values in all possible cases are called logically equivalent.
- Example: Show that $\neg p \vee q$ and $p \rightarrow q$ are logically equivalent.

Truth Tables for $\neg p \vee q$ and $p \rightarrow q$.

p	q	$\neg p$	$\neg p \vee q$	$p \rightarrow q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

Propositional Equivalences

- In general, 2^n rows are required if a compound proposition involves *n* propositional variables in order to get the combination of all truth values.

- Prove that $\neg(\neg p) \equiv p$

Solution

p	¬p	¬ (¬p)
T	F	T
F	T	F

As you can see the corresponding truth values of p and ¬ (¬p) are same, hence equivalence is justified.

Propositional Equivalences

Example

Show that the proposition forms $\neg(p \wedge q)$ and $\neg p \wedge \neg q$ are NOT logically equivalent.

p	q	$\neg p$	$\neg q$	$(p \wedge q)$	$\neg(p \wedge q)$	$\neg p \wedge \neg q$
T	T	F	F	T	F	F
T	F	F	T	F	T	F
F	T	T	F	F	T	F
F	F	T	T	F	T	T

Here the corresponding truth values differ and hence equivalence does not hold

Applications: Boolean Searches

- Logical connectives are used extensively in searches of large collections of information.
  - Example: indexes of Web pages.

- AND - used to match records that contain both of two search terms.
- OR - used to match one or both of two search terms.
- NOT - used to exclude a particular search term.

- Read about: Web Page Searching

Applications: Logic Puzzles

- Puzzles (important job interview question) that can be solved using logical reasoning

- [Sm78] Smullyan: An island that has two kinds of inhabitants.
  - knights, who always tell the truth.
  - knaves, who always lie.
- You encounter two people A and B.
- What are A and B if:
  - A says “B is a knight” and
  - B says “The two of us are opposite types?”

Example 1:

- p: A is a knight ¬p: A is a knave
- q: B is a knight ¬q: B is a knave

- Consider the possibility that A is a knight;
  - So, p is true. And he is telling truth.
  - So, q is true. So, A and B are the same type.

- However, if B is a knight, then B ’s statement that A and B are of opposite types, the statement $(p \wedge \neg q) \vee (\neg p \wedge q)$, would have to be true, which it is not, because A and B are both knights. Consequently, we can conclude that A is not a knight, that is, that p is false.

Example 1: Solution (cont..)

- Consider the possibility that A is a knave,
  - everything a knave says is false; q is true, is a lie.
  - So, q is false. B is also a knave.
  - B ’s statement that A and B are opposite types is a lie, which is consistent with both A and B being knaves.
- We can conclude that both A and B are knaves.

Example 2:

- A father tells his two children, a boy and a girl, to play in their backyard without getting dirty.
- However, while playing, both children get mud on their foreheads. When the children stop playing, the father says “At least one of you has a muddy forehead,” and then asks the children to answer “Yes” or “No” to the question: “Do you know whether you have a muddy forehead?”
- The father asks this question twice. What will the children answer each time this question is asked, assuming that a child can see whether his or her sibling has a muddy forehead, but cannot see his or her own forehead?
- Assume that both children are honest and that the children answer each question simultaneously.

Example 2:

Solution: S denotes "Son has a muddy forehead" and D denotes " Daughter has a muddy forehead". The father states that $S \vee D$ is True. Boy can know D is True but can't know S. Girl can know D is True but can't know S. So no for the first time. After that they can conclude that both D and S are True. Since one of them will say yes for the first time if one of D and S is not True.

Example 3:

6. Determine whether these system specifications are consistent:
   "the diagnostic message is stored in the buffer or it is retransmitted"
   "the diagnostic message is not stored in the buffer"
   "if the diagnostic message is stored in the buffer, then it is retransmitted"

Solution: p denotes "the diagnostic message is stored in the buffer", q denotes "the diagnostic message is retransmitted". Then the specifications can be written as $p \vee q$, $\neg p$ and $p \rightarrow q$. $\neg p$ is True, so p is False. $p \vee q$ is True, so q is True. So $p \rightarrow q$ is True. They are consistent.

De Morgan's laws

De Morgan's laws state that:

The negation of an **and** proposition is logically equivalent to the **or** proposition in which each component is negated.

1. $\neg(p \wedge q) \equiv \neg p \vee \neg q$

The negation of an **or** proposition is logically equivalent to the **and** proposition in which each component is negated.

2. $\neg(p \vee q) \equiv \neg p \wedge \neg q$

Applying De-Morgan's Law

Question: Negate the following compound Propositions

1. John is six feet tall and he weights at least 200 pounds.

2. The bus was late or Tom's watch was slow.

Applying De-Morgan's Law

Question: Negate the following compound Propositions

1. John is six feet tall and he weights at least 200 pounds.

2. The bus was late or Tom's watch was slow.

Solution

a) John is not six feet tall or he weighs less than 200 pounds.

b) The bus was not late and Tom's watch was not slow.

Inequalities and De Morgan's Laws

Question Use De Morgan's laws to write the negation of

$$-1 < x \leq 4$$

Solution: The given proposition is equivalent to

$$-1 < x \quad \text{and} \quad x \leq 4,$$

By De Morgan's laws, the negation is

$$-1 \geq x \quad \text{or} \quad x > 4.$$

Laws of Logic

1. Commutative laws

   $p \wedge q \equiv q \wedge p$; $p \vee q \equiv q \vee p$

2. Associative laws

   $p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$; $p \vee (q \vee r) \equiv (p \vee q) \vee r$

3. Distributive laws

   $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
   $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

Laws of Logic

4. Identity laws

   $p \wedge t \equiv p$; $p \vee c \equiv p$

5. Negation laws

   $p \vee \neg p \equiv t$; $p \wedge \neg p \equiv c$

6. Double negation law

   $\neg(\neg p) \equiv p$

7. Idempotent laws

   $p \wedge p \equiv p$; $p \vee p \equiv p$

Laws of Logic

8. Universal bound laws

$p \vee t \equiv t$; $p \wedge c \equiv c$

9. Absorption laws

$p \wedge (p \vee q) \equiv p$; $p \vee (p \wedge q) \equiv p$

10. Negation of *t* and *c*

$\neg t \equiv c$; $\neg c \equiv t$

Example

Show that the proposition form $p \vee \neg p$ is a tautology and the proposition form $p \wedge \neg p$ is a contradiction.

p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

Exercise: If t is a tautology and c is contradiction, show that $p \vee t \equiv p$ and $p \wedge c \equiv c$?

Propositional Equivalences

Constructing New Logical Equivalences

- Example: Show that $\neg(p \rightarrow q)$ and $p \wedge \neg q$ are logically equivalent.

  Solution:

  $\neg(p \rightarrow q) \equiv \neg(\neg p \vee q)$ *by example discussed in slide 66*

  $\equiv \neg(\neg p) \wedge \neg q$ *by the second De Morgan law*

  $\equiv p \wedge \neg q$ *by the double negation law*

- *Example: Show that* $(p \wedge q) \rightarrow (p \vee q)$ is a tautology.

  Solution: To show that this statement is a tautology, we will use logical equivalences to demonstrate that it is logically equivalent to T.

  $(p \wedge q) \rightarrow (p \vee q) \equiv \neg(p \wedge q) \vee (p \vee q)$ *by example already discussed*

  $\equiv (\neg p \vee \neg q) \vee (p \vee q)$ *by the first De Morgan law*

  $\equiv (\neg p \vee p) \vee (\neg q \vee q)$ *by the associative and communicative law for disjunction*

  $\equiv T \vee T$

Exercise

Using laws of logic, show that

$\neg(\neg p \wedge q) \wedge (p \vee q) \equiv p.$

Solution

Take $\neg(\neg p \wedge q) \wedge (p \vee q)$

$\equiv (\neg(\neg p) \vee \neg q) \wedge (p \vee q)$, (by De Morgan's laws)

$\equiv (p \vee \neg q) \wedge (p \vee q)$, (by double negative law)

$\equiv p \vee (\neg q \wedge q)$, (by distributive law)

$\equiv p \vee (q \wedge \neg q)$, (by the commutative law)

$\equiv p \vee c$, (by the negation law)

$\equiv p$, (by the identity law)

Skill in simplifying proposition forms is useful in constructing logically efficient computer programs and in designing digital circuits.

Exercise

Prove that $\neg[r \vee (q \wedge (\neg r \rightarrow \neg p))] \equiv \neg r \wedge (p \vee \neg q)$

$\neg[r \vee (q \wedge (\neg r \rightarrow \neg p))]$

$\equiv \neg r \wedge \neg(q \wedge (\neg r \rightarrow \neg p))$,	De Morgan's law
$\equiv \neg r \wedge \neg(q \wedge (\neg\neg r \vee \neg p))$,	Conditional rewritten as disjunction
$\equiv \neg r \wedge \neg(q \wedge (r \vee \neg p))$,	Double negation law
$\equiv \neg r \wedge (\neg q \vee \neg(r \vee \neg p))$,	De Morgan's law
$\equiv \neg r \wedge (\neg q \vee (\neg r \wedge p))$,	De Morgan's law, double negation
$\equiv (\neg r \wedge \neg q) \vee (\neg r \wedge (\neg r \wedge p))$,	Distributive law
$\equiv (\neg r \wedge \neg q) \vee ((\neg r \wedge \neg r) \wedge p)$,	Associative law
$\equiv (\neg r \wedge \neg q) \vee (\neg r \wedge p)$,	Idempotent law
$\equiv \neg r \wedge (\neg q \vee p)$,	Distributive law
$\equiv \neg r \wedge (p \vee \neg q)$,	Commutative law

Exercise

Prove that:

$\neg(P \leftrightarrow Q) \equiv P \leftrightarrow \neg Q$

** $P \rightarrow Q \equiv \neg P \vee Q$

$P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$

$\equiv (\neg P \vee Q) \wedge (\neg Q \vee P)$

$\neg(P \leftrightarrow Q)$

$\equiv \neg((\neg P \vee Q) \wedge (\neg Q \vee P))$ ##

$\equiv \neg(\neg P \vee Q) \vee \neg(\neg Q \vee P)$ *De Morgan's Laws*

$\equiv (P \wedge \neg Q) \vee (Q \wedge \neg P)$ *De Morgan's Laws*

$\equiv ((P \wedge \neg Q) \vee Q) \wedge ((P \wedge \neg Q) \vee \neg P)$ *Distributive Laws*

$\equiv ((P \vee Q) \wedge (\neg Q \vee Q)) \wedge ((P \vee \neg P) \wedge (\neg Q \vee \neg P))$ *Distributive Laws*

$\equiv (P \vee Q) \wedge T \wedge T \wedge (\neg Q \vee \neg P)$ *Negation Laws*

$\equiv (P \vee Q) \wedge (\neg Q \vee \neg P)$ *Identify Laws*

$\equiv P \leftrightarrow \neg Q$ ##

The Foundations: Logic and Proofs

Chapter 1, Part II: Predicate Logic

Summary

- Predicate Logic (First-Order Logic (FOL), Predicate Calculus)
  - The Language of Quantifiers
  - Logical Equivalences
  - Nested Quantifiers
  - Translation from Predicate Logic to English
  - Translation from English to Predicate Logic

Predicates and Quantifiers

Section 1.4

Section Summary

- Predicates
- Variables
- Quantifiers
  - Universal Quantifier
  - Existential Quantifier
- Negating Quantifiers
  - De Morgan’s Laws for Quantifiers
- Translating English to Logic
- Logic Programming (*optional*)

Propositional Logic Not Enough

- If we have:
  - “All men are mortal.”
  - “Socrates is a man.”
- Does it follow that “Socrates is mortal?”
- Can’t be represented in propositional logic. Need a language that talks about objects, their properties, and their relations.
- Later we’ll see how to draw inferences.

Introducing Predicate Logic

- Predicate logic uses the following new features:
  - Variables: x, y, z
  - Predicates: $P(x)$, $M(x)$
  - Quantifiers (*to be covered in a few slides*):
- *Propositional functions* are a generalization of propositions.
  - They contain variables and a predicate, e.g., $P(x)$
  - Variables can be replaced by elements from their *domain*.

Propositional Functions

- Propositional functions become propositions (and have truth values) when their variables are each replaced by a value from the *domain* (or *bound* by a quantifier, as we will see later).
- The statement *P(x)* is said to be the value of the propositional function *P* at *x*.
- For example, let *P(x)* denote "$x > 0$" and the domain be the integers. Then:

  P(-3) is false.

  P(0) is false.

  P(3) is true.
- Often the domain is denoted by *U*. So in this example *U* is the integers.

Examples of Propositional Functions

- Let "$x + y = z$" be denoted by $R(x, y, z)$ and U (for all three variables) be the integers. Find these truth values:

  R(2,-1,5)

  Solution: F

  R(3,4,7)

  Solution: T

  R(x, 3, z)

  Solution: Not a Proposition

- Now let "$x - y = z$" be denoted by $Q(x, y, z)$, with U as the integers. Find these truth values:

  Q(2,-1,3)

  Solution: T

  Q(3,4,7)

  Solution: F

  Q(x, 3, z)

  Solution: Not a Proposition

Compound Expressions

- Connectives from propositional logic carry over to predicate logic.
- If *P(x)* denotes "$x > 0$," find these truth values:

  $P(3) \vee P(-1)$ **Solution**: T

  $P(3) \wedge P(-1)$ **Solution**: F

  $P(3) \rightarrow P(-1)$ **Solution**: F

  $P(3) \rightarrow P(1)$ **Solution**: T
- Expressions with variables are not propositions and therefore do not have truth values. For example,

  $P(3) \wedge P(y)$

  $P(x) \rightarrow P(y)$
- When used with quantifiers (to be introduced next), these expressions (propositional functions) become propositions.

Quantifiers

Charles Peirce (1839-1914)

- We need *quantifiers* to express the meaning of English words including *all* and *some*:
  - "All men are Mortal."
  - "Some cats do not have fur."
- The two most important quantifiers are:
  - *Universal Quantifier,* "For all," symbol: $\forall$
  - *Existential Quantifier*, "There exists," symbol: $\exists$
- We write as in $\forall x\, P(x)$ and $\exists x\, P(x)$.
- $\forall x\, P(x)$ asserts $P(x)$ is true for <u>every</u> x in the *domain*.
- $\exists x\, P(x)$ asserts $P(x)$ is true for <u>some</u> x in the *domain*.
- The quantifiers are said to bind the variable x in these expressions.

Quantifiers

- *Universal Quantifier,* **“**For all,” symbol: ∀
- *Existential Quantifier*, “There exists,” symbol: ∃

TABLE 1 Quantifiers.

Statement	*When True?*	*When False?*
$\forall x P(x)$	$P(x)$ is true for every x.	There is an x for which $P(x)$ is false.
$\exists x P(x)$	There is an x for which $P(x)$ is true.	$P(x)$ is false for every x.

Universal Quantifier

$\forall x\ P(x)$ is read as " For all x, P(x)" or "For every x, P(x)"

Examples:

1) If $P(x)$ denotes "$x > 0$" and U is the integers, then $\forall x\ P(x)$ is false.
2) If $P(x)$ denotes "$x > 0$" and U is the positive integers, then $\forall x\ P(x)$ is true.
3) If $P(x)$ denotes "x is even" and U is the integers, then $\forall\ x\ P(x)$ is false.

Existential Quantifier

- $\exists x\ P(x)$ is read as "For some x, P(x)", or as "There is an x such that P(x)," or "For at least one x, P(x)."

Examples:

1. If $P(x)$ denotes "$x > 0$" and U is the integers, then $\exists x\ P(x)$ is true. It is also true if U is the positive integers.
2. If $P(x)$ denotes "$x < 0$" and U is the positive integers, then $\exists x\ P(x)$ is false.
3. If $P(x)$ denotes "x is even" and U is the integers, then $\exists x\ P(x)$ is true.

Uniqueness Quantifier (*optional*)

- $\exists! x\ P(x)$ means that $P(x)$ is true for <u>one and only one</u> x in the universe of discourse.
- This is commonly expressed in English in the following equivalent ways:
  - "There is a unique x such that $P(x)$."
  - "There is one and only one x such that $P(x)$"
- Examples:
  1. If $P(x)$ denotes "$x + 1 = 0$" and U is the integers, then $\exists! x\ P(x)$ is true.
  2. But if $P(x)$ denotes "$x > 0$," then $\exists! x\ P(x)$ is false.
- The uniqueness quantifier is not really needed as the restriction that there is a unique x such that $P(x)$ can be expressed as:

$$\exists x\ (P(x) \wedge \forall y\ (P(y) \rightarrow y = x))$$

Thinking about Quantifiers

- When the domain of discourse is finite, we can think of quantification as looping through the elements of the domain.
- To evaluate $\forall x\, P(x)$ loop through all x in the domain.
  - If at every step $P(x)$ is true, then $\forall x\, P(x)$ is true.
  - If at a step $P(x)$ is false, then $\forall x\, P(x)$ is false and the loop terminates.
- To evaluate $\exists x\, P(x)$ loop through all x in the domain.
  - If at some step, $P(x)$ is true, then $\exists x\, P(x)$ is true and the loop terminates.
  - If the loop ends without finding an x for which $P(x)$ is true, then $\exists x\, P(x)$ is false.
- Even if the domains are infinite, we can still think of the quantifiers this fashion, but the loops will not terminate in some cases.

Properties of Quantifiers

- The truth value of $\exists x\, P(x)$ and $\forall x\, P(x)$ depend on both the propositional function $P(x)$ and on the domain U.
- **Examples**:
  1. If U is the positive integers and $P(x)$ is the statement "$x < 2$", then $\exists x\, P(x)$ is true, but $\forall x\, P(x)$ is false.
  2. If U is the negative integers and $P(x)$ is the statement "$x < 2$", then both $\exists x\, P(x)$ and $\forall x\, P(x)$ are true.
  3. If U consists of 3, 4, and 5, and $P(x)$ is the statement "$x > 2$", then both $\exists x\, P(x)$ and $\forall x\, P(x)$ are true. But if $P(x)$ is the statement "$x < 2$", then both $\exists x\, P(x)$ and $\forall x\, P(x)$ are false.

Precedence of Quantifiers

- The quantifiers $\forall$ and $\exists$ have higher precedence than all the logical operators.
- For example, $\forall x\, P(x) \vee Q(x)$ means $(\forall x\, P(x)) \vee Q(x)$
- $\forall x\, (P(x) \vee Q(x))$ means something different.
- Unfortunately, often people write $\forall x\, P(x) \vee Q(x)$ when they mean $\forall x\, (P(x) \vee Q(x))$.

Translating from English to Logic

Example 1: Translate the following sentence into predicate logic: "Every student in this class has taken a course in Java."

Solution:

First decide on the domain *U*.

Solution 1: If *U* is all students in this class, define a propositional function J(x) denoting "x has taken a course in Java" and translate as $\forall x\ J(x)$.

Solution 2: But if *U* is all people, also define a propositional function S(x) denoting "x is a student in this class" and translate as $\forall x\ (S(x) \rightarrow J(x))$.

$\forall x\ (S(x) \wedge J(x))$ is not correct. What does it

For example,
we may be interested in a wider group of people than only those in this class. If we change the domain to consist of all people, we will need to express our statement as
“For every person x, if person x is a student in this class then x has studied Java.”
If S(x) represents the statement that person x is in this class, we see that our statement can be expressed as $\forall x(S(x) \rightarrow J(x))$.
[Caution! Our statement cannot be expressed as $\forall x(S(x) \wedge J(x))$ because this statement says that all people are students in this class and have studied Java!]

Translating from English to Logic

Example 2: Translate the following sentence into predicate logic: "Some student in this class has taken a course in Java."

Solution:

First decide on the domain *U*.

Solution 1: If *U* is all students in this class, translate as

$$\exists x\ J(x)$$

Solution 1: But if *U* is all people, then translate as

$$\exists x\ (S(x) \wedge J(x))$$

$\exists x\ (S(x) \rightarrow J(x))$ is not correct. What does it mean?

if we are interested in people other than those in this class, we look at the statement a little differently. Our statement can be expressed as
“There is a person x having the properties that x is a student in this class and x has studied Java.”
In this case, the domain for the variable x consists of all people. We introduce S(x) to represent
“x is a student in this class.” Our solution becomes $\exists x(S(x) \wedge J(x))$ because the statement is that there is a person x who is a student in this class and who has studied Java.
[Caution! Our statement cannot be expressed as $\exists x(S(x) \rightarrow J(x))$, which is true when there is someone not in the class because, in that case, for such a person x, $S(x) \rightarrow J(x)$ becomes either $F \rightarrow T$ or $F \rightarrow F$, both of which are true.]

Returning to the Socrates Example

- Introduce the propositional functions *Man(x)* denoting "*x* is a man" and *Mortal(x)* denoting "*x* is mortal." Specify the domain as all people.
- The two premises are:

$$\forall x Man(x) \rightarrow Mortal(x)$$

$$Man(Socrates)$$

- The conclusion is:

$$Mortal(Socrates)$$

- Later we will show how to prove that the conclusion follows from the premises.

Equivalences in Predicate Logic

- Statements involving predicates and quantifiers are *logically equivalent* if and only if they have the same truth value
  - for every predicate substituted into these statements and
  - for every domain of discourse used for the variables in the expressions.
- The notation $S \equiv T$ indicates that S and T are logically equivalent.
- **Example**: $\forall x \neg\neg S(x) \equiv \forall x\, S(x)$

Thinking about Quantifiers as Conjunctions and Disjunctions

- If the domain is finite, a universally quantified proposition is equivalent to a conjunction of propositions without quantifiers and an existentially quantified proposition is equivalent to a disjunction of propositions without quantifiers.
- If *U* consists of the integers 1,2, and 3:

$$\forall x P(x) \equiv P(1) \wedge P(2) \wedge P(3)$$

$$\exists x P(x) \equiv P(1) \vee P(2) \vee P(3)$$

- Even if the domains are infinite, you can still think of the quantifiers in this fashion, but the equivalent expressions without quantifiers will be infinitely long.

Negating Quantified Expressions

- Consider $\forall x\ J(x)$

  "Every student in your class has taken a course in Java."

  Here $J(x)$ is "x has taken a course in Java" and the domain is students in your class.

- Negating the original statement gives "It is not the case that every student in your class has taken a course in Java." This implies that "There is a student in your class who has not taken a course in Java."

  Symbolically $\neg \forall x\ J(x)$ and $\exists x\ \neg J(x)$ are equivalent

Negating Quantified Expressions (*continued*)

- Now Consider $\exists x\ J(x)$

  "There is a student in this class who has taken a course in Java."

  Where $J(x)$ is "x has taken a course in Java."
- Negating the original statement gives "It is not the case that there is a student in this class who has taken Java." This implies that "Every student in this class has not taken a course in Java"

  Symbolically $\neg\exists x\ J(x)$ and $\forall x\ \neg J(x)$ are equivalent

De Morgan's Laws for Quantifiers

- The rules for negating quantifiers are:

TABLE 2 **De Morgan's Laws for Quantifiers.**

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

- The reasoning in the table shows that:

$$\neg\forall x P(x) \equiv \exists x \neg P(x)$$

$$\neg\exists x P(x) \equiv \forall x \neg P(x)$$

- These are important. You will use these.

Translation from English to Logic

Examples:

1. "Some student in this class has visited Mexico."

   Solution: Let $M(x)$ denote "x has visited Mexico" and $S(x)$ denote "x is a student in this class," and U be all people.

$$\exists x \ (S(x) \wedge M(x))$$

2. "Every student in this class has visited Canada or Mexico."

   Solution: Add $C(x)$ denoting "x has visited Canada."

   and U be all people.

$$\forall x \ (S(x) \rightarrow (M(x) \vee C(x)))$$

Some Fun with Translating from English into Logical Expressions

- U = {Nitwits, Blubbers, Oddments}

  F(x): *x* is a Nitwit

  S(x): *x* is a Blubber

  T(x): x is a Oddment

  Translate “Everything is a Nitwit”

Solution: $\forall x\ F(x)$

Translation (cont)

- U = {Nitwits, Blubbers, Oddments}

  F(x): *x* is a Nitwit

  S(x): *x* is a Blubber

  T(x): *x* is a Oddment

  "Nothing is a Blubber."

Solution: $\neg\exists x\, S(x)$ What is this equivalent to?

Solution: $\forall x \neg S(x)$

Translation (cont)

- U = {Nitwits, Blubbers, Oddments}

  F(x): *x* is a Nitwit

  S(x): *x* is a Blubber

  T(x): *x* is a Oddment

"All Nitwits are Blubbers."

Solution: $\forall x\ (F(x) \rightarrow S(x))$

Translation (cont)

- U = {Nitwits, Blubbers, Oddments}

  F(x): *x* is a Nitwit

  S(x): *x* is a Blubber

  T(x): *x* is a Oddment

"Some Nitwits are Oddments."

Solution: $\exists x\ (F(x) \wedge T(x))$

Translation (cont)

- U = {Nitwits, Blubbers, Oddments}

  F(x): *x* is a Nitwit

  S(x): *x* is a Blubber

  T(x): *x* is a Oddment

  "No Blubber is a Oddment."

  Solution: $\neg\exists x\ (S(x) \wedge T(x))$ What is this equivalent to?

  Solution: $\forall x \neg\ (S(x) \wedge T(x))$

  $$\forall x\ (\neg S(x) \vee \neg T(x))$$

Translation (cont)

- U = {Nitwits, Blubbers, Oddments}

  F(x): x is a Nitwit

  S(x): *x* is a Blubber

  T(x): *x* is a Oddment

"If any Nitwit is a Blubber then it is also a Oddment."

Solution: $\forall x\ ((F(x) \wedge S(x)) \rightarrow T(x))$

System Specification Example

- Predicate logic is used for specifying properties that systems must satisfy.
- For example, translate into predicate logic:
  - "Every mail message larger than one megabyte will be compressed."
  - "If a user is active, at least one network link will be available."
- Decide on predicates and domains (left implicit here) for the variables:
  - Let $L(m, y)$ be "Mail message m is larger than y megabytes."
  - Let $C(m)$ denote "Mail message m will be compressed."
  - Let $A(u)$ represent "User u is active."
  - Let $S(n, x)$ represent "Network link n is state x.
- Now we have:

$$\forall m(L(m,1) \rightarrow C(m))$$

$$\exists u\, A(u) \rightarrow \exists n\, S(n, available)$$

Lewis Carroll Example

Charles Lutwidge Dodgson
(AKA Lewis Caroll)
(1832-1898)

- The first two are called *premises* and the third is called the *conclusion*.
  1. "All lions are fierce."
  2. "Some lions do not drink coffee."
  3. "Some fierce creatures do not drink coffee."
- Here is one way to translate these statements to predicate logic. Let P(x), Q(x), and R(x) be the propositional functions "x is a lion," "x is fierce," and "x drinks coffee," respectively.
  1. $\forall x\ (P(x) \rightarrow Q(x))$
  2. $\exists x\ (P(x) \wedge \neg R(x))$
  3. $\exists x\ (Q(x) \wedge \neg R(x))$
- Later we will see how to prove that the conclusion follows from the premises.

Nested Quantifiers

Section 1.5

Section Summary

- Nested Quantifiers
- Order of Quantifiers
- Translating from Nested Quantifiers into English
- Translating Mathematical Statements into Statements involving Nested Quantifiers.
- Translated English Sentences into Logical Expressions.
- Negating Nested Quantifiers.

Nested Quantifiers

- Nested quantifiers are often necessary to express the meaning of sentences in English as well as important concepts in computer science and mathematics.

  Example: "Every real number has an inverse" is

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

  where the domains of x and y are the real numbers.

- We can also think of nested propositional functions:

  $\forall x\, \exists y(x + y = 0)$ can be viewed as $\forall x\, Q(x)$ where $Q(x)$ is $\exists y\, P(x, y)$ where $P(x, y)$ *is* $(x + y = 0)$

Thinking of Nested Quantification

- Nested Loops
  - To see if $\forall x \forall y P(x,y)$ is true, loop through the values of x :
    - At each step, loop through the values for y.
    - If for some pair of x and y, $P(x,y)$ is false, then $\forall x \ \forall y P(x,y)$ is false and both the outer and inner loop terminate.

    $\forall x \ \forall y \ P(x,y)$ is true if the outer loop ends after stepping through each x.
  - To see if $\forall x \exists y P(x,y)$ is true, loop through the values of x:
    - At each step, loop through the values for y.
    - The inner loop ends when a pair x and y is found such that $P(x, y)$ is true.
    - If no y is found such that $P(x, y)$ is true the outer loop terminates as $\forall x \exists y P(x,y)$ has been shown to be false.

    $\forall x \exists y \ P(x,y)$ is true if the outer loop ends after stepping through each x.
- If the domains of the variables are infinite, then this process can not actually be carried out.

Order of Quantifiers

Examples:

1. Let $P(x,y)$ be the statement "$x + y = y + x$." Assume that U is the real numbers. Then $\forall x \ \forall y P(x,y)$ and $\forall y \ \forall x P(x,y)$ have the same truth value.
2. Let $Q(x,y)$ be the statement "$x + y = 0$." Assume that U is the real numbers. Then $\forall x \exists y P(x,y)$ is true, but $\exists y \ \forall x P(x,y)$ is false.

Questions on Order of Quantifiers

Example 1: Let *U* be the real numbers,

Define $P(x,y) : x \bullet y = 0$

What is the truth value of the following:

1. $\forall x \forall y P(x,y)$

   Answer: False

2. $\forall x \exists y P(x,y)$

   Answer: True

3. $\exists x \forall y\ P(x,y)$

   Answer: True

4. $\exists x\ \exists\ y\ P(x,y)$

   Answer: True

Questions on Order of Quantifiers

Example 2: Let *U* be the real numbers,

Define *P(x,y)* : *x / y* = 1

What is the truth value of the following:

1. $\forall x \forall y P(x,y)$

   Answer: False

2. $\forall x \exists y P(x,y)$

   Answer: True

3. $\exists x \forall y\ P(x,y)$

   Answer: False

4. $\exists x\ \exists\ y\ P(x,y)$

   Answer: True

Quantifications of Two Variables

Statement	When True?	When False
$\forall x \forall y P(x,y)$ $\forall y \forall x P(x,y)$	*P*(*x*,*y*) is true for every pair *x*,*y*.	There is a pair *x, y* for which *P*(*x*,*y*) is false.
$\forall x \exists y P(x,y)$	For every *x* there is a *y* for which *P*(*x*,*y*) is true.	There is an x such that *P*(*x*,*y*) is false for every *y*.
$\exists x \forall y P(x,y)$	There is an *x* for which *P*(*x*,*y*) is true for every *y*.	For every *x* there is a y for which *P*(x,y) is false.
$\exists x \exists y P(x,y)$ $\exists y \exists x P(x,y)$	There is a pair *x, y* for which *P*(*x*,*y*) is true.	*P*(x,y) is false for every pair *x*,*y*

Translating Nested Quantifiers into English

Example 1: Translate the statement

$$\forall x \ (C(x) \vee \exists y \ (C(y) \wedge F(x, y)))$$

where C(x) is "*x* has a computer," and *F*(*x*,*y*) is "*x* and *y* are friends," and the domain for both *x* and *y* consists of all students in your school.

Solution: Every student in your school has a computer or has a friend who has a computer.

Translating Mathematical Statements into Predicate Logic

Example : Translate "The sum of two positive integers is always positive" into a logical expression.

Solution:

1. Rewrite the statement to make the implied quantifiers and domains explicit:

   "For every two integers, if these integers are both positive, then the sum of these integers is positive."

2. Introduce the variables x and y, and specify the domain, to obtain:

   "For all positive integers x and y, $x + y$ is positive."

3. The result is:

$$\forall x\ \forall y\ ((x > 0) \wedge (y > 0) \rightarrow (x + y > 0))$$

where the domain of both variables consists of all integers

Translating English into Logical Expressions Example

Example: Use quantifiers to express the statement "There is a woman who has taken a flight on every airline in the world."

Solution:

1. Let $P(w,f)$ be "w has taken f" and $Q(f,a)$ be "f is a flight on a."
2. The domain of w is all women, the domain of f is all flights, and the domain of a is all airlines.
3. Then the statement can be expressed as:

Questions on Translation from English

Choose the obvious predicates and express in predicate logic.

Example 1: "Brothers are siblings."

Solution: $\forall x\ \forall y\ (B(\mathrm{x},\mathrm{y}) \rightarrow S(\mathrm{x},\mathrm{y}))$

Example 2: "Siblinghood is symmetric."

Solution: $\forall x\ \forall y\ (S(x,y) \rightarrow S(y,x))$

Example 3: "Everybody loves somebody."

Solution: $\forall x\ \exists y\ L(x,y)$

Example 4: "There is someone who is loved by everyone."

Solution: $\exists y\ \forall x\ L(x,y)$

Example 5: "There is someone who loves someone."

Solution: $\exists x\ \exists y\ L(x,y)$

Example 6: "Everyone loves himself"

Solution: $\forall x\ L(x,x)$

Negating Nested Quantifiers

Example 1: Recall the logical expression developed three slides back:

$$\exists w\ \forall a\ \exists f\ \ (P(w,f) \wedge Q(f,a))$$

Part 1: Use quantifiers to express the statement that "There does not exist a woman who has taken a flight on every airline in the world."

Solution: $\neg\exists w\ \forall a\ \exists f\ \ (P(w,f) \wedge Q(f,a))$

Part 2: Now use De Morgan's Laws to move the negation as far inwards as possible.

Solution:

1. $\neg\exists w\ \forall a\ \exists f\ \ (P(w,f) \wedge Q(f,a))$
2. $\forall w \neg\ \forall a\ \exists f\ \ (P(w,f) \wedge Q(f,a))$ by De Morgan's for $\exists$
3. $\forall w\ \exists\ a \neg\ \exists f\ \ (P(w,f) \wedge Q(f,a))$ by De Morgan's for $\forall$
4. $\forall w\ \exists\ a\ \forall f \neg\ (P(w,f) \wedge Q(f,a))$ by De Morgan's for $\exists$
5. $\forall w\ \exists\ a\ \forall f\ (\neg\ P(w,f) \vee \neg\ Q(f,a))$ by De Morgan's for $\wedge$.

Part 3: Can you translate the result back into English?

Solution:

"For every woman there is an airline such that for all flights, this woman has not taken that flight or that flight is not on this airline"

Some Questions about Quantifiers

- Can you switch the order of quantifiers?
  - Is this a valid equivalence? $\forall x \forall y P(x,y) \equiv \forall y \forall x P(x,y)$

Solution: Yes! The left and the right side will always have the same truth value. The order in which *x* and *y* are picked does not matter.

  - Is this a valid equivalence? $\forall x \exists y P(x,y) \equiv \exists y \forall x P(x,y)$

Solution: No! The left and the right side may have different truth values for some propositional functions for *P*. Try "x + y = 0" for *P(x,y)* with *U* being the integers. The order in which the values of *x* and *y* are picked does matter.

- Can you distribute quantifiers over logical connectives?
  - Is this a valid equivalence? $\forall x(P(x) \wedge Q(x)) \equiv \forall x P(x) \wedge \forall x Q(x)$

Solution: Yes! The left and the right side will always have the same truth value no matter what propositional functions are denoted by *P(x)* and *Q(x)*.

  - Is this a valid equivalence? $\forall x(P(x) \rightarrow Q(x)) \equiv \forall x P(x) \rightarrow \forall x Q(x)$

Solution: No! The left and the right side may have different truth values. Pick "x is a fish" for P(x) and "x has scales" for Q(x) with the domain of discourse being all animals. Then the left side is false, because there are some fish that do not have scales. But the right side is true since not all animals are fish.

The Foundations: Logic and Proofs

Chapter 1, Part III: Proofs

Summary

- Valid Arguments and Rules of Inference
- Proof Methods
- Proof Strategies

Rules of Inference

Section 1.6

Section Summary

- Valid Arguments
- Inference Rules for Propositional Logic
- Using Rules of Inference to Build Arguments
- Rules of Inference for Quantified Statements
- Building Arguments for Quantified Statements

Revisiting the Socrates Example

- We have the two premises:
  - “All men are mortal.”
  - “Socrates is a man.”
- And the conclusion:
  - “Socrates is mortal.”
- How do we get the conclusion from the premises?

The Argument

- We can express the premises (above the line) and the conclusion (below the line) in predicate logic as an argument:

$$\frac{\begin{array}{c}\forall x(Man(x) \rightarrow Mortal(x)) \\ Man(Socrates)\end{array}}{\therefore \quad Mortal(Socrates)}$$

- We will see shortly that this is a valid argument.

Valid Arguments

- We will show how to construct valid arguments in two stages; first for propositional logic and then for predicate logic. The rules of inference are the essential building block in the construction of valid arguments.

1. Propositional Logic

   Inference Rules

2. Predicate Logic

   Inference rules for propositional logic plus

Valid Arguments (Propositional Logic)

- “If you have a current password, then you can log onto the network.”
- “You have a current password.”
- Therefore, “You can log onto the network.”

- Sequence of propositions (argument) is valid; the conclusion must be true when both premise are true.

Argument Form

- p: “You have a current password”
- q: “You can log onto the network.”
- the argument has the form

$$p \rightarrow q$$
$$p$$
$$\therefore q$$

the statement $((p \rightarrow q) \wedge p) \rightarrow q$ is a tautology

Argument Form

- p: ““You have access to the network”
- q: “You can change your grade”
- the argument has the form

“If you have access to the network, then you can change your grade.”

“You have access to the network.”

∴ “You can change your grade.”

- The argument is valid, but if premise is false, we cannot conclude that the conclusion is true.

Arguments in Propositional Logic

- A *argument* in propositional logic is a sequence of propositions. All but the final proposition are called *premises*. The last statement is the *conclusion*.
- The argument is valid if the premises imply the conclusion. An *argument form* is an argument that is valid no matter what propositions are substituted into its propositional variables.
- If the premises are $p_1, p_2, \ldots, p_n$ and the conclusion is q then

  $(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \rightarrow q$ is a tautology.
- Inference rules are all argument simple argument forms that will be used to construct more complex argument forms.

TABLE 1 **Rules of Inference.**

Rule of Inference	*Tautology*	*Name*
p $p \to q$ $\therefore q$	$(p \wedge (p \to q)) \to q$	Modus ponens
$\neg q$ $p \to q$ $\therefore \neg p$	$(\neg q \wedge (p \to q)) \to \neg p$	Modus tollens
$p \to q$ $q \to r$ $\therefore p \to r$	$((p \to q) \wedge (q \to r)) \to (p \to r)$	Hypothetical syllogism
$p \vee q$ $\neg p$ $\therefore q$	$((p \vee q) \wedge \neg p) \to q$	Disjunctive syllogism

TABLE 1 **Rules of Inference.**

Rule of Inference	*Tautology*	*Name*
p $\therefore\ p \vee q$	$p \rightarrow (p \vee q)$	Addition
$p \wedge q$ $\therefore\ p$	$(p \wedge q) \rightarrow p$	Simplification
p q $\therefore\ p \wedge q$	$((p) \wedge (q)) \rightarrow (p \wedge q)$	Conjunction
$p \vee q$ $\neg p \vee r$ $\therefore\ q \vee r$	$((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$	Resolution

Rules of Inference for Propositional Logic: ***Modus Ponendo Ponens***

$$\frac{P \to Q, P}{\therefore Q}$$

- Latin for “the way that affirms by affirming”

- **Implication elimination**
- Abbreviated to **MP** or ***modus ponens***

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

- whenever $p \to q$ is true
- and p is true,
- q must also be true

Modus Ponens

$$\begin{array}{l} p \rightarrow q \\ p \\ \hline \therefore q \end{array}$$

Corresponding Tautology:

$(p \wedge (p \rightarrow q)) \rightarrow q$

Example:

Let *p* be “It is snowing.”

Let *q* be “I will study discrete math.”

“If it is snowing, then I will study discrete math.”

“It is snowing.”

“Therefore , I will study discrete math.”

Modus Ponens

- State which rule of inference is the basis of the following argument:
- “If it snows today, then we will go skiing”. “It is snowing today”, “therefore we will go skiing”.
- Let p be the proposition “It is snowing today” and q the proposition “We will go skiing” Then this argument is of the form

$$\begin{array}{l} p \rightarrow q \\ p \\ \therefore q \end{array}$$

- This is an argument that uses the Modus Ponens rule.

Modus Tollendo Tollens

$$\frac{P \to Q, \neg Q}{\therefore \neg P}$$

- Latin for "the way that denies by denying"

- **Denying the consequent**
- ***Abbreviated to modus tollens***

- in every instance in which
- p → q is true and q is false,
- p must also be false.

p	q	p → q
T	T	T
T	F	F
F	T	T
F	F	T

Modus Tollens

$$\begin{array}{l} p \rightarrow q \\ \neg q \\ \hline \therefore \neg p \end{array}$$

Corresponding Tautology:

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

Example:
Let *p* be “it is snowing.”
Let *q* be “I will study discrete math.”

“If it is snowing, then I will study discrete math.”
“I will not study discrete math.”

“Therefore , it is not snowing.”

Hypothetical Syllogism

$$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array}$$

Corresponding Tautology:
$((p \rightarrow q) \wedge (\mathrm{q} \rightarrow r)) \rightarrow (p \rightarrow r)$

Example:
Let *p* be “it snows.”
Let *q* be “I will study discrete math.”
Let *r* be “I will get an A.”

“If it snows, then I will study discrete math.”
“If I study discrete math, I will get an A.”

“Therefore , If it snows, I will get an A.”

Hypothetical Syllogism

- State which rule of inference is used in the argument:
- If it rains today, then we will not have a barbecue today. If we do not have a barbecue today, then we will have a barbecue tomorrow. Therefore, if it rains today, then we will have a barbecue tomorrow.

$$p \rightarrow q$$
$$q \rightarrow r$$
$$\therefore p \rightarrow r$$

- Hence, this argument is a hypothetical

Disjunctive Syllogism

$$\begin{array}{l} p \vee q \\ \neg p \\ \hline \therefore q \end{array}$$

Corresponding Tautology:
$(\neg p \wedge (p \vee q)) \rightarrow q$

Example:
Let *p* be “I will study discrete math.”
Let *q* be “I will study English literature.”

“I will study discrete math or I will study English literature.”
“I will not study discrete math.”

“Therefore , I will study English literature.”

Addition

$$\begin{array}{l} p \\ \hline \therefore p \vee q \end{array}$$

Corresponding Tautology:

$p \rightarrow (p \vee q)$

Example:
Let *p* be “I will study discrete math.”
Let *q* be “I will visit Las Vegas.”

“I will study discrete math.”

“Therefore, I will study discrete math or I will visit Las Vegas.”

Addition Rule

- State which rule of inference is the basis of the following argument:
- “It is below freezing now. Therefore, it is either below freezing or raining now.”
- Let p be the proposition “It is below freezing now” and q the proposition “It is raining now.” Then this argument is of the form

$$p$$
$$\therefore p \vee q$$

- This is an argument that uses the addition rule.

Simplification

$$\frac{p \wedge q}{\therefore q}$$

Corresponding Tautology:

$(p \wedge q) \rightarrow p$

Example:
Let *p* be “I will study discrete math.”
Let *q* be “I will study English literature.”

“I will study discrete math and English literature”

“Therefore, I will study discrete math.”

Simplification Rule

- State which rule of inference is the basis of the following argument:
- “It is below freezing and raining now. Therefore, it is below freezing now.”
- Let p be the proposition “It is below freezing now,” and let q be the proposition “It is raining now.” This argument is of the form

$$p \wedge q$$
$$\therefore p$$

- This argument uses the simplification rule.

Conjunction

$$\begin{array}{l} p \\ q \\ \hline \therefore p \wedge q \end{array}$$

Corresponding Tautology:
$((p) \wedge (q)) \rightarrow (p \wedge q)$

Example:
Let *p* be “I will study discrete math.”
Let *q* be “I will study English literature.”

“I will study discrete math.”
“I will study English literature.”

“Therefore, I will study discrete math and I will study English literature.”

Resolution

Resolution plays an important role in AI and is used in Prolog.

$$\begin{array}{l} \neg p \vee r \\ p \vee q \\ \hline \therefore q \vee r \end{array}$$

Corresponding Tautology:
$((\neg p \vee r) \wedge (p \vee q)) \rightarrow (q \vee r)$

Example:
Let *p* be “I will study discrete math.”
Let *r* be “I will study English literature.”
Let q be “I will study databases.”

“I will not study discrete math or I will study English literature.”
“I will study discrete math or I will study databases.”

“Therefore, I will study databases or I will English literature.”

Using the Rules of Inference to Build Valid Arguments

- A *valid argument* is a sequence of statements. Each statement is either a premise or follows from previous statements by rules of inference. The last statement is called conclusion.
- A valid argument takes the following form:

$$
\begin{array}{l}
S_1 \\
S_2 \\
\vdots \\
S_n \\
\therefore C
\end{array}
$$

Fallacies

- Arguments are based on tautologies.
- Fallacies are based on contingencies.
- The proposition $((p \rightarrow q) \wedge q) \rightarrow p$ is not a tautology, because it is false when p is false and q is true.
- This type of incorrect reasoning is called the **fallacy of affirming the conclusion**.
- Example:
- If you do every problem in this book, then you will learn discrete mathematics. You learned discrete mathematics. Therefore, you did every problem in this book.

Fallacy of denying the hypothesis.

- The proposition $((p \rightarrow q) \wedge \neg p) \rightarrow \neg q$ is not a tautology, because it is false when p is false and q is true.
- Is it correct to assume that you did not learn discrete mathematics if you did not do every problem in the book, assuming that if you do every problem in this book, then you will learn discrete mathematics?
- It is possible that you learned discrete mathematics even if you did not do every problem in this book. This incorrect argument is of the form $p \rightarrow q$ and $\neg p$ imply $\neg q$, which is an example of the fallacy of denying the hypothesis.

Using Rules of Inference

Fallacies

- **Are the following arguments correct?**
  - **Example 1 (Fallacy of affirming the conclusion)**

    Hypothesis
    - If you success, you work hard $p \rightarrow q$
    - You work hard q

    Conclusion
    - You success $\therefore\ p$

    ✗
  - **Example 2 (Fallacy of denying the hypothesis)**

    Hypothesis
    - If you success, you work hard $p \rightarrow q$
    - You do not success $\neg p$

    Conclusion
    - You do not work hard $\therefore\ \neg q$

    ✗

V. Tell the validity of each argument by choosing one of the following:

MP- Modus Ponens **LS- Law of Syllogism (Hypothetical)**
MT- Modus Tollens **LC- Law of Contrapositive**
NVC- No Valid Conclusion

a. _____

$p \rightarrow \sim r$
$\sim p$
_____
$\therefore r$

b. _____

$s \rightarrow t$
$\sim t$
_____
$\therefore \sim s$

c. _____

$r \rightarrow \sim t$
$\sim t \rightarrow v$
_____
$\therefore r \rightarrow v$

d. _____

$r \rightarrow p$
p
_____
$\therefore r$

e. _____

$\sim w \rightarrow \sim s$
$\sim w$
_____
$\therefore \sim s$

f. _____

$t \rightarrow (r \rightarrow s)$
$\sim(r \rightarrow s)$
_____
$\therefore \sim t$

g. _____

$p \rightarrow \sim r$
_____
$\therefore r \rightarrow \sim p$

h. _____

$r \rightarrow q$
$s \rightarrow q$
_____
$\therefore r \rightarrow s$

V. Tell the validity of each argument by choosing one of the following:

MP- Modus Ponens **LS- Law of Syllogism (Hypothetical)**
MT- Modus Tollens **LC- Law of Contrapositive**
NVC- No Valid Conclusion

a. NVC

$p \rightarrow \sim r$
$\sim p$

$\therefore r$

b. M.T

$(s \rightarrow t)$ ✓
$\wedge (\sim t)$

$\therefore \boxed{\sim s}$

c. LS (H.S)

$r \rightarrow \cancel{\sim} t$
$\cancel{\sim} t \rightarrow v$

$\therefore r \rightarrow v$

d. NVC

$r \rightarrow p$
p

$\therefore r$

e. M.P

$\sim w \rightarrow \sim s$ ✓
$\sim w$

$\therefore \sim s$

f. M.T

$t \rightarrow (r \rightarrow s)$ ✓
$\sim(r \rightarrow s)$

$\therefore \sim t$

g. L.C

$p \rightarrow \sim r$

$\therefore r \rightarrow \sim p$

h. N.V.C

$r \rightarrow q$
$s \rightarrow q$

$\therefore r \rightarrow s$

Comparison between Inference and Equivalence

- **Inference ($p \rightarrow q$)**
  - Meaning:
    If p, then q
  - $p \rightarrow q$ does not mean $q \rightarrow p$
  - Either inference or equivalence rules can be used
  - $p \leftrightarrow q$ implies $p \rightarrow q$
  - $\Rightarrow$ is used in proof
- **Equivalence ($p \leftrightarrow q$)**
  - Meaning:
    p is equal to q
  - $p \leftrightarrow q$ mean $q \leftrightarrow p$
  - Only equivalence rules can be used
  - $p \leftrightarrow q$ can be proved by showing $p \rightarrow q$ and $q \rightarrow p$
  - $\Leftrightarrow$ is used in proof
- Equivalence ($\leftrightarrow$) is a **more restrictive** relation than Inference ($\rightarrow$)

Using Rules of Inference

- Example 1:
  - **Given:**
    - It is not sunny this afternoon and it is colder than yesterday.
    - We will go swimming only if it is sunny
    - If we do not go swimming, then we will take a canoe trip
    - If we take a canoe trip, then we will be home by sunset
  - Can these propositions lead to the **conclusion** "We will be home by sunset" ?

Let	p:	It is sunny this afternoon
	q:	It is colder than yesterday
	r:	We go swimming
	s:	We take a canoe trip
	t:	We will be home by sunset

$\neg p \wedge q$	It is not sunny this afternoon and it is colder than yesterday
$r \rightarrow p$	We will go swimming only if it is sunny
$\neg r \rightarrow s$	If we do not go swimming, then we will take a canoe trip
$s \rightarrow t$	If we take a canoe trip, then we will be home by sunset
t	We will be home by sunset

Using Rules of Inference

Hypothesis:

$\neg p \wedge q$

$r \rightarrow p$

$\neg r \rightarrow s$

$s \rightarrow t$

Conclusion:

t

Step		Reason
1.	$\neg p \wedge q$	Premise
2.	$\neg p$	Simplification using (1)
3.	$r \rightarrow p$	Premise
4.	$\neg r$	Modus tollens using (2) and (3)
5.	$\neg r \rightarrow s$	Premise
6.	s	Modus ponens using (4) and (5)
7.	$s \rightarrow t$	Premise
8.	t	Modus ponens using (6) and (7)

Therefore, the propositions can lead to the conclusion
We will be home by sunset

Using Rules of Inference

- Or, another presentation method:

Hypothesis:

$\neg p \wedge q$

$r \rightarrow p$

$\neg r \rightarrow s$

$s \rightarrow t$

Conclusion:

t

$(\neg p \wedge q) \wedge (r \rightarrow p) \wedge (\neg r \rightarrow s) \wedge (s \rightarrow t)$

$\Rightarrow \neg p \wedge (r \rightarrow p) \wedge (\neg r \rightarrow s) \wedge (s \rightarrow t)$ By Simplification

$\Rightarrow \neg r \wedge (\neg r \rightarrow s) \wedge (s \rightarrow t)$ By Modus Tollens

$\Rightarrow s \wedge (s \rightarrow t)$ By Modus Ponens

$\Rightarrow t$ By Modus Ponens

EXAMPLE #2

- **Given:**
  - If you send me an e-mail message, then I will finish writing the program
  - If you do not send me an e-mail message, then I will go to sleep early
  - If I go to sleep early, then I will wake up feeling refreshed
- Can these propositions lead to the **conclusion** "If I do not finish writing the program, then I will wake up feeling refreshed."

Let p: you send me an e-mail message
q: I will finish writing the program
r: I will go to sleep early
s: I will wake up feeling refreshed

$p \rightarrow q$

- If you send me an e-mail message, then I will finish writing the program

$\neg p \rightarrow r$

- If you do not send me an e-mail message, then I will go to sleep early

$r \rightarrow s$

- If I go to sleep early, then I will wake up feeling refreshed

---

$\neg q \rightarrow s$

- If I do not finish writing the program, then I will wake up feeling refreshed

Hypothesis:

$p \rightarrow q$

$\neg p \rightarrow r$

$r \rightarrow s$

Conclusion:

$\neg q \rightarrow s$

Step		Reason
1.	$p \rightarrow q$	Premise
2.	$\neg q \rightarrow \neg p$	Contrapositive of (1)
3.	$\neg p \rightarrow r$	Premise
4.	$\neg q \rightarrow r$	Hypothetical Syllogism using (2) and (3)
5.	$r \rightarrow s$	Premise
6.	$\neg q \rightarrow s$	Hypothetical Syllogism using (4) and (5)

Therefore, the propositions can lead to the conclusion

If I do not finish writing the program,

then I will wake up feeling refreshed

- Or, another presentation method:

Hypothesis:

$p \to q$

$\neg p \to r$

$r \to s$

Conclusion:

$\neg q \to s$

$$\boxed{(p \to q)} \wedge (\neg p \to r) \wedge (r \to s)$$

$$\Leftrightarrow \boxed{(\neg q \to \neg p) \wedge (\neg p \to r)} \wedge (r \to s)$$ Contrapositive

$$\Rightarrow \boxed{(\neg q \to r) \wedge (r \to s)}$$ By Hypothetical Syllogism

$$\Rightarrow (\neg q \to s)$$ By Hypothetical Syllogism

Valid Arguments

Example 3: From the single proposition

$$p \wedge (p \rightarrow q)$$

Show that *q* is a conclusion.

Solution:

Step	**Reason**
1. $p \wedge (p \rightarrow q)$	Premise
2. p	Conjunction using (1)
3. $p \rightarrow q$	Conjunction using (1)
4. q	Modus Ponens using (2) and (3)

Valid Arguments (Predicate Logic)

TABLE 2 Rules of Inference for Quantified Statements.

Rule of Inference	*Name*
$\forall x\, P(x)$ $\therefore\ P(c)$	Universal instantiation
$P(c)$ for an arbitrary 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

Universal Instantiation

- Universal instantiation is the rule of inference used to conclude that P(c) is true, where c is a particular member of the domain, given the premise $\forall x\, P(x)$.
- •Universal instantiation is used when we conclude from the statement “All women are wise” that “Lisa is wise,” where Lisa is a member of the domain of all women.

Universal Generalization

- Universal generalization is the rule of inference that states that $\forall x P(x)$ is true, given the premise that $P(c)$ is true for all elements c in the domain.
- •Universal generalization is used when we show that $\forall x\ P(x)$ is true by taking an arbitrary element c from the domain and showing that $P(c)$ is true.
- •Universal generalization is used implicitly in many proofs in mathematics and is seldom mentioned explicitly. However, the error of adding unwarranted assumptions about the arbitrary element c when universal generalization is used is all too common in incorrect reasoning.

Existential Instantiation

- Existential instantiation is the rule that allows us to conclude that there is an element c in the domain for which P(c) is true if we know that $\exists x\ P(x)$ is true.
- We cannot select an arbitrary value of c here, but rather it must be a c for which P(c) is true.
- Usually we have no knowledge of what c is, only that it exists. Because it exists, we may give it a name (c) and continue our argument.

Existential Generalization

- Existential generalization is the rule of inference that is used to conclude that $\exists x\ P(x)$ is true when a particular element c with P(c) true is known.
- That is, if we know one element c in the domain for which P(c) is true, then we know that $\exists x\ P(x)$ is true.

Basic Structures: Sets and Functions

Chapter 2

Sets

Section 2.1

Section Summary

- Definition of sets
- Describing Sets
  - Roster Method
  - Set-Builder Notation
- Some Important Sets in Mathematics
- Empty Set and Universal Set
- Subsets and Set Equality
- Cardinality of Sets
- Tuples
- Cartesian Product

Introduction

- Sets are one of the basic building blocks for the types of objects considered in discrete mathematics.
  - Important for counting.
  - Programming languages have set operations.
- Set theory is an important branch of mathematics.
  - Many different systems of axioms have been used to develop set theory.
  - Here we are not concerned with a formal set of axioms for set theory. Instead, we will use what is called naïve set theory.

Sets

- A *set* is an unordered collection of objects.
  - the students in this class
  - the chairs in this room
- 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 $a \notin A$

Describing a Set: Roster Method

- $S = \{a,b,c,d\}$
- Order not important

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

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

Roster Method

- Set of all vowels in the English alphabet:

$$V = \{\mathrm{a,e,i,o,u}\}$$

- Set of all odd positive integers less than 10:

$$O = \{1,3,5,7,9\}$$

- Set of all positive integers less than 100:

$$S = \{1,2,3,.....,99\}$$

- Set of all integers less than 0:

$$S = \{...., -3,-2,-1\}$$

Some Important Sets

$\mathbf{N}$ = *natural numbers* = $\{0,1,2,3....\}$

$\mathbf{Z}$ = *integers* = $\{...,-3,-2,-1,0,1,2,3,...\}$

$\mathbf{Z}^{+}$ = *positive integers* = $\{1,2,3,.....\}$

$\mathbf{R}$ = set of *real numbers*

$\mathbf{R}^{+}$ = set of *positive real numbers*

$\mathbf{C}$ = set of *complex numbers*.

Q = set of rational numbers

Set-Builder Notation

- Specify the property or properties that all members must satisfy:

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

  $O = \{x \mid x \text{ is an odd positive integer less than } 10\}$

  $O = \{x \in \mathbf{Z}^{+} \mid x \text{ is odd and } x < 10\}$
- A predicate may be used:

  $$S = \{x \mid \mathrm{P}(x)\}$$
- Example: $S = \{x \mid \mathrm{Prime}(x)\}$
- Positive rational numbers*:*

  $\mathbf{Q}^{+} = \{x \in \mathbf{R} \mid x = p/q, \text{ for some positive integers } p,q\}$

Universal Set and Empty Set

- The *universal set U* is the set containing everything currently under consideration.
  - Sometimes implicit
  - Sometimes explicitly stated.
  - Contents depend on the context.
- The empty set is the set with no elements. Symbolized $\emptyset$, but {} also used.

Venn Diagram

John Venn (1834-1923)
Cambridge, UK

Some things to remember

- Sets can be elements of sets.

$$\{\{1,2,3\},a,\ \{b,c\}\}$$

$$\{\mathbf{N},\mathbf{Z},\mathbf{Q},\mathbf{R}\}$$

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

$$\emptyset \neq \{\ \emptyset\ \}$$

Set Equality

Definition: Two sets are *equal* if and only if they have the same elements.

- Therefore if A and B are sets, then A and B are equal if and only if $\forall x(x \in A \leftrightarrow x \in B)$
.
- We write $A = B$ if A and B are equal sets.

$$\{1,3,5\} = \{3, 5, 1\}$$

$$\{1,5,5,5,3,3,1\} = \{1,3,5\}$$

Subsets

Definition: 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 $a \in \emptyset$ is always false, $\emptyset \subseteq S$,for every set *S*.
2. Because $a \in S \rightarrow a \in S$, $S \subseteq S$, for every set *S*.

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, $A \nsubseteq B$, find an element $x \in A$ with $x \notin B$. (Such an x is a counterexample to the claim that $x \in A$ implies $x \in B$.)

Examples:

1. The set of all computer science majors at your school is a subset of all students at your school.
2. The set of integers with squares less than 100 is not a subset of the set of nonnegative integers.

Another look at Equality of Sets

- Recall that two sets A and B are *equal*, denoted by $A = B$, iff

$$\forall x(x \in A \leftrightarrow x \in B)$$

- Using logical equivalences we have that $A = B$ iff

$$\forall x[(x \in A \rightarrow x \in B) \wedge (x \in B \rightarrow x \in A)]$$

- This is equivalent to

$$A \subseteq B \quad \text{and} \quad B \subseteq A$$

Proper Subsets

Definition: 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

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

is true.

Venn Diagram

Set Cardinality

Definition: If there are exactly n distinct elements in *S* where *n* is a nonnegative integer, we say that *S* is *finite*. Otherwise it is *infinite*.

Definition: The *cardinality* of a finite set *A*, denoted by |*A*|, is the number of (distinct) elements of *A*.

Examples:

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

Definition: The set of all subsets of a set *A*, denoted P**(***A***)**, is called the *power set* of *A*.

Example: If *A* = {a,b} then

P(A) = {ø, {a},{b},{a,b}}

- If a set has *n* elements, then the cardinality of the power set is 2^n. (In Chapters 5 and 6, we will discuss different ways to show this.)

Tuples

- The *ordered n-tuple* $(a_1, a_2, \ldots\ldots, a_n)$ 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

Cartesian Product

René Descartes
(1596-1650)

Definition: 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 $b \in B$.

$$A \times B = \{(a,b) | a \in A \wedge b \in B\}$$

Example:

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

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

- **Definition**: A subset R of the Cartesian product $A \times B$ is called a *relation* from the set A to the set B. (Relations will be covered in depth in Chapter 9.)

Cartesian Product

Definition: The cartesian products of the sets $A_1, A_2, \ldots\ldots, A_n$, denoted by $A_1 \times A_2 \times \ldots\ldots \times A_n$, is the set of ordered *n*-tuples $(a_1, a_2, \ldots\ldots, a_n)$ where a_i belongs to A_i for $i = 1, \ldots n$.

$$A_1 \times A_2 \times \cdots \times A_n = \{(a_1, a_2, \ldots, a_n) | a_i \in A_i \text{ for } i = 1, 2, \ldots n\}$$

Example: What is $A \times B \times C$ where $A = \{0,1\}$, $B = \{1,2\}$ and $C = \{0,1,2\}$

Solution: $A \times B \times C$ **=** {(0,1,0), (0,1,1), (0,1,2),(0,2,0), (0,2,1), (0,2,2),(1,1,0), (1,1,1), (1,1,2), (1,2,0), (1,2,1), (1,1,2)}

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 $\{x \in D | P(x)\}$

- **Example**: The truth set of *P*(*x*) where the domain is the integers and *P*(*x*) is "|*x*| = 1" is the set $\{-1,1\}$

Set Operations

Section 2.2

Section Summary

- Set Operations
  - Union
  - Intersection
  - Complementation
  - Difference
- More on Set Cardinality
- Set Identities
- Proving Identities
- Membership Tables

Boolean Algebra

- Propositional calculus and set theory are both instances of an algebraic system called a *Boolean Algebra*. This is discussed in Chapter 12.
- The operators in set theory are analogous to the corresponding operator in propositional calculus.
- As always there must be a universal set U. All sets are assumed to be subsets of U.

Union

- **Definition**: Let *A* and *B* be sets. The *union* of the sets *A* and *B*, denoted by *A* ∪ *B*, is the set:

$$\{x | x \in A \vee x \in B\}$$

- **Example**: What is $\{1,2,3\} \cup \{3, 4, 5\}$?

Solution: $\{1,2,3,4,5\}$

Venn Diagram for $A \cup B$

Intersection

- **Definition**: The *intersection* of sets *A* and *B*, denoted by $A \cap B$, is

$$\{x | x \in A \wedge x \in B\}$$

- Note if the intersection is empty, then *A* and *B* are said to be *disjoint*.
- **Example**: What is? $\{1,2,3\} \cap \{3,4,5\}$?

  Solution: $\{3\}$
- **Example:** What is?

$$\{1,2,3\} \cap \{4,5,6\} \text{ ?}$$

Solution: $\emptyset$

Venn Diagram for $A \cap B$

Complement

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

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

(The complement of A is sometimes denoted by A^c.)

Example: If U is the positive integers less than 100, what is the complement of $\{x \mid x > 70\}$

Solution: $\{x \mid x \leq 70\}$

Venn Diagram for Complement

Difference

- **Definition**: Let A and B be sets. The *difference* of A and B, denoted by $A - B$, is the set containing the elements of A that are not in B. The difference of A and B is also called the complement of B with respect to A.

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

Venn Diagram for $A - B$

The Cardinality of the Union of Two Sets

Inclusion-Exclusion

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

Venn Diagram for A, B, $A \cap B$, $A \cup B$

Example: Let A be the math majors in your class and B be the CS majors. To count the number of students who are either math majors or CS majors, add the number of math majors and the number of CS majors, and subtract the number of joint CS/math majors.

We will return to this principle in Chapter 6 and Chapter 8 where we will derive a formula for the cardinality of the union of n sets, where n is a positive integer.

Review Questions

Example: $U = \{0,1,2,3,4,5,6,7,8,9,10\}$ $A = \{1,2,3,4,5\}$, $B = \{4,5,6,7,8\}$

1. $A \cup B$

   Solution: $\{1,2,3,4,5,6,7,8\}$
2. $A \cap B$

   Solution: $\{4,5\}$
3. $\bar{A}$

   Solution: $\{0,6,7,8,9,10\}$
4. $\bar{B}$

   Solution: $\{0,1,2,3,9,10\}$
5. $A - B$

   Solution: $\{1,2,3\}$
6. $B - A$

   Solution: $\{6,7,8\}$

Symmetric Difference (*optional*)

Definition: The *symmetric difference* of **A** and **B**, denoted by $A \oplus B$ is the set

$$(A - B) \cup (B - A)$$

Example:

$U = \{0,1,2,3,4,5,6,7,8,9,10\}$

$A = \{1,2,3,4,5\}$ $B = \{4,5,6,7,8\}$

What is:

- **Solution**: $\{1,2,3,6,7,8\}$

Venn Diagram

Set Identities

- Identity laws

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

- Domination laws

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

- Idempotent laws

$$A \cup A = A \qquad A \cap A = A$$

- Complementation law

$$\overline{(\overline{A})} = A$$

Continued on next slide ▯

Set Identities

- Commutative laws

$$A \cup B = B \cup A \qquad A \cap B = B \cap A$$

- Associative laws

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

- Distributive laws

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

Continued on next slide ▯

Set Identities

- De Morgan's laws

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

- Absorption laws

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

- Complement laws

$$A \cup \overline{A} = U \qquad A \cap \overline{A} = \emptyset$$

Proving Set Identities

- Different ways to prove set identities:
  1. Prove that each set (side of the identity) is a subset of the other.
  2. Use set builder notation and propositional logic.
  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.

Proving Set Identities

Example:

Let A, B, and C be sets. Show that

$$\overline{A \cup (B \cap C)} = (\overline{C} \cup \overline{B}) \cap \overline{A}.$$

Solution: We have

$$\begin{aligned}
\overline{A \cup (B \cap C)} &= \overline{A} \cap (\overline{B \cap C}) && \text{by the first De Morgan law} \\
&= \overline{A} \cap (\overline{B} \cup \overline{C}) && \text{by the second De Morgan law} \\
&= (\overline{B} \cup \overline{C}) \cap \overline{A} && \text{by the commutative law for intersections} \\
&= (\overline{C} \cup \overline{B}) \cap \overline{A} && \text{by the commutative law for unions.}
\end{aligned}$$

Proof of Second De Morgan Law

Example: Prove that $\overline{A \cap B} = \overline{A} \cup \overline{B}$

Solution: We prove this identity by showing that:

1) $\overline{A \cap B} \subseteq \overline{A} \cup \overline{B}$ and

2) $\overline{A} \cup \overline{B} \subseteq \overline{A \cap B}$

Continued on next slide ▯

Proof of Second De Morgan Law

These steps show that: $\overline{A \cap B} \subseteq \overline{A} \cup \overline{B}$

$x \in \overline{A \cap B}$	by assumption
$x \notin A \cap B$	defn. of complement
$\neg((x \in A) \wedge (x \in B))$	defn. of intersection
$\neg(x \in A) \vee \neg(x \in B)$	1st De Morgan Law for Prop Logic
$x \notin A \vee x \notin B$	defn. of negation
$x \in \overline{A} \vee x \in \overline{B}$	defn. of complement
$x \in \overline{A} \cup \overline{B}$	defn. of union

Continued on next slide ▯

Proof of Second De Morgan Law

These steps show that:

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

$x \in \overline{A} \cup \overline{B}$	by assumption
$(x \in \overline{A}) \vee (x \in \overline{B})$	defn. of union
$(x \notin A) \vee (x \notin B)$	defn. of complement
$\neg(x \in A) \vee \neg(x \in B)$	defn. of negation
$\neg((x \in A) \wedge (x \in B))$	by 1st De Morgan Law for Prop Logic
$\neg(x \in A \cap B)$	defn. of intersection
$x \in \overline{A \cap B}$	defn. of complement

Set-Builder Notation: Second De Morgan Law

$$
\begin{array}{rcll}
\overline{A \cap B} & = & \{x | x \notin A \cap B\} & \text{by defn. of complement} \\
 & = & \{x | \neg(x \in (A \cap B))\} & \text{by defn. of does not belong symbol} \\
 & = & \{x | \neg(x \in A \wedge x \in B\} & \text{by defn. of intersection} \\
 & = & \{x | \neg(x \in A) \vee \neg(x \in B)\} & \text{by 1st De Morgan law} \\
 & & & \text{for Prop Logic} \\
 & = & \{x | x \notin A \vee x \notin B\} & \text{by defn. of not belong symbol} \\
 & = & \{x | x \in \overline{A} \vee x \in \overline{B}\} & \text{by defn. of complement} \\
 & = & \{x | x \in \overline{A} \cup \overline{B}\} & \text{by defn. of union} \\
 & = & \overline{A} \cup \overline{B} & \text{by meaning of notation}
\end{array}
$$

Membership Table

Example: Construct a membership table to show that the distributive law holds.

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

Solution:

A	B	C	$B \cap C$	$A \cup (B \cap C)$	$A \cup B$	$A \cup C$	$(A \cup B) \cap (A \cup C)$
1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	0	0	0	1	1	1	1
0	1	1	1	1	1	1	1
0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	0

Functions

Section 2.3

Section Summary

- Definition of a Function.
  - Domain, Codomain
  - Image, Pre image
- Injection, Surjection, Bijection
- Inverse Function
- Function Composition
- Graphing Functions
- Floor, Ceiling, Factorial

Functions

Definition: Let A and B be nonempty sets. A *function* f from A to B, denoted $f: A \to B$ 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*.

Functions

Definition: Let A and B be nonempty sets. A *function* f from A to B, denoted $f: A \to B$ 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 ele

Functions

- A function f: $A \to B$ 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 $a \in A$.

$$\forall x[x \in A \to \exists y[y \in B \wedge (x, y) \in f]]$$

and $$\forall x, y_1, y_2[[(x, y_1) \in f \wedge (x, y_2)] \to y_1 = y_2]$$

Functions

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

- 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.

Equal Functions

- Two functions are **equal** when they
  - have the same domain,
  - have the same codomain,
  - map each element of their common domain to the same element in their common codomain.
- If we change either the domain or the codomain of a function, then we obtain a different function.
- If we change the mapping of elements, then we also obtain a different function.

Representing Functions

- Functions may be specified in different ways:
  - An explicit statement of the assignment.

    Students and grades example.
  - A formula.

    $f(x) = x + 1$
  - A computer program.
    - A Java program that when given an integer *n*, produces the *n*th Fibonacci Number (covered in the next section and also inChapter 5).

Activity Time

What are the domain, codomain, and range of the following function

FIGURE 1 Assignment of Grades in a Discrete Mathematics Class.

Solution

- Let G be the function that assigns a grade to a student in our discrete mathematics class.
- Note that $G(\text{Adams}) = A$, for instance.
- The domain of G is the set {Adams, Chou, Goodfriend, Rodriguez, Stevens},
- The codomain is the set $\{A,B,C,D,F\}$.
- The range of G is the set $\{A,B,C, F\}$,

Questions

$f(\text{a}) = ?$ z

The image of d is ? z

The domain of f is ? A

The codomain of f is ? B

The preimage of y is ? b

$f(A) = ?$

The preimage(s) of z is (are) ? {a,c,d}

Question on Functions and Sets

- If $f : A \to B$ and S is a subset of A, then

$$f(S) = \{f(s) | s \in S\}$$

f {a,b,c,} is ? {y,z}

f {c,d} is ? {z }

Injections

Definition: A function f is said to be *one-to-one* , or *injective*, if and only if $f(a) = f(b)$ implies that $a = b$ for all a and b in the domain of f. A function is said to be an *injection* if it is one-to-one.

Surjections

Definition: A function f from A to B is called *onto* or *surjective*, if and only if for every element $b \in B$ there is an element $a \in A$ with $f(a) = b$. A function f is called a *surjection* if it is onto.

Bijections

Definition: A function f is a *one-to-one correspondence*, or a *bijection*, if it is both one-to-one and onto (surjective and injective).

FIGURE 5 Examples of Different Types of Correspondences.

real-valued

- A function is called **real-valued** if its codomain is the set of real numbers.

- Let $f1$ and $f2$ be functions from A to $\mathbf{R}$.
- Then $f1 + f2$ and $f1f2$ are also functions from A to $\mathbf{R}$ defined for all $x \in A$ by
  - $(f1 + f2)(x) = f1(x) + f2(x)$,
  - $(f1f2)(x) = f1(x)f2(x)$.

real-valued Functions

Examples:

- **Assume**
  - $f1(x) = x - 1$
  - $f2(x) = x^3 + 1$

  then
  - $(f1 + f2)(x) = x^3 + x$
  - $(f1 * f2)(x) = x^4 - x^3 + x - 1.$

Showing that *f* is one-to-one or onto

Suppose that $f : A \to B$.

To show that f is injective Show that if $f(x) = f(y)$ for arbitrary $x, y \in A$ with $x \neq y$, then $x = y$.

To show that f is not injective Find particular elements $x, y \in A$ such that $x \neq y$ and $f(x) = f(y)$.

To show that f is surjective Consider an arbitrary element $y \in B$ and find an element $x \in A$ such that $f(x) = y$.

To show that f is not surjective Find a particular $y \in B$ such that $f(x) \neq y$ for all $x \in A$.

Showing that f is one-to-one or onto

Example 1: Let f be the function from $\{a,b,c,d\}$ to $\{1,2,3\}$ defined by $f(a) = 3$, $f(b) = 2$, $f(c) = 1$, and $f(d) = 3$. Is f an onto function?

Solution:

Yes, f is onto since all three elements of the codomain are images of elements in the domain. If the codomain were changed to $\{1,2,3,4\}$, f would not be onto.

Example 2:

Determine whether the function $f(x) = x^2$ from the set of integers to the set of integers is one-to-one.

Solution:

The function $f(x) = x^2$ is not one-to-one because, for instance, $f(1) = f(-1) = 1$, but $1 \quad -1$.

Increasing/ decreasing functions

- A function f whose domain and codomain are subsets of the set of real numbers is called *increasing* if $f(x) \leq f(y)$, and *strictly increasing* if $f(x) < f(y)$, whenever $x < y$ and x and y are in the domain of f.
- Similarly, f is called *decreasing* if $f(x) \geq f(y)$, and *strictly decreasing* if $f(x) > f(y)$, whenever $x < y$ and x and y are in the domain of f.

Increasing/ decreasing functions

- A function f is
  - increasing if $\forall x \forall y (x < y \rightarrow f(x) \leq f(y))$,
  - strictly increasing if $\forall x \forall y (x < y \rightarrow f(x) < f(y))$,
  - decreasing if $\forall x \forall y (x < y \rightarrow f(x) \geq f(y))$,
  - strictly decreasing if $\forall x \forall y (x < y \rightarrow f(x) > f(y))$,

where the universe of discourse is the domain of f.

Increasing/ decreasing functions

Example:

- Let $g : \mathbf{R} \to \mathbf{R}$, where $g(x) = 2x - 1$. Is it increasing ?

Proof .

For $x>y$ holds $2x > 2y$ and subsequently $2x-1 > 2y-1$

Thus g is strictly increasing.

Relationship with one-to-one

- A function that is either strictly increasing or strictly decreasing must be one-to-one.

  Why?
- One-to-one function: A function is one-to-one if and only if $f(x) \neq f(y)$, whenever $x \neq y$.

- A function that is increasing, but not strictly increasing, or decreasing, but not strictly decreasing, is not one-to-one.

The *identity function*

- Let A be a set. The *identity function* on A is the function $i\,A : A \to A$, where

$$i\,A(x) = x$$

- for all $x \in A$.

- The function $i\,A$ is one-to-one and onto, so it is a bijection.

The *identity function*

Example:

- *Let* $A = \{1, 2, 3\}$

Solution:

$i\ A(1) = 1$

$i\ A(2) = 2$

$i\ A(3) = 3$

Inverse Functions

Definition: Let f be a bijection from A to B. Then the *inverse* of f, denoted f^{-1}, is the function from B to A defined as

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

No inverse exists unless f is a bijection. Why?

Inverse Functions

Questions

Example 1: Let f be the function from $\{a,b,c\}$ to $\{1,2,3\}$ such that $f(a) = 2$, $f(b) = 3$, and $f(c) = 1$. Is f invertible and if so what is its inverse?

Solution: The function f is invertible because it is a one-to-one correspondence. The inverse function f^{-1} reverses the correspondence given by f, so $f^{-1}(1) = c$, $f^{-1}(2) = a$, and $f^{-1}(3) = b$.

Questions

Example 2: Let f: **Z** → **Z** be such that $f(x) = x + 1$. Is f invertible, and if so, what is its inverse?

Solution: The function f is invertible because it is a one-to-one correspondence. The inverse function f^{-1} reverses the correspondence so $f^{-1}(y) = y - 1$.

Questions

Example 3: Let *f*: **R** $\rightarrow$ **R** be such that $f(x) = x^2$. Is *f* invertible, and if so, what is its inverse?

Solution: The function *f* is not invertible because it is not one-to-one .

Composition

- **Definition**: Let f: $B \to C$, g: $A \to B$. The *composition of* f *with* g, denoted $f \circ g$ is the function from A to C defined by

$$f \circ g(x) = f(g(x))$$

Composition

Composition

Example 1: If $f(x) = x^2$ and $g(x) = 2x + 1$, then

$$f(g(x)) = (2x+1)^2$$

and

$$g(f(x)) = 2x^2 + 1$$

Composition Questions

Example 2: Let g be the function from the set $\{a,b,c\}$ to itself such that $g(a) = b$, $g(b) = c$, and $g(c) = a$. Let f be the function from the set $\{a,b,c\}$ to the set $\{1,2,3\}$ such that $f(a) = 3$, $f(b) = 2$, and $f(c) = 1$.

What is the composition of f and g, and what is the composition of g and f.

Solution: The composition $f \circ g$ is defined by

$f \circ g\ (a) = f(g(a)) = f(b) = 2.$

$f \circ g\ (b) = f(g(b)) = f(c) = 1.$

$f \circ g\ (c) = f(g(c)) = f(a) = 3.$

Note that $g \circ f$ is not defined, because the range of f is not a subset of the domain of g.

Composition Questions

Example 2: Let f and g be functions from the set of integers to the set of integers defined by $f(x) = 2x + 3$ and $g(x) = 3x + 2$.

What is the composition of f and g, and also the composition of g and f ?

Solution:

$f \circ g\ (x) = f(g(x)) = f(3x + 2) = 2(3x + 2) + 3 = 6x + 7$

$g \circ f\ (x) = g(f(x)) = g(2x + 3) = 3(2x + 3) + 2 = 6x + 11$

Graphs of Functions

- Let f be a function from the set A to the set B. The *graph* of the function f is the set of ordered pairs $\{(a,b) \mid a \in A \text{ and } f(a) = b\}$.

Graph of $f(n) = 2n + 1$ from Z to Z

Graph of $f(x) = x^2$ from Z to Z

Some Important Functions

- The *floor* function, denoted

$$f(x) = \lfloor x \rfloor$$

is the largest integer less than or equal to x.

- The *ceiling* function, denoted

$$f(x) = \lceil x \rceil$$

is the smallest integer greater than or equal to x

Example: $\lceil 3.5 \rceil = 4$ $\lfloor 3.5 \rfloor = 3$

$\lceil -1.5 \rceil = -1$ $\lfloor -1.5 \rfloor = -2$

Floor and Ceiling Functions

Graph of (a) Floor and (b) Ceiling Functions

Floor Functions

- Determine whether the function from R to Z is Injective OR Surjective.

  $f(a) =$

Solution:

- It is Surjective (onto function). This can be shown by an example; $f(0) = 0$, and $f(1) = 0$.

Ceiling Functions

- Determine whether the function from R to Z is Injective OR Surjective.

  $f(a) = \lceil a/2 \rceil$

Solution:

- It is Surjective (onto function). This can be shown by an example; $f(1) = 1$, and $f(2) = 1$.

Floor and Ceiling Functions

TABLE 1 Useful Properties of the Floor and Ceiling Functions.
(n is an integer, x is a real number)

(1a)	$\lfloor x \rfloor = n$ if and only if $n \leq x < n + 1$
(1b)	$\lceil x \rceil = n$ if and only if $n - 1 < x \leq n$
(1c)	$\lfloor x \rfloor = n$ if and only if $x - 1 < n \leq x$
(1d)	$\lceil x \rceil = n$ if and only if $x \leq n < x + 1$
(2)	$x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$
(3a)	$\lfloor -x \rfloor = -\lceil x \rceil$
(3b)	$\lceil -x \rceil = -\lfloor x \rfloor$
(4a)	$\lfloor x + n \rfloor = \lfloor x \rfloor + n$
(4b)	$\lceil x + n \rceil = \lceil x \rceil + n$

Proving Properties of Functions

Example: Prove that x is a real number, then

$$\lfloor 2x \rfloor = \lfloor x \rfloor + \lfloor x + 1/2 \rfloor$$

Solution: Let $x = n + \varepsilon$, where n is an integer and $0 \le \varepsilon < 1$.

Case 1: $\varepsilon < ½$

- $2x = 2n + 2\varepsilon$ and $\lfloor 2x \rfloor = 2n$, since $0 \le 2\varepsilon < 1$.
- $\lfloor x + 1/2 \rfloor = n$, since $x + ½ = n + (1/2 + \varepsilon)$ and $0 \le ½ + \varepsilon < 1$.
- Hence, $\lfloor 2x \rfloor = 2n$ and $\lfloor x \rfloor + \lfloor x + 1/2 \rfloor = n + n = 2n$.

Case 2: $\varepsilon \ge ½$

- $2x = 2n + 2\varepsilon = (2n + 1) + (2\varepsilon - 1)$ and $\lfloor 2x \rfloor = 2n + 1$, since $0 \le 2\varepsilon - 1 < 1$.
- $\lfloor x + 1/2 \rfloor = \lfloor n + (1/2 + \varepsilon) \rfloor = \lfloor n + 1 + (\varepsilon - 1/2) \rfloor = n + 1$ since $0 \le \varepsilon - 1/2 < 1$.
- Hence, $\lfloor 2x \rfloor = 2n + 1$ and $\lfloor x \rfloor + \lfloor x + 1/2 \rfloor = n + (n + 1) = 2n + 1$. ◀

Factorial Function

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

$f(n) = 1 \bullet 2 \bullet\bullet\bullet (n-1) \bullet n,$ $\qquad f(0) = 0! = 1$

Examples:

$f(1) = 1! = 1$

$f(2) = 2! = 1 \bullet 2 = 2$

$f(6) = 6! = 1 \bullet 2 \bullet 3 \bullet 4 \bullet 5 \bullet 6 = 720$

$f(20) = 2{,}432{,}902{,}008{,}176{,}640{,}000.$

Stirling's Formula:

$$n! \sim \sqrt{2\pi n}(n/e)^n$$

$$f(n) \sim g(n) \doteq lim_{n\rightarrow\infty} f(n)/g(n) = 1$$

Sequences and Series(Sums)

Scanned with CamScanner

Section Summary

- Sequences.
  - Examples: Geometric Progression, Arithmetic Progression
- Recurrence Relations
  - Example: Fibonacci Sequence
- Summations

Scanned with CamScanner

Introduction

- Sequences are ordered lists of elements.
- EXAMPLES:
  - 1, 2, 3, 5, 8
  - 1, 3, 9, 27, 81, ...
  - 1, 2, 3, 4, 5, ...
  - 4, 8, 12, 16, 20,...
  - 2, 4, 8, 16, 32, ...
  - 1, 1/2, 1/3, 1/4, 1/5, ...
  - 1, 4, 9, 16, 25, ...
  - 1, -1, 1, -1, 1, -1, ...
- Sequences arise throughout mathematics, computer science, and in many other disciplines, ranging from botany to music.
- We will introduce the terminology to represent sequences and sums of the terms in the sequences.

Scanned with CamScanner

SEQUENCES IN COMPUTER PROGRAMMING:

- An important data type in computer programming consists of finite sequences known as one-dimensional arrays; a single variable in which a sequence of variables may be stored.

EXAMPLE:

- The names of k students in a class may be represented by an array of k elements “name” as:

name [0], name[1], name[2], …, name[k-1]

Scanned with CamScanner

Sequences

Definition: A *sequence* is a function from a subset of the integers (usually either the set {0, 1, 2, 3, 4, ...} or {1, 2, 3, 4, ...}) to a set S.

- The notation a_n is used to denote the image of the integer n. We can think of a_n as the equivalent of $f(n)$ where f is a function from {0,1,2,.....} to S. We call a_n a *term* of the sequence.

OR

A sequence is just a list of elements usually written in a row.

Scanned with CamScanner

Sequences

Example: Consider the sequence $\{a_n\}$ where

$$a_n = \frac{1}{n} \qquad \{a_n\} = \{a_1, a_2, a_3, \ldots\}$$

$$1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4} \ldots$$

Scanned with CamScanner

Sequences

EXAMPLE:

Write the first four terms of the sequence defined by the formula: $b_j = 1 + 2^j$, for all integers $j \geq 0$

SOLUTION:

- $b_0 = 1 + 2^0 = 1 + 1 = 2$
- $b_1 = 1 + 2^1 = 1 + 2 = 3$
- $b_2 = 1 + 2^2 = 1 + 4 = 5$
- $b_3 = 1 + 2^3 = 1 + 8 = 9$

REMARK:

The formula $b_j = 1 + 2^j$, for all integers $j \geq 0$ defines an infinite sequence having infinite number of values.

Scanned with CamScanner

Sequences

- **EXERCISE:**

Compute the first six terms of the sequence defined by the formula

- $C_n = 1 + (-1)^n$ for all integers $n \geq 0$

SOLUTION :

- $C_0 = 1 + (-1)^0 = 1 + 1 = 2$
- $C_1 = 1 + (-1)^1 = 1 + (-1) = 0$
- $C_2 = 1 + (-1)^2 = 1 + 1 = 2$
- $C_3 = 1 + (-1)^3 = 1 + (-1) = 0$
- $C_4 = 1 + (-1)^4 = 1 + 1 = 2$
- $C_5 = 1 + (-1)^5 = 1 + (-1) = 0$

REMARK:

1) If n is even, then $C_n = 2$ and if n is odd, then $C_n = 0$. Hence, the sequence oscillates endlessly between 2 and 0.

2) An infinite sequence may have only a finite number of values.

Scanned with CamScanner

Sequences

EXAMPLE:

Write the first four terms of the sequence defined by

$$C_n = \frac{(-1)^n n}{n+1} \quad \text{for all integers } n \geq 1$$

SOLUTION:

$$C_1 = \frac{(-1)^1(1)}{1+1} = \frac{-1}{2}, C_2 = \frac{(-1)^2(2)}{2+1} = \frac{2}{3}, C_3 = \frac{(-1)^3(3)}{3+1} = \frac{-3}{4}$$

$$And \quad fourth \ \ term \ is C_4 = \frac{(-1)^4(4)}{4+1} = \frac{4}{5}$$

REMARK: A sequence whose terms alternate in sign is called an alternating sequence.

Scanned with CamScanner

Sequences

Find explicit formulas for sequences with the initial terms given:

1) 0, 1, -2, 3, -4, 5, …

SOLUTION:

$a_n = (-1)^{n+1} n$ for all integers $n \geq 0$

2) $1 - \frac{1}{2}, \frac{1}{2} - \frac{1}{3}, \frac{1}{3} - \frac{1}{4}, \frac{1}{4} - \frac{1}{5}, \ldots$

SOLUTION:

$b_k = \frac{1}{k} - \frac{1}{k+1}$ for all integers $n \geq 1$

Scanned with CamScanner

Sequences

3) 2, 6, 12, 20, 30, 42, 56, . . .

SOLUTION:

$$C_n = n\,(n + 1) \text{ for all integers } n \geq 1$$

4) 1/4, 2/9, 3/16, 4/25, 5/36, 6/49, . . .

SOLUTION:

$$d_i = \frac{i}{(i+1)^2} \quad \text{for all integers} \quad i \geq 1$$

OR

$$d_j = \frac{j+1}{(j+2)^2} \quad \text{for all integers} \quad j \geq 0$$

Scanned with CamScanner

Arithmetic Progression OR Sequences

- A sequence in which every term after the first is obtained from the preceding term by adding a constant number is called an arithmetic sequence or arithmetic progression (A.P.)
- The constant number, being the difference of any two consecutive terms is called the common difference of A.P., commonly denoted by “d”.

EXAMPLES:

1. 5, 9, 13, 17, … (common difference = 4)
2. 0, -5, -10, -15, … (common difference = -5)
3. x + a, x + 3a, x + 5a, … (common difference = 2a)

Scanned with CamScanner

Arithmetic Sequences

GENERAL TERM OF AN ARITHMETIC SEQUENCE:

Let **a** be the first term and **d** be the common difference of an arithmetic sequence. Then the sequence is:

$$a, a+d, a+2d, a+3d, \ldots$$

If a_i, for $i \geq 1$, represents the terms of the sequence then

a_1 = first term = $a = a + (1-1)\,d$

a_2 = second term = $a + d = a + (2-1)\,d$

a_3 = third term = $a + 2d = a + (3-1)\,d$

By symmetry

a_n = nth term = $a + (n-1)d$ for all integers $n \geq 1$.

Scanned with CamScanner

Arithmetic Sequences

Examples:

1. Let $a = -1$ and $d = 4$:

$$\{s_n\} = \{s_0, s_1, s_2, s_3, s_4, \ldots\} = \{-1, 3, 7, 11, 15, \ldots\}$$

2. Let $a = 7$ and $d = -3$:

$$\{t_n\} = \{t_0, t_1, t_2, t_3, t_4, \ldots\} = \{7, 4, 1, -2, -5, \ldots\}$$

3. Let $a = 1$ and $\mathrm{d} = 2$:

$$\{u_n\} = \{u_0, u_1, u_2, u_3, u_4, \ldots\} = \{1, 3, 5, 7, 9, \ldots\}$$

Scanned with CamScanner

Arithmetic Sequences

EXAMPLE:

Find the 20th term of the arithmetic sequence

3, 9, 15, 21, …

SOLUTION:

- Here a = first term = 3
- d = common difference = 9 - 3 = 6
- n = term number = 20
- a_{20} = value of 20th term = ?
- Since $a_n = a + (n - 1)\,d;\ n \geq 1$

$\therefore a_{20} = 3 + (20 - 1)\,6$

$= 3 + 114$

$= 117$

Scanned with CamScanner

Arithmetic Sequences

EXERCISE:

Find the 36th term of the arithmetic sequence whose 3rd term is 7 and 8th term is 17.

SOLUTION:

Let **a** be the first term and **d** be the common difference of the arithmetic sequence.

Then $a_n = a + (n - 1)d \quad n \geq 1$

$\Rightarrow a_3 = a + (3 - 1)\,d$ and $a_8 = a + (8 - 1)\,d$

Given that $a_3 = 7$ and $a_8 = 17$. Therefore

$7 = a + 2d$……(1) and $17 = a + 7d$……..(2)

Subtracting (1) from (2), we get,

$10 = 5d \qquad \Rightarrow d = 2$

Substituting $d = 2$ in (1) we have

$7 = a + 2(2)$ which gives $a = 3$

Scanned with CamScanner

Arithmetic Sequences

Thus, $a_n = a + (n - 1)\, d$

$a_n = 3 + (n - 1)\, 2$ (using values of a and d)

Hence the value of 36th term is

$$a_{36} = 3 + (36 - 1)\, 2$$

$$= 3 + 70$$

$$a_{36} = 73$$

Scanned with CamScanner

Geometric Progression OR Sequence

- A sequence in which every term after the first is obtained from the preceding term by multiplying it with a constant number is called a geometric sequence or geometric progression (G.P.)
- The constant number, being the ratio of any two consecutive terms is called the common ratio of the G.P. commonly denoted by “r”.
- **EXAMPLE:**

1. 1, 2, 4, 8, 16, … (common ratio = 2)
2. 3, - 3/2, 3/4, - 3/8, … (common ratio = - 1/2)
3. 0.1, 0.01, 0.001, 0.0001, … (common ratio = 0.1 = 1/10)

Scanned with CamScanner

GENERAL TERM OF A GEOMETRIC SEQUENCE:

Let **a** be the first tem and **r** be the common ratio of a geometric sequence. Then the sequence is $a, ar, ar^2, ar^3, \ldots$

If a_i, for $i \geq 1$ represent the terms of the sequence, then

a_1 = first term = $a = ar^{1-1}$

a_2 = second term = $ar = ar^{2-1}$

a_3 = third term = $ar^2 = ar^{3-1}$

.....

.....

a_n = nth term = ar^{n-1}; for all integers $n \geq 1$

Scanned with CamScanner

Geometric Progression

Examples:

1. Let $a = 1$ and $r = -1$. Then:

$$\{b_n\} = \{b_0, b_1, b_2, b_3, b_4, \ldots\} = \{1, -1, 1, -1, 1, \ldots\}$$

2. Let $a = 2$ and $r = 5$. Then:

$$\{c_n\} = \{c_0, c_1, c_2, c_3, c_4, \ldots\} = \{2, 10, 50, 250, 1250, \ldots\}$$

3. Let $a = 6$ and $r = 1/3$. Then:

$$\{d_n\} = \{d_0, d_1, d_2, d_3, d_4, \ldots\} = \{6, 2, \frac{2}{3}, \frac{2}{9}, \frac{2}{27}, \ldots\}$$

Scanned with CamScanner

Geometric Sequence

EXAMPLE:

Find the 8th term of the following geometric sequence

4, 12, 36, 108, …

SOLUTION:

Here a = first term = 4

r = common ratio = $\frac{12}{4} = 3$

n = term number = 8

a_8 = value of 8th term = ?

Since $a_n = ar^{n-1}$; $n \geq 1$

$\Rightarrow$ $a_8 = (4)(3)^{8-1}$

$= 4\,(2187)$

$= 8748$

Scanned with CamScanner

Geometric Sequence

EXERCISE:

Write the geometric sequence with positive terms whose second term is 9 and fourth term is 1.

SOLUTION:

Let **a** be the first term and **r** be the common ratio of the geometric sequence. Then

$$a_n = ar^{n-1} \qquad n \geq 1$$

Now $\quad a_2 = ar^{2-1}$

$\Rightarrow \quad 9 = ar \ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$

Also $\quad a_4 = ar^{4-1}$

$1 = ar^3 \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$

Dividing (2) by (1), we get,

$$\frac{1}{9} = \frac{ar^3}{ar}$$

$$\Rightarrow \quad \frac{1}{9} = r^2$$

$$\Rightarrow \quad r = \frac{1}{3} \qquad \left(\text{rejecting } r = -\frac{1}{3}\right)$$

Substituting r = 1/3 in (1), we get

$$9 = a\left(\frac{1}{3}\right)$$

$$\Rightarrow \quad a = 9 \times 3 = 27$$

Hence the geometric sequence is

27, 9, 3, 1, 1/3, 1/9, …

Scanned with CamScanner

Useful Sequences

TABLE 1 Some Useful Sequences.

nth Term	*First 10 Terms*
n^2	1, 4, 9, 16, 25, 36, 49, 64, 81, 100, . . .
n^3	1, 8, 27, 64, 125, 216, 343, 512, 729, 1000, . . .
n^4	1, 16, 81, 256, 625, 1296, 2401, 4096, 6561, 10000, . . .
2^n	2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, . . .
3^n	3, 9, 27, 81, 243, 729, 2187, 6561, 19683, 59049, . . .
$n!$	1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800, . . .
f_n	1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, . . .

Scanned with CamScanner

SERIES

The sum of the terms of a sequence forms a series. If $a_1, a_2, a_3, \ldots$ represent a sequence of numbers, then the corresponding series is

$$a_1 + a_2 + a_3 + \ldots = \sum_{k=1}^{\infty} a_k$$

Scanned with CamScanner

SUMMATIONS

SUMMATION NOTATION:

The capital Greek letter sigma $\sum$ is used to write a sum in a short hand notation.

where k varies from 1 to n represents the sum given in expanded form by

$= a_1 + a_2 + a_3 + \ldots + a_n$

More generally if m and n are integers and $m \leq n$, then the summation from k equal m to n of a_k is

$$\sum_{k=m}^{n} a_k = a_m + a_{m+1} + a_{m+2} + \cdots + a_n$$

Here **k** is called the index of the summation; **m** the lower limit of the summation and **n** the upper limit of the summation.

Scanned with CamScanner

SUMMATIONS

COMPUTING SUMMATIONS:

Let $a_0 = 2$, $a_1 = 3$, $a_2 = -2$, $a_3 = 1$ and $a_4 = 0$.Compute each of the summations:

$$(a) \quad \sum_{i=0}^{4} a_i \qquad (b) \quad \sum_{j=0}^{2} a_{2j} \qquad (c) \quad \sum_{k=1}^{1} a_k$$

SOLUTION:

(a) $$\sum_{i=0}^{4} a_i = a_0 + a_1 + a_2 + a_3 + a_4$$
$$= 2 + 3 + (-2) + 1 + 0 = 4$$

(b) $$\sum_{j=0}^{2} a_{2j} = a_0 + a_2 + a_4$$
$$= 2 + (-2) + 0 = 0$$

(c) $$\sum_{k=1}^{1} a_k = a_1$$
$$= 3$$

Scanned with CamScanner

ARITHMETIC SERIES:

The sum of the terms of an arithmetic sequence forms an arithmetic series (A.S). For example

$$1 + 3 + 5 + 7 + \ldots$$

is an arithmetic series of positive odd integers.

In general, if a is the first term and d the common difference of an arithmetic series, then the series is given as: $a + (a+d) + (a+2d) + \ldots$

Scanned with CamScanner

SUM OF n TERMS OF AN ARITHMETIC SERIES:

Let a be the first term and d be the common difference of an arithmetic series. Then its nth term is:

$$a_n = a + (n - 1)d; \quad n \geq 1$$

If S_n denotes the sum of first n terms of the A.S, then

$$S_n = a + (a + d) + (a + 2d) + \ldots + [a + (n-1) d]$$
$$= a + (a+d) + (a + 2d) + \ldots + a_n$$
$$= a + (a+d) + (a + 2d) + \ldots + (a_n - d) + a_n \ldots\ldots\ldots(1)$$

where $a_n = a + (n - 1) d$

Rewriting the terms in the series in reverse order,

$$Sn = a_n + (a_n - d) + (a_n - 2d) + \ldots + (a + d) + a \ldots\ldots\ldots(2)$$

Adding (1) and (2) term by term, gives

$$2 S_n = (a + a_n) + (a + a_n) + (a + a_n) + \ldots + (a + a_n) \qquad \text{(n terms)}$$
$$2 S_n = n (a + a_n)$$
$$\Rightarrow \quad S_n = n(a + a_n)/2$$
$$S_n = n(a + l)/2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

Where $l = an = a + (n - 1)d$

Therefore

$$S_n = n/2 [a + a + (n - 1) d]$$
$$S_n \; n/2 [2 a + (n - 1) d] \ldots\ldots\ldots(4)$$

Scanned with CamScanner

ARITHMETIC SERIES:

EXERCISE:

Find the sum of first n natural numbers.

SOLUTION:

Let $S_n = 1 + 2 + 3 + \ldots + n$

Clearly the right hand side forms an arithmetic series with

$a = 1, \quad d = 2 - 1 = 1 \qquad \text{and} \quad n = n$

$$\begin{aligned} \therefore \qquad S_n &= \frac{n}{2}[2a + (n-1)d] \\ &= \frac{n}{2}[2(1) + (n-1)(1)] \\ &= \frac{n}{2}[2 + n - 1] \\ &= \frac{n(n+1)}{2} \end{aligned}$$

Scanned with CamScanner

GEOMETRIC SERIES:

The sum of the terms of a geometric sequence forms a geometric series (G.S.). For example

$$1 + 2 + 4 + 8 + 16 + \ldots$$

is geometric series.

In general, if **a** is the first term and **r** the common ratio of a geometric series, then the series is given as: $a + ar + ar^2 + ar^3 + \ldots$

Scanned with CamScanner

Strings

Definition: A *string* is a finite sequence of characters from a finite set (an alphabet).

- Sequences of characters or bits are important in computer science.
- The *empty string* is represented by λ.
- The string *abcde* has *length* 5.

Scanned with CamScanner

Recurrence Relations

Definition: A *recurrence relation* for the sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms of the sequence, namely, $a_0, a_1, \ldots, a_{n-1}$, for all integers n with $n \geq n_0$, where n_0 is a nonnegative integer.

- A sequence is called a *solution* of a recurrence relation if its terms satisfy the recurrence relation.
- The *initial conditions* for a sequence specify the terms that precede the first term where the recurrence relation takes effect.

Scanned with CamScanner

Questions about Recurrence Relations

Example 1: Let $\{a_n\}$ be a sequence that satisfies the recurrence relation $a_n = a_{n-1} + 3$ for $n = 1,2,3,4,....$ and suppose that $a_0 = 2$. What are a_1, a_2 and a_3?

[Here $a_o = 2$ is the initial condition.]

Solution: We see from the recurrence relation that

$$a_1 = a_o + 3 = 2 + 3 = 5$$
$$a_2 = 5 + 3 = 8$$
$$a_3 = 8 + 3 = 11$$

Scanned with CamScanner

Questions about Recurrence Relations

Example 2: Let $\{a_n\}$ be a sequence that satisfies the recurrence relation $a_n = a_{n-1} - a_{n-2}$ for $n = 2,3,4,\ldots.$ and suppose that $a_0 = 3$ and $a_1 = 5$. What are a_2 and a_3?

[Here the initial conditions are $a_0 = 3$ and $a_1 = 5$.]

Solution: We see from the recurrence relation that

$$a_2 = a_1 - a_0 = 5 - 3 = 2$$

$$a_3 = a_2 - a_1 = 2 - 5 = -3$$

Scanned with CamScanner

Solving Recurrence Relations

- Finding a formula for the *n*th term of the sequence generated by a recurrence relation is called *solving the recurrence relation*.
- Such a formula is called a *closed formula*.
- Various methods for solving recurrence relations will be covered in Chapter 8 where recurrence relations will be studied in greater depth.
- Here we illustrate by example the method of iteration in which we need to guess the formula. The guess can be proved correct by the method of induction (Chapter 5).

Scanned with CamScanner

Fibonacci Sequence

Definition: Define the *Fibonacci sequence*, $f_0, f_1, f_2, \ldots,$ by:

- Initial Conditions: $f_0 = 0, f_1 = 1$
- Recurrence Relation: $f_n = f_{n-1} + f_{n-2}$

Example: Find f_2, f_3, f_4, f_5 and f_6 .

Answer:

$$f_2 = f_1 + f_0 = 1 + 0 = 1,$$
$$f_3 = f_2 + f_1 = 1 + 1 = 2,$$
$$f_4 = f_3 + f_2 = 2 + 1 = 3,$$
$$f_5 = f_4 + f_3 = 3 + 2 = 5,$$
$$f_6 = f_5 + f_4 = 5 + 3 = 8.$$

Scanned with CamScanner

Iterative Solution Example

Method 1: Working upward, forward substitution

Let $\{a_n\}$ be a sequence that satisfies the recurrence relation $a_n = a_{n-1} + 3$ for $n = 2,3,4,\ldots.$ and suppose that $a_1 = 2$.

$$a_2 = 2 + 3$$

$$a_3 = (2 + 3) + 3 = 2 + 3 \cdot 2$$

$$a_4 = (2 + 2 \cdot 3) + 3 = 2 + 3 \cdot 3$$

$$\vdots$$

$$a_n = a_{n-1} + 3 = (2 + 3 \cdot (n - 2)) + 3 = 2 + 3(n - 1)$$

Scanned with CamScanner

Iterative Solution Example

Method 2: Working downward, backward substitution

Let $\{a_n\}$ be a sequence that satisfies the recurrence relation $a_n = a_{n-1} + 3$ for $n = 2,3,4,\ldots.$ and suppose that $a_1 = 2$.

$$
\begin{aligned}
a_n &= a_{n-1} + 3 \\
&= (a_{n-2} + 3) + 3 = a_{n-2} + 3 \cdot 2 \\
&= (a_{n-3} + 3) + 3 \cdot 2 = a_{n-3} + 3 \cdot 3 \\
&\quad . \\
&\quad . \\
&\quad . \\
&= a_2 + 3(n-2) = (a_1 + 3) + 3(n-2) = 2 + 3(n-1)
\end{aligned}
$$

Scanned with CamScanner

Some Useful Summation Formulae

TABLE 2 Some Useful Summation Formulae.

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

Geometric Series: We just proved this.

Later we will prove some of these by induction.

Proof in text (requires calculus)

Scanned with CamScanner

Relations

Chapter 9

Mr. SHOAIB RAZA

Scanned with CamScanner

Chapter Summary

- Relations and Their Properties
- Representing Relations
- Equivalence Relations
- Partial Orderings

Scanned with CamScanner

Relations and Their Properties

Section 9.1

Scanned with CamScanner

Section Summary

- Relations and Functions
- Properties of Relations
  - Reflexive Relations
  - Symmetric Relations
  - Antisymmetric Relations
  - Transitive Relations
  - Irreflexive Relations
  - Asymmetric Relations
- Combining Relations

Scanned with CamScanner

Recall, Function is…

- Let **A** and **B** be nonempty sets Function ***f*** from **A** to **B** is an assignment of exactly one element of **B** to each element of **A.**
- By **defining** using a **relation**, a **function** from A to B contains **unique** ordered pair (a, b) for **every** element a ∈ A.

Scanned with CamScanner

What is Relation?

Scanned with CamScanner

Binary Relations

Definition: A *binary relation R* from a set *A* to a set *B* is a subset $R \subseteq A \times B$.

- Recall, for example:

  A = {a1, a2} and B = {b1, b2, b3}

  A x B = { (a1, b1), (a1, b2), (a1, b3),
  (a2, b1), (a2, b2), (a2, b3)}

- Ordered pairs, which
  - First element comes from A
  - Second element comes from B
  - **aRb**: (a, b) ∈ R
  - **a$\not$Rb** : (a, b) ∉ R

Moreover, when (a, b) belongs to R, a is said to be related to b by R.

Scanned with CamScanner

Binary Relations

Example:

- Let $A = \{0,1,2\}$ and $B = \{a,b\}$
- $\{(0, a), (0, b), (1,a), (2, b)\}$ is a relation from A to B.
- We can represent relations from a set A to a set B graphically or using a table:

R	a	b
0	×	×
1	×	
2		×

Scanned with CamScanner

Binary Relations

EXAMPLE:

- Let A = {eggs, milk, corn} and B = {cows, goats, hens}

Define a relation R from A to B by (a, b) ∈R *iff* a is produced by b.

- Then R = {(eggs, hens), (milk, cows), (milk, goats)}
- Thus, with respect to this relation eggs R hens , milk R cows, etc.

Scanned with CamScanner

Binary Relations

EXAMPLE #1:

- S = {Peter, Paul, Mary}
- C = {C++, DisMath}

- Given
  - Peter takes C++ — Peter R C++ — Peter $\not R$ DisMath
  - Paul takes DisMath — Paul $\not R$ C++ — Paul R DisMath
  - Mary takes none of them — Mary $\not R$ C++ — Mary $\not R$ DisMath

- R = {(Peter, C++), (Paul, DisMath)}

Scanned with CamScanner

Domain and Range of a Relation

DOMAIN OF A RELATION:

The domain of a relation R from A to B is the set of all first elements of the ordered pairs which belong to R denoted by Dom(R).

Symbolically, $\text{Dom}(R) = \{a \in A \mid (a, b) \in R\}$

RANGE OF A RELATION:

The range of a relation R from A to B is the set of all second elements of the ordered pairs which belong to R denoted Ran(R).

Symbolically, $\text{Ran}(R) = \{b \in B \mid (a, b) \in R\}$

Scanned with CamScanner

Domain and Range of a Relation

EXERCISE:

Let $A = \{1, 2\}$, $B = \{1, 2, 3\}$,

Define a binary relation R from A to B as follows:

$R = \{(a, b) \in A \times B \mid a < b\}$ Then

- a. Find the ordered pairs in R.
- b. Find the Domain and Range of R.
- c. Is 1R3, 2R2?

SOLUTION:

Given $A = \{1, 2\}$, $B = \{1, 2, 3\}$,

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

- a. $R = \{(a, b) \in A \times B \mid a < b\}$

  $R = \{(1,2), (1,3), (2,3)\}$

Scanned with CamScanner

Domain and Range of a Relation

Given $A = \{1, 2\}$, $B = \{1, 2, 3\}$,

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

- b. Find the Domain and Range of R.

Solution:

$Dom(R) = \{1,2\}$ and $Ran(R) = \{2, 3\}$

- c. Is $1R3$, $2R2$?

Solution:

c. Since $(1, 3) \in R$ so $1R3$.

Since $(2, 2) \in R$ so $2R2$.

Scanned with CamScanner

Representing Relations

Scanned with CamScanner

Representing Relations Using Digraphs

Definition: A *directed graph*, or *digraph*, consists of a set V of *vertices* (or *nodes*) together with a set E of ordered pairs of elements of V called *edges* (or *arcs*). The vertex a is called the *initial vertex* of the edge (a,b), and the vertex b is called the *terminal vertex* of this edge.

- An edge of the form (a,a) is called a *loop*.

Example: A drawing of the directed graph with vertices a, b, c, and d, and edges (a, b), (a, d), (b, b), (b, d), (c, a), (c, b), and (d, b) is shown here.

Scanned with CamScanner

Representing Relations Using Matrices

- A relation between finite sets can be represented using a zero-one matrix.
- Suppose R is a relation from $A = \{a_1, a_2, \ldots, a_m\}$ to $B = \{b_1, b_2, \ldots, b_n\}$.
  - The elements of the two sets can be listed in any particular arbitrary order. When $A = B$, we use the same ordering.
- The relation R is represented by the matrix $M_R = [m_{ij}]$, where

$$m_{ij} = \begin{cases} 1 \text{ if } (a_i, b_j) \in R, \\ 0 \text{ if } (a_i, b_j) \notin R. \end{cases}$$

- The matrix representing R has a 1 as its (i,j) entry when a_i is related to b_j and a 0 if a_i is not related to b_j.

Scanned with CamScanner

Examples of Representing Relations Using Matrices

Example 1: Suppose that $A = \{1,2,3\}$ and $B = \{1,2\}$. Let R be the relation from A to B containing (a,b) if $a \in A$, $b \in B$, and $a > b$. What is the matrix representing R (assuming the ordering of elements is the same as the increasing numerical order)?

Solution: Because $R = \{(2,1), (3,1),(3,2)\}$, the matrix is

$$M_R = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Scanned with CamScanner

Binary Relations

EXAMPLE #1: (cont.)

- Peter R C++, Peter $\not R$ DisMath
  Paul $\not R$ C++, Paul R DisMath
  Mary $\not R$ C++, Mary $\not R$ DisMath

Directed Graph

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Matrix

Scanned with CamScanner

Binary Relation on a Set

Definition: A binary relation *R on a set A* is a subset of $A \times A$ or a relation from A to A.

Example:

- 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

  $\{(1,1), (1, 2), (1,3), (1, 4), (2, 2), (2, 4), (3, 3), \text{and } (4, 4)\}$.

REMARK:

For any set A

1. $A \times A$ is known as the universal relation.
2. $\emptyset$ is known as the empty relation.

Scanned with CamScanner

Binary Relation on a Set

- Let A be the set {1, 2, 3, 4}, which ordered pairs are in the relation R = {(a, b) | a divides b}?

- **R** = { (1,1), (1,2), (1,3), (1,4), (2,2), (2,4), (3,3), (4,4)}

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Scanned with CamScanner

Relations and Their Properties

Scanned with CamScanner

Binary Relation on a Set (*cont.*)

Question: How many different relations are there on a set A with n elements?

Solution:

- Suppose A has n elements
- Recall, a relation on a set A is a subset of A x A.
- A x A has n^2elements.
- If a set has m element, its has 2^m subsets.
- Therefore, the answer is 2^{n^2}.

Scanned with CamScanner

Reflexive Relations

Definition: R is *reflexive* iff $(a,a) \in R$ for every element $a \in A$. Written symbolically, R is reflexive if and only if

$$\forall a[a \in U \longrightarrow (a,a) \in R]$$

Reflexive

$\forall a\ ((a, a) \in R)$

Every node has a self-loop

$$\begin{bmatrix} 1 & & & ? \\ & 1 & & \\ & & 1 & \\ ? & & & 1 \end{bmatrix}$$

Reflexive

$\forall a\ ((a, a) \in R)$

All 1's on diagonal

If $A = \emptyset$ then the empty relation is reflexive vacuously. That is the empty relation on an empty set is reflexive!

Scanned with CamScanner

Reflexive Relations

EXAMPLE: Let A = {1, 2, 3, 4} and determine whether relations R1, R2, R3, and R4 are Reflexive?

$R1 = \{(1, 1), (3, 3), (2, 2), (4, 4)\}$

$R2 = \{(1, 1), (1, 4), (2, 2), (3, 3), (4, 3)\}$

$R3 = \{(1, 1), (1, 2), (2, 1), (2, 2), (3, 3), (4, 4)\}$

$R4 = \{(1, 3), (2, 2), (2, 4), (3, 1), (4, 4)\}$

Solution:

- R1 is reflexive, since $(a, a) \in R1$ for all $a \in A$.
- R2 is not reflexive, because $(4, 4) \notin R2$.
- R3 is reflexive, since $(a, a) \in R3$ for all $a \in A$.
- R4 is not reflexive, because $(1, 1) \notin R4$, $(3, 3) \notin R4$.

Scanned with CamScanner

Symmetric Relations

Definition: R is *symmetric* iff $(b,a) \in R$ whenever $(a,b) \in R$ for all $a,b \in A$. Written symbolically, R is symmetric if and only if

$$\forall a \forall b\, [(a,b) \in R \longrightarrow (b,a) \in R]$$

Symmetric

$\forall a\ \forall b\ (\ ((a, b) \in R) \rightarrow ((b, a) \in R)\ )$

Every link is bidirectional

Symmetric

$\forall a\ \forall b\ (\ ((a, b) \in R) \rightarrow ((b, a) \in R)\ )$

All identical across diagonal

Accordingly, R is symmetric if the elements in the ith row are the same as the elements in the ith column of the matrix M representing R. More precisely, M is a symmetric matrix
i.e. $M = M^t$

Scanned with CamScanner

Symmetric Relations

EXAMPLE: Let A = {1, 2, 3, 4} and determine whether relations R1, R2, R3, and R4 are Symmetric?

$R1 = \{(1, 1), (1, 3), (2, 4), (3, 1), (4,2)\}$

$R2 = \{(1, 1), (2, 2), (3, 3), (4, 4)\}$

$R3 = \{(2, 2), (2, 3), (3, 4)\}$

$R4 = \{(1, 1), (2, 2), (3, 3), (4, 3), (4, 4)\}$

Solution:

- R1 is Symmetric, since (a, b) and (b, a) $\in$R1 for all (a, b) $\in$A.
- R2 is also symmetric. We say it is vacuously true.
- R3 is not symmetric, because (2,3) $\in$ R3 but (3,2) $\notin$ R3.
- R4 is not symmetric because (4,3) $\in$ R4 but (3,4) $\notin$ R4.

Scanned with CamScanner

Transitive Relations

Definition: 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 $a,b,c \in A$. Written symbolically, R is transitive if and only if

$$\forall a \forall b \forall c [(a,b) \in R \wedge (b,c) \in R \longrightarrow (a,c) \in R]$$

Transitive

$\forall a \forall b \forall c\ ((a,b) \in R \wedge (b,c) \in R) \rightarrow ((a,c) \in R))$

Every two adjacent forms a triangle
(Not easy to observe in Graph)

?

Transitive

$\forall a \forall b \forall c\ ((a,b) \in R \wedge (b,c) \in R) \rightarrow ((a,c) \in R))$

Not easy to observe in Matrix

For a transitive directed graph, whenever there is an arrow going from one point to the second, and from the second to the third, there is an arrow going directly from the first to the third.

Scanned with CamScanner

Transitive Relations

EXAMPLE: Let A = {1, 2, 3, 4} and determine whether relations R1, R2 and R3 are Transitive?

R1 = {(1, 1), (1, 2), (1, 3), (2, 3)}

R2 = {(1, 2), (1, 4), (2, 3), (3, 4)}

R3 = {(2, 1), (2, 4), (2, 3), (3,4)}

Solution:

- R1 is transitive because (1, 1), (1, 2) are in R, then to be transitive relation(1,2) must be there and it belongs to R.
- R2 is not transitive since (1,2) and (2,3) ∈ R2 but (1,3) ∉ R2.
- R3 is transitive.(check by definition)

Scanned with CamScanner

Irreflexive Relations

Definition: R is irreflexive iff for all a∈A,(a,a) ∉R. That is, R is irreflexive if no element in A is related to itself by R.

Written symbolically, R is irreflexive if and only if

$$\forall a\,[(a \in A) \longrightarrow (a\,,\,a) \notin R]$$

Irreflexive

$\forall a\ (\ (a \in A) \rightarrow ((a,\ a) \notin R)\ )$

No node links to itself

R is not irreflexive iff there is an element a∈A such that (a,a) ∈R.

$$\begin{bmatrix} 0 & & & ? \\ & 0 & & \\ & & 0 & \\ ? & & & 0 \end{bmatrix}$$

Irreflexive

$\forall a\ (\ (a \in A) \rightarrow ((a,\ a) \notin R)\ )$

All 0's on diagonal

Scanned with CamScanner

Irreflexive Relations

EXAMPLE: Let A = {1, 2, 3, 4} and determine whether relations R1, R2 and R3 are Irreflexive?

R1 = {(1,3), (1,4), (2,3), (2,4), (3,1), (3,4)}

R2 = {(1,1), (1,2), (2,1), (2,2), (3,3), (4,4)}

R3= {(1,2), (2,3), (3,3), (3,4)}

Solution:

- R1 is irreflexive since no element of A is related to itself in R1. i.e. (1,1)$\notin$ R1, (2,2) $\notin$ R1, (3,3) $\notin$ R1,(4,4) $\notin$ R1.
- R2 is not irreflexive, since all elements of A are related to themselves in R2.
- R3 is not irreflexive since (3,3) $\in$R3. Note that R3 is not reflexive.

Scanned with CamScanner

Antisymmetric Relations

Definition: 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*. Written symbolically, R is antisymmetric if and only if $\forall a \forall b\, [(a,b) \in R \wedge (b,a) \in R \longrightarrow a = b]$

Note: (a,a) may be an element in R.

Antisymmetric

$\forall a\ \forall b\ (\ ((a, b) \in R \wedge (b, a) \in R) \rightarrow (a = b)\ )$

No link is bidirectional

Antisymmetric

$\forall a\ \forall b\ (\ ((a, b) \in R \wedge (b, a) \in R) \rightarrow (a = b)\ )$

All 1's are across from 0's

Let R be an anti-symmetric relation on a set A = {a1, a2, ..., an}. Then if (ai, aj) ∈R for i ≠ j then (ai, aj) ∉R. Thus in the matrix representation of R there is a 1 in the ith row and jth column iff the jth row and ith column contains 0 vice versa.

Scanned with CamScanner

Antisymmetric Relations

EXAMPLE: Let A = {1, 2, 3, 4} and determine whether relations R1, R2, R3, and R4 are Antisymmetric?

R1 = {(1,1),(2,2),(3,3)}

R2 = {(1,2),(2,2), (2,3), (3,4), (4,1)}

R3={(1,3),(2,2), (2,4), (3,1), (4,2)}

R4={(1,3),(2,4), (3,1), (4,3)}

Solution:

- R1 is anti-symmetric and symmetric.
- R2 is anti-symmetric but not symmetric because (1,2) ∈ R2 but (2,1) ∉ R2.
- R3 is not anti-symmetric since (1,3) & (3,1) ∈ R3 but 1 ≠ 3. Note that R3 is symmetric.
- R4 is neither anti-symmetric because (1,3) & (3,1) ∈ R4 but 1 ≠ 3 nor symmetric because (2,4) ∈ R4 but (4,2) ∉R4.

Scanned with CamScanner

Asymmetric Relations

Definition: R is Asymmetric iff for all $(a,b) \in R$ *than* $(b,a) \notin R$.
Written symbolically, R is Asymmetric if and only if

$$\forall a \forall b[((a,b) \in R)) \longrightarrow ((b, a) \notin R)]$$

Note:(a,a) cannot be an element in R.

Asymmetric

$\forall a \forall b (((a, b) \in R) \rightarrow ((b, a) \notin R))$

All 1's are across from 0's (Antisymmetric)
All 0's on diagonal (Irreflexive)

Asymmetry = Antisymmetry + Irreflexivity

Asymmetric

$\forall a \forall b (((a, b) \in R) \rightarrow ((b, a) \notin R))$

No link is bidirectional (Antisymmetric)
No node links to itself (Irreflexive)

Scanned with CamScanner

Asymmetric Relations

- **EXAMPLE**: Let A = {1, 2, 3, 4} and determine whether relations R1, R2 and R3 are Asymmetric?

  R1 = {(1,1), (1,2), (2,1), (2,2), (3,4), (4,1), (4,4)}

  R2 = {(1,2),(2,3),(3,4)}

  R3 = {(2,3),(3,3),(3,4)}

Solution:

- R1 is not Asymmetric since R1 is neither Antisymmetric nor Irreflexive.
- R2 is Asymmetric since R2 is both Antisymmetric and Irreflexive.
- R3 is not Asymmetric since it is Antisymmetric but not irreflexive.

Scanned with CamScanner

Activity Time

Scanned with CamScanner

Consider the following relations on {1, 2, 3, 4}:

$R1 = \{(1, 1), (1, 2), (2, 1), (2, 2), (3, 4), (4, 1), (4, 4)\}$,
$R2 = \{(1, 1), (1, 2), (2, 1)\}$,
$R3 = \{(1, 1), (1, 2), (1, 4), (2, 1), (2, 2), (3, 3), (4, 1), (4, 4)\}$,
$R4 = \{(2, 1), (3, 1), (3, 2), (4, 1), (4, 2), (4, 3)\}$,
$R5 = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)\}$,
$R6 = \{(3, 4)\}$.

Determine which of these relation are Reflexive, Symmetric, Transitive, Antisymmetric, Irreflexive and Asymmetric.

Scanned with CamScanner

Combining Relations

As R is a subsets of A x B, the set operations can be applied

- Union (U)
- Intersection ($\cap$)
- Difference (-)
- Symmetric Complement ($\oplus$)

Given two relations R_1 and R_2, we can combine them using basic set operations to form new relations such as $R_1 \cup R_2$, $R_1 \cap R_2$, $R_1 - R_2$, $R_2 - R_1$ and $R_1 \oplus R_2$.

Scanned with CamScanner

Combining Relations

Given, $A = \{1,2,3\}$, $B = \{1,2,3,4\}$

$R_1 = \{(1,1),(2,2),(3,3)\}$,

$R_2 = \{(1,1),(1,2),(1,3),(1,4)\}$

- $R_1 \cup R_2 = \{(1,1),(1,2),(1,3),(1,4),(2,2),(3,3)\}$
- $R_1 \cap R_2 = \{(1,1)\}$
- $R_1 - R_2 = \{(2,2),(3,3)\}$
- $R_2 - R_1 = \{(1,2),(1,3),(1,4)\}$
- $R_1 \oplus R_2 = \{(1,2),(1,3),(1,4),(2,2),(3,3)\}$

Scanned with CamScanner

Composition of Relations

Definition: 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 $R_2 \circ R_1$.

Scanned with CamScanner

Representing the Composition of a Relation

$$R_1 \circ R_2 = \{(b,D),(b,B)\}$$

Scanned with CamScanner

Composition of Relations

What is the composite of the relations R and S, where

- R is the relation from **{1,2,3}** to **{1,2,3,4}** with

  R = {(1,1),(1,4),(2,3),(3,1),(3,4)}

- S is the relation from **{1,2,3,4}** to **{0,1,2}** with

  S = {(1,0),(1,2),(2,0),(3,1),(3,2),(4,1)}?

- S o R = {(1,0),(1,2),(1,1), (2,2),((2,1), 3,0),(3,2),(3,1)}

Scanned with CamScanner

INVERSE OF A RELATION

Let R be a relation from A to B. The inverse relation R^{-1} from B to A is defined as:

$$R^{-1} = \{(b,a) \in B \times A \mid (a,b) \in R\}$$

More simply, the inverse relation R^{-1} of R is obtained by interchanging the elements of all the ordered pairs in R.

- **Example**

X = {a, b, c} and Y={1, 2}

R = {(a, 1), (b, 2), (c, 1)}

- R^{-1} = {(1, a), (2, b), (1, c)}

Scanned with CamScanner

INVERSE OF A RELATION

The relation

R = {(2,2), (2,6), (2,8), (3,6), (4,8)} is represented by the arrow diagram.

Then inverse of the above relation can be obtained simply changing the directions of the arrows and hence the diagram is

Scanned with CamScanner

Equivalence Relations

Scanned with CamScanner

Equivalence Relations

Definition 1: A relation on a set A is called an *equivalence relation* if it is reflexive, symmetric, and transitive.

Definition 2: 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.

Scanned with CamScanner

Strings

Example:

Suppose that R is the relation on the set of strings of English letters such that aRb if and only if $l(a) = l(b)$, where $l(x)$ is the length of the string x. Is R an equivalence relation?

Solution: Show that all of the properties of an equivalence relation hold.

- *Reflexivity*: Because $l(a) = l(a)$, it follows that aRa for all strings a.
- *Symmetry*: Suppose that aRb. Since $l(a) = l(b)$, $l(b) = l(a)$ also holds and bRa.
- *Transitivity*: Suppose that aRb and bRc. Since $l(a) = l(b)$, and $l(b) = l(c)$, $l(a) = l(a)$ also holds and aRc.

Scanned with CamScanner

Congruence Modulo m

Example: Let m be an integer with $m > 1$. Show that the relation

$$R = \{(a,b) \mid a \equiv b \pmod{m}\}$$

is an equivalence relation on the set of integers.

Solution: Recall that $a \equiv b \pmod{m}$ if and only if m divides $a - b$.

- *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 $b - a = (-k)\, m$, 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:

  $$a - c = (a - b) + (b - c) = km + lm = (k + l)\, m.$$

  Therefore, $a \equiv c \pmod{m}$.

Scanned with CamScanner

Divides

Example: Show that the "divides" relation on the set of positive integers is not an equivalence relation.

Solution: The properties of reflexivity, and transitivity do hold, but there relation is not transitive. Hence, "divides" is not an equivalence relation.

- *Reflexivity*: $a \mid a$ for all a.
- *Not Symmetric*: For example, $2 \mid 4$, but $4 \nmid 2$. Hence, the relation is not symmetric.
- *Transitivity*: Suppose that a divides b and b divides c. Then there are positive integers k and l such that $b = ak$ and $c = bl$. Hence, $c = a(kl)$, so a divides c. Therefore, the relation is transitive.

Scanned with CamScanner

Partial Orderings

Scanned with CamScanner

Partial Orderings

Definition 1: A relation R on a set S is called a *partial ordering*, or *partial order*, if it is reflexive, antisymmetric, and transitive.

A set together with a partial ordering R is called a *partially ordered set*, or *poset*, and is denoted by (S, R). Members of S are called *elements* of the poset.

Scanned with CamScanner

Partial Orderings (*continued*)

Example 1: Show that the "greater than or equal" relation ($\geq$) is a partial ordering on the set of integers.

- *Reflexivity*: $a \geq a$ for every integer a.
- *Antisymmetry*: If $a \geq b$ and $b \geq a$, then $a = b$.
- *Transitivity*: If $a \geq b$ and $b \geq c$, then $a \geq c$.

These properties all follow from the order axioms for the integers. (*See Appendix* 1).

Scanned with CamScanner

Partial Orderings (*continued*)

Example 2: Show that the divisibility relation (|) is a partial ordering on the set of integers.

- *Reflexivity*: $a \mid a$ for all integers a. (*see Example* 9 *in Section* 9.1)
- *Antisymmetry*: If a and b are positive integers with $a \mid b$ and $b \mid a$, then $a = b$. (*see Example* 12 *in Section* 9.1)
- *Transitivity*: Suppose that a divides b and b divides c. Then there are positive integers k and l such that $b = ak$ and $c = bl$. Hence, $c = a(kl)$, so a divides c. Therefore, the relation is transitive.

- $(\boldsymbol{Z}^+, \mid)$ is a poset.

Scanned with CamScanner

Partial Orderings (*continued*)

Example 3: Show that the inclusion relation ($\subseteq$) is a partial ordering on the power set of a set S.

- *Reflexivity*: $A \subseteq A$ whenever A is a subset of S.
- *Antisymmetry*: If A and B are positive integers with $A \subseteq B$ and $B \subseteq A$, then $A = B$.
- *Transitivity*: If $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$.

The properties all follow from the definition of set inclusion.

Scanned with CamScanner

Number Theory and Cryptography

Chapter 4

Scanned with CamScanner

Chapter Motivation

- *Number theory* is the part of mathematics devoted to the study of the integers and their properties.
- Key ideas in number theory include divisibility and the primality of integers.
- Number theory has long been studied because of the beauty of its ideas, its accessibility, and its wealth of open questions.
- We'll use many ideas developed in Chapter 1 about proof methods and proof strategy in our exploration of number theory.
- Mathematicians have long considered number theory to be pure mathematics, but it has important applications to computer science and cryptography studied in Sections 4.5 and 4.6.

Scanned with CamScanner

Chapter Summary

- Divisibility and Modular Arithmetic
- Primes and Greatest Common Divisors
- Solving Congruencies
- Applications of Congruencies
- Cryptography

Scanned with CamScanner

Divisibility and Modular Arithmetic

Section 4.1

Scanned with CamScanner

Section Summary

- Division
- Division Algorithm
- Modular Arithmetic

Scanned with CamScanner

Division

Definition: If a and b are integers with $a \neq 0$, then a *divides* b if there exists an integer c such that $b = ac$.

- When a divides b we say that a is a *factor* or *divisor* of b and that b is a multiple of a.
- The notation $a \mid b$ denotes that a divides b.
- If $a \mid b$, then b/a is an integer.
- If a does not divide b, we write $a \nmid b$.

Example: Determine whether $3 \mid 7$ and whether $3 \mid 12$.

Scanned with CamScanner

Properties of Divisibility

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

i. If $a \mid b$ and $a \mid c$, then $a \mid (b + c)$;
ii. If $a \mid b$, then $a \mid bc$ for all integers c;
iii. If $a \mid b$ and $b \mid c$, then $a \mid c$.

Proof: (i) Suppose $a \mid b$ and $a \mid c$, then it follows that there are integers s and t with $b = as$ and $c = at$. Hence,

$$b + c = as + at = a(s + t). \quad \text{Hence, } a \mid (b + c)$$

(Exercises 3 and 4 ask for proofs of parts (ii) and (iii).) ◄

Corollary: If a, b, and c be integers, where $a \neq 0$, such that $a \mid b$ and $a \mid c$, then $a \mid mb + nc$ whenever m and n are integers.

Can you show how it follows easily from from (ii) and (i) of Theorem 1?

Scanned with CamScanner

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 a positive integer, then there are unique integers q and r, with $0 \le r < d$, such that $a = dq + r$ (*proved in Section* 5.2).

- d is called the *divisor*.
- a is called the *dividend*.
- q is called the *quotient*.
- r is called the *remainder*.

Definitions of Functions
div and **mod**

$q = a$ **div** d
$r = a$ **mod** d

Examples:

- What are the quotient and remainder when 101 is divided by 11?
  Solution: The quotient when 101 is divided by 11 is 9 = 101 **div** 11, and the remainder is 2 = 101 **mod** 11.
- What are the quotient and remainder when −11 is divided by 3?
  Solution: The quotient when −11 is divided by 3 is −4 = −11 **div** 3, and the remainder is 1 = −11 **mod** 3.

Scanned with CamScanner

Congruence Relation

Definition: 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

$$a \not\equiv b \pmod{m}$$

Example: Determine whether 17 is congruent to 5 modulo 6 and whether 24 and 14 are congruent modulo 6.

Solution:

- $17 \equiv 5 \pmod{6}$ because 6 divides $17 - 5 = 12$.
- $24 \not\equiv 14 \pmod{6}$ since 6 divides $24 - 14 = 10$ is not divisible by 6.

Scanned with CamScanner

More on Congruences

Theorem 4: 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 $a = b + km$.

Proof:

- If $a \equiv b \pmod{m}$, then (by the definition of congruence) $m \mid a - b$. Hence, there is an integer k such that $a - b = km$ and equivalently $a = b + km$.
- Conversely, if there is an integer k such that $a = b + km$, then $km = a - b$. Hence, $m \mid a - b$ and $a \equiv b \pmod{m}$. ◀

Scanned with CamScanner

The Relationship between (mod m) and **mod** m Notations

- The use of "mod" in $a \equiv b \pmod{m}$ and a **mod** $m = b$ are different.
  - $a \equiv b \pmod{m}$ is a relation on the set of integers.
  - In a **mod** $m = b$, the notation **mod** denotes a function.
- The relationship between these notations is made clear in this theorem.
- **Theorem 3**: Let a and b be integers, and let m be a positive integer. Then $a \equiv b \pmod{m}$ if and only if a **mod** $m = b$ **mod** m. (*Proof in the exercises*)

Scanned with CamScanner

Congruencies of Sums and Products

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

$$a + c \equiv b + d \pmod{m} \text{ and } ac \equiv bd \pmod{m}$$

Example: Because $7 \equiv 2 \pmod{5}$ and $11 \equiv 1 \pmod{5}$, it follows from Theorem 5 that

$$18 = 7 + 11 \equiv 2 + 1 = 3 \pmod{5}$$

$$77 = 7\ 11 \equiv 2 * 1 = 2 \pmod{5}$$

◀

Scanned with CamScanner

Algebraic Manipulation of Congruencies

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

  If $a \equiv b \pmod{m}$ holds then $c \cdot a \equiv c \cdot b \pmod{m}$, where c is any integer, holds by Theorem 5 with $d = c$.

- Adding an integer to both sides of a valid congruence preserves validity.

  If $a \equiv b \pmod{m}$ holds then $c + a \equiv c + b \pmod{m}$, where c is any integer, holds by Theorem 5 with $d = c$.

- Dividing a congruence by an integer does not always produce a valid congruence.

  Example: The congruence $14 \equiv 8 \pmod{6}$ holds. But dividing both sides by 2 does not produce a valid congruence since $14/2 = 7$ and $8/2 = 4$, but $7 \not\equiv 4 \pmod{6}$.

Scanned with CamScanner

Computing the **mod** *m* Function of Products and Sums

- We use the following corollary to Theorem 5 to compute the remainder of the product or sum of two integers when divided by m from the remainders when each is divided by m.

Corollary: Let m be a positive integer and let a and b be integers. Then

$$(a + b)\ (\mathbf{mod}\ m) = ((a\ \mathbf{mod}\ m) + (b\ \mathbf{mod}\ m))\ \mathbf{mod}\ m$$

and

$$ab\ \mathbf{mod}\ m = ((a\ \mathbf{mod}\ m)\ (b\ \mathbf{mod}\ m))\ \mathbf{mod}\ m.$$

(*proof in text*)

Scanned with CamScanner

Applications of Congruences

Scanned with CamScanner

Section Summary

- Hashing Functions
- Pseudorandom Numbers
- Check Digits

Scanned with CamScanner

Hashing Functions

Definition: A *hashing function h* assigns memory location $h(k)$ to the record that has k as its key.

- A common hashing function is $h(k) = k$ **mod** m, where m is the number of memory locations.
- Because this hashing function is onto, all memory locations are possible.

Example: Let $h(k) = k$ **mod** 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 **mod** 111 = 14

h(037149212) = 037149212 **mod** 111 = 65

h(107405723) = 107405723 **mod** 111 = 14, but since location 14 is already occupied, the record is assigned to the next available position, which is 15.

- The hashing function is not one-to-one as there are many more possible keys than memory locations. When more than one record is assigned to the same location, we say a *collision* occurs. Here a collision has been resolved by assigning the record to the first free location.
- For collision resolution, we can use a *linear probing function*:

$$h(k,i) = (h(k) + i) \textbf{ mod } m, \text{ where } i \text{ runs from } 0 \text{ to } m - 1.$$

- There are many other methods of handling with collisions. You may cover these in a later CS course.

Scanned with CamScanner

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.
- The *linear congruential method* is one commonly used procedure for generating pseudorandom numbers.
- Four integers are needed: the *modulus* m, the *multiplier* a, the *increment* c, and *seed* x_0, with $2 \leq a < m$, $0 \leq c < m$, $0 \leq x_0 < m$.
- 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

$$x_{n+1} = (ax_n + c) \textbf{ mod } m.$$

(*an example of a recursive definition, discussed in Section* 5.3)

- If psudorandom numbers between 0 and 1 are needed, then the generated numbers are divided by the modulus, x_n / m.

Scanned with CamScanner

Pseudorandom Numbers

- **Example**: Find the sequence of pseudorandom numbers generated by the linear congruential method with modulus $m = 9$, multiplier $a = 7$, increment $c = 4$, and seed $x_0 = 3$.
- **Solution**: Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4)$ **mod** 9, with $x_0 = 3$.

$$x_1 = 7x_0 + 4 \textbf{ mod } 9 = 7\cdot 3 + 4 \textbf{ mod } 9 = 25 \textbf{ mod } 9 = 7,$$
$$x_2 = 7x_1 + 4 \textbf{ mod } 9 = 7\cdot 7 + 4 \textbf{ mod } 9 = 53 \textbf{ mod } 9 = 8,$$
$$x_3 = 7x_2 + 4 \textbf{ mod } 9 = 7\cdot 8 + 4 \textbf{ mod } 9 = 60 \textbf{ mod } 9 = 6,$$
$$x_4 = 7x_3 + 4 \textbf{ mod } 9 = 7\cdot 6 + 4 \textbf{ mod } 9 = 46 \textbf{ mod } 9 = 1,$$
$$x_5 = 7x_4 + 4 \textbf{ mod } 9 = 7\cdot 1 + 4 \textbf{ mod } 9 = 11 \textbf{ mod } 9 = 2,$$
$$x_6 = 7x_5 + 4 \textbf{ mod } 9 = 7\cdot 2 + 4 \textbf{ mod } 9 = 18 \textbf{ mod } 9 = 0,$$
$$x_7 = 7x_6 + 4 \textbf{ mod } 9 = 7\cdot 0 + 4 \textbf{ mod } 9 = 4 \textbf{ mod } 9 = 4,$$
$$x_8 = 7x_7 + 4 \textbf{ mod } 9 = 7\cdot 4 + 4 \textbf{ mod } 9 = 32 \textbf{ mod } 9 = 5,$$
$$x_9 = 7x_8 + 4 \textbf{ mod } 9 = 7\cdot 5 + 4 \textbf{ mod } 9 = 39 \textbf{ mod } 9 = 3.$$

The sequence generated is 3,7,8,6,1,2,0,4,5,3,7,8,6,1,2,0,4,5,3,...

It repeats after generating 9 terms.

- Commonly, computers use a linear congruential generator with increment $c = 0$. This is called a *pure multiplicative generator*. Such a generator with modulus $2^{31} - 1$ and multiplier $7^5 = 16{,}807$ generates $2^{31} - 2$ numbers before repeating.

Scanned with CamScanner

Check Digits: UPCs

- A common method of detecting errors in strings of digits is to add an extra digit 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.

Example: Retail products are identified by their *Universal Product Codes* (*UPC*s). Usually these have 12 decimal digits, the last one being the check digit. The check digit is determined by the congruence:

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

a. Suppose that the first 11 digits of the UPC are 79357343104. What is the check digit?

b. Is 041331021641 a valid UPC?

Solution:

a. $3\cdot 7 + 9 + 3\cdot 3 + 5 + 3\cdot 7 + 3 + 3\cdot 4 + 3 + 3\cdot 1 + 0 + 3\cdot 4 + x_{12} \equiv 0 \pmod{10}$

$21 + 9 + 9 + 5 + 21 + 3 + 12 + 3 + 3 + 0 + 12 + x_{12} \equiv 0 \pmod{10}$

$98 + x_{12} \equiv 0 \pmod{10}$

$x_{12} \equiv 0 \pmod{10}$ So, the check digit is 2.

b. $3\cdot 0 + 4 + 3\cdot 1 + 3 + 3\cdot 3 + 1 + 3\cdot 0 + 2 + 3\cdot 1 + 6 + 3\cdot 4 + 1 \equiv 0 \pmod{10}$

$0 + 4 + 3 + 3 + 9 + 1 + 0 + 2 + 3 + 6 + 12 + 1 = 44 \equiv 4 \not\equiv \pmod{10}$

Hence, 041331021641 is not a valid UPC.

Scanned with CamScanner

Check Digits:ISBNs

Books are identified by an *International Standard Book Number* (ISBN-10), a 10 digit code. The first 9 digits identify the language, the publisher, and the book. The tenth digit is a check digit, which is determined by the following congruence

$$x_{10} \equiv \sum_{i=1}^{9} i x_i \pmod{11}.$$

The validity of an ISBN-10 number can be evaluated with the equivalent $\sum_{i=1}^{10} i x_i \equiv 0 \pmod{11}$.

a. Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?

b. Is 084930149X a valid ISBN10?

Solution:

a. $X_{10} \equiv 1\cdot 0 + 2\cdot 0 + 3\cdot 7 + 4\cdot 2 + 5\cdot 8 + 6\cdot 8 + 7\cdot 0 + 8\cdot 0 + 9\cdot 8 \pmod{11}$.

$X_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72 \pmod{11}$.

$X_{10} \equiv 189 \equiv 2 \pmod{11}$. Hence, $X_{10} = 2$.

X is used for the digit 10.

b. $1\cdot 0 + 2\cdot 8 + 3\cdot 4 + 4\cdot 9 + 5\cdot 3 + 6\cdot 0 + 7\cdot 1 + 8\cdot 4 + 9\cdot 9 + 10\cdot 10 =$

$0 + 16 + 12 + 36 + 15 + 0 + 7 + 32 + 81 + 100 = 299 \equiv 2 \not\equiv 0 \pmod{11}$

Hence, 084930149X is not a valid ISBN-10.

- A *single error* is an error in one digit of an identification number and a *transposition error* is the accidental interchanging of two digits. Both of these kinds of errors can be detected by the check digit for ISBN-10. (*see text for more details*)

Scanned with CamScanner

Arithmetic Modulo *m*

Definitions: Let $\mathbf{Z}_m$ be the set of nonnegative integers less than m: $\{0,1, \ldots, m-1\}$

- The operation $+_m$ is defined as $a +_m b = (a + b)$ **mod** m. This is *addition modulo m*.
- The operation $\cdot_m$ is defined as $a \cdot_m b = (a \,.\, b)$ **mod** m. This is *multiplication modulo m*.
- Using these operations is said to be doing *arithmetic modulo m*.

Example: Find $7 +_{11} 9$ and $7 \cdot_{11} 9$.

Solution: Using the definitions above:

- $7 +_{11} 9 = (7 + 9)$ **mod** $11 = 16$ **mod** $11 = 5$
- $7 \cdot_{11} 9 = (7 \cdot 9)$ **mod** $11 = 63$ **mod** $11 = 8$

Scanned with CamScanner

Arithmetic Modulo *m*

- The operations $+_m$ and $\cdot_m$ satisfy many of the same properties as ordinary addition and multiplication.
  - *Closure*: If a and b belong to $\mathbf{Z}_m$, then $a +_m b$ and $a \cdot_m b$ belong to $\mathbf{Z}_m$.
  - *Associativity*: If a, b, and c belong to $\mathbf{Z}_m$, then $(a +_m b) +_m c = a +_m (b +_m c)$ and $(a \cdot_m b) \cdot_m c = a \cdot_m (b \cdot_m c)$.
  - *Commutativity*: If a and b belong to $\mathbf{Z}_m$, then $a +_m b = b +_m a$ and $a \cdot_m b = b \cdot_m a$.
  - *Identity elements*: The elements 0 and 1 are identity elements for addition and multiplication modulo m, respectively.
    - If a belongs to $\mathbf{Z}_m$, then $a +_m 0 = a$ and $a \cdot_m 1 = a$.

continued →

Scanned with CamScanner

Arithmetic Modulo *m*

- *Additive inverses*: If $a \neq 0$ belongs to $\mathbf{Z}_m$, then $m - a$ is the additive inverse of a modulo m and o is its own additive inverse.
  - $a +_m (m - a) = 0$ and $0 +_m 0 = 0$
- *Distributivity*: If *a*, *b*, and *c* belong to $\mathbf{Z}_m$, then
  - $a \cdot_m (b +_m c) = (a \cdot_m b) +_m (a \cdot_m c)$ and $(a +_m b) \cdot_m c = (a \cdot_m c) +_m (b \cdot_m c)$.
- Exercises 42-44 ask for proofs of these properties.
- Multiplicatative inverses have not been included since they do not always exist. For example, there is no multiplicative inverse of 2 modulo 6.
- (*optional*) Using the terminology of abstract algebra, $\mathbf{Z}_m$ with $+_m$ is a commutative group and $\mathbf{Z}_m$ with $+_m$ and $\cdot_m$ is a commutative ring.

Scanned with CamScanner

Primes and Greatest Common Divisors

Section 4.3

Scanned with CamScanner

Section Summary

- Prime Numbers and their Properties
- Greatest Common Divisors and Least Common Multiples
- The Euclidian Algorithm
- gcds as Linear Combinations

Scanned with CamScanner

Primes

Definition: A positive integer p greater than 1 is called *prime* if the only positive factors of p are 1 and p. A positive integer that is greater than 1 and is not prime is called *composite*.

Example: The integer 7 is prime because its only positive factors are 1 and 7, but 9 is composite because it is divisible by 3.

Scanned with CamScanner

The Fundamental Theorem of Arithmetic

Theorem: Every positive integer greater than 1 can be written uniquely as a prime or as the product of two or more primes where the prime factors are written in order of non decreasing size.

Examples:

- $100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 \cdot 5^2$
- $641 = 641$
- $999 = 3 \cdot 3 \cdot 3 \cdot 37 = 3^3 \cdot 37$
- $1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}$

Scanned with CamScanner

The Sieve of Erastosthenes

TABLE 1 **The Sieve of Eratosthenes.**

Integers divisible by 2 other than 2 receive an underline.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Integers divisible by 3 other than 3 receive an underline.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Integers divisible by 5 other than 5 receive an underline.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Integers divisible by 7 other than 7 receive an underline; integers in color are prime.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

If an integer n is a composite integer, then it has a prime divisor less than or equal to $\sqrt{n}$.

To see this, note that if $n = ab$, then $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$.

Trial division, a very inefficient method of determining if a number n is prime, is to try every integer $i \leq \sqrt{n}$ and see if n is divisible by i.

Scanned with CamScanner

Erastothenes
(276-194 B.C.)

The Sieve of Erastosthenes

- The *Sieve of Erastosthenes* can be used to find all primes not exceeding a specified positive integer. For example, begin with the list of integers between 1 and 100.
  a. Delete all the integers, other than 2, divisible by 2.
  b. Delete all the integers, other than 3, divisible by 3.
  c. Next, delete all the integers, other than 5, divisible by 5.
  d. Next, delete all the integers, other than 7, divisible by 7.
  e. Since all the remaining integers are not divisible by any of the previous integers, other than 1, the primes are:

{2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89, 97}

continued →

Scanned with CamScanner

Marin Mersenne
(1588-1648)

Mersenne Primes

Definition: Prime numbers of the form $2^p - 1$, where p is prime, are called *Mersenne primes*.

- $2^2 - 1 = 3$, $2^3 - 1 = 7$, $2^5 - 1 = 37$, and $2^7 - 1 = 127$ are Mersenne primes.
- $2^{11} - 1 = 2047$ is not a Mersenne prime since $2047 = 23 \cdot 89$.
- There is an efficient test for determining if $2^p - 1$ is prime.
- The largest known prime numbers are Mersenne primes.
- As of mid 2011, 47 Mersenne primes were known, the largest is $2^{43,112,609} - 1$, which has nearly 13 million decimal digits.
- The *Great Internet Mersenne Prime Search* (*GIMPS*) is a distributed computing project to search for new Mersenne Primes.

http://www.mersenne.org/

Scanned with CamScanner

Greatest Common Divisor

Definition: Let a and b be integers, not both zero. The largest integer d such that $d \mid a$ and also $d \mid b$ is called the greatest common divisor of a and b. The greatest common divisor of a and b is denoted by $\gcd(a,b)$.

One can find greatest common divisors of small numbers by inspection.

Example: What is the greatest common divisor of 24 and 36?

Solution: $\gcd(24,36) = 12$

Example: What is the greatest common divisor of 17 and 22?

Solution: $\gcd(17,22) = 1$

Scanned with CamScanner

Greatest Common Divisor

Definition: The integers a and b are *relatively prime* if their greatest common divisor is 1.

Example: 17 and 22

Definition: The integers $a_1, a_2, \ldots, a_n$ are *pairwise relatively prime* if $\gcd(a_i, a_j) = 1$ whenever $1 \leq i < j \leq n$.

Example: Determine whether the integers 10, 17 and 21 are pairwise relatively prime.

Solution: Because $\gcd(10,17) = 1$, $\gcd(10,21) = 1$, and $\gcd(17,21) = 1$, 10, 17, and 21 are pairwise relatively prime.

Example: Determine whether the integers 10, 19, and 24 are pairwise relatively prime.

Solution: Because $\gcd(10,24) = 2$, 10, 19, and 24 are not pairwise relatively prime.

Scanned with CamScanner

Finding the Greatest Common Divisor Using Prime Factorizations

- Suppose the prime factorizations of *a* and *b* are:

$$a = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n} \ , \qquad b = p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n} \ ,$$

where each exponent is a nonnegative integer, and where all primes occurring in either prime factorization are included in both. Then:

$$\gcd(a, b) = p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \cdots p_n^{\min(a_n, b_n)} \ .$$

- This formula is valid since the integer on the right (of the equals sign) divides both *a* and *b*. No larger integer can divide both *a* and *b*.

Example: $120 = 2^3 \cdot 3 \cdot 5 \qquad 500 = 2^2 \cdot 5^3$

$\gcd(120,500) = 2^{\min(3,2)} \cdot 3^{\min(1,0)} \cdot 5^{\min(1,3)} = 2^2 \cdot 3^0 \cdot 5^1 = 20$

- Finding the gcd of two positive integers using their prime factorizations is not efficient because there is no efficient algorithm for finding the prime factorization of a positive integer.

Scanned with CamScanner

Least Common Multiple

Definition: The least common multiple of the positive integers a and b is the smallest positive integer that is divisible by both a and b. It is denoted by $\text{lcm}(a,b)$.

- The least common multiple can also be computed from the prime factorizations.

$$\text{lcm}(a,b) = p_1^{\max(a_1,b_1)} p_2^{\max(a_2,b_2)} \cdots p_n^{\max(a_n,b_n)}$$

This number is divided by both a and b and no smaller number is divided by a and b.

Example: $\text{lcm}(2^3 3^5 7^2, 2^4 3^3) = 2^{\max(3,4)}\, 3^{\max(5,3)}\, 7^{\max(2,0)} = 2^4\, 3^5\, 7^2$

- The greatest common divisor and the least common multiple of two integers are related by:

Theorem 5: Let a and b be positive integers. Then

$$ab = \gcd(a,b) \cdot \text{lcm}(a,b)$$

(*proof is Exercise* 31)

Scanned with CamScanner

Euclidean Algorithm

Euclid
(325 B.C.E. – 265 B.C.E.)

- The Euclidian algorithm is an efficient method for computing the greatest common divisor of two integers. It is based on the idea that gcd(a,b) is equal to gcd(a,c) when $a > b$ and c is the remainder when a is divided by b.

Example: Find gcd(91, 287):

- $287 = 91 \cdot 3 + 14$ — Divide 287 by 91
- $91 = 14 \cdot 6 + 7$ — Divide 91 by 14
- $14 = 7 \cdot 2 + 0$ — Divide 14 by 7

Stopping condition

$$\gcd(287, 91) = \gcd(91, 14) = \gcd(14, 7) = 7$$

continued →

Scanned with CamScanner

Euclidean Algorithm

- The Euclidean algorithm expressed in pseudocode is:

```
procedure gcd(a, b: positive integers)
x := a
x := b
while  y ≠ 0
      r := x mod y
      x := y
      y := r
return x {gcd(a,b) is x}
```

- In Section 5.3, we'll see that the time complexity of the algorithm is $O(\log b)$, where $a > b$.

Scanned with CamScanner

Étienne Bézout
(1730-1783)

gcds as Linear Combinations

Bézout's Theorem: If a and b are positive integers, then there exist integers s and t such that $\gcd(a,b) = sa + tb$.

Definition: If a and b are positive integers, then integers s and t such that $\gcd(a,b) = sa + tb$ are called *Bézout coefficients* of a and b. The equation $\gcd(a,b) = sa + tb$ is called *Bézout's identity*.

- By Bézout's Theorem, the gcd of integers a and b can be expressed in the form $sa + tb$ where s and t are integers. This is a *linear combination* with integer coefficients of a and b.
  - $\gcd(6,14) = (-2)\cdot 6 + 1\cdot 14$

Scanned with CamScanner

Finding gcds as Linear Combinations

Example: Express gcd(252,198) = 18 as a linear combination of 252 and 198.

Solution: First use the Euclidean algorithm to show gcd(252,198) = 18

i. $252 = 1 \cdot 198 + 54$
ii. $198 = 3 \cdot 54 + 36$
iii. $54 = 1 \cdot 36 + 18$
iv. $36 = 2 \cdot 18$

- Now working backwards, from iii and i above
  - $18 = 54 - 1 \cdot 36$
  - $36 = 198 - 3 \cdot 54$
- Substituting the 2nd equation into the 1st yields:
  - $18 = 54 - 1 \cdot (198 - 3 \cdot 54) = 4 \cdot 54 - 1 \cdot 198$
- Substituting $54 = 252 - 1 \cdot 198$ (from i)) yields:
  - $18 = 4 \cdot (252 - 1 \cdot 198) - 1 \cdot 198 = 4 \cdot 252 - 5 \cdot 198$

- This method illustrated above is a two pass method. It first uses the Euclidian algorithm to find the gcd and then works backwards to express the gcd as a linear combination of the original two integers. A one pass method, called the *extended Euclidean algorithm*, is developed in the exercises.

Scanned with CamScanner

Dividing Congruencies by an Integer

- Dividing both sides of a valid congruence by an integer does not always produce a valid congruence (see Section 4.1).
- But dividing by an integer relatively prime to the modulus does produce a valid congruence:

Theorem 7: Let m be a positive integer and let a, b, and c be integers. If $ac \equiv bc \pmod{m}$ and $\gcd(c,m) = 1$, then $a \equiv b \pmod{m}$.

Proof: Since $ac \equiv bc \pmod{m}$, $m \mid ac - bc = c(a - b)$ by Lemma 2 and the fact that $\gcd(c,m) = 1$, it follows that $m \mid a - b$. Hence, $a \equiv b \pmod{m}$. ◀

Scanned with CamScanner

Solving Congruencies

Section 4.4

Scanned with CamScanner

Section Summary

- Linear Congruencies
- The Chinese Remainder Theorem
- Fermat's Little Theorem
- Pseudo primes

Scanned with CamScanner

Linear Congruencies

Definition: A congruence of the form

$$ax \equiv b(\bmod m),$$

where m is a positive integer, a and b are integers, and x is a variable, is called a *linear congruence*.

- The solutions to a linear congruence $ax \equiv b(\bmod m)$ are all integers x that satisfy the congruence.

Definition: An integer $\bar{a}$ such that $\bar{a}a \equiv 1(\bmod m)$ is said to be an *inverse* of a modulo m.

Example: 5 is an inverse of 3 modulo 7 since $5 \cdot 3 = 15 \equiv 1 (\bmod 7)$

- One method of solving linear congruencies makes use of an inverse $\bar{a}$, if it exists. Although we can not divide both sides of the congruence by a, we can multiply by $\bar{a}$ to solve for x.

Scanned with CamScanner

Inverse of a modulo m

- The following theorem guarantees that an inverse of a modulo m exists whenever a and m are relatively prime. Two integers a and b are relatively prime when $\gcd(a,b) = 1$.

Theorem 1: If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists. Furthermore, this inverse is unique modulo m. (This means that there is a unique positive integer $\bar{a}$ less than m that is an inverse of a modulo m and every other inverse of a modulo m is congruent to $\bar{a}$ modulo m.)

Proof: Since $\gcd(a,m) = 1$, by Theorem 6 of Section 4.3, there are integers s and t such that $sa + tm = 1$.

- Hence, $sa + tm \equiv 1 \pmod{m}$.
- Since $tm \equiv 0 \pmod{m}$, it follows that $sa \equiv 1 \pmod{m}$
- Consequently, s is an inverse of a modulo m.
- The uniqueness of the inverse is Exercise 7. ◀

Scanned with CamScanner

Finding Inverses

- The Euclidean algorithm and Bézout coefficients gives us a systematic approaches to finding inverses.

  Example: Find an inverse of 3 modulo 7.

  Solution: Because gcd(3,7) = 1, by Theorem 1, an inverse of 3 modulo 7 exists.

  - Using the Euclidian algorithm: $7 = 2 \cdot 3 + 1$.
  - From this equation, we get $-2 \cdot 3 + 1 \cdot 7 = 1$, and see that -2 and 1 are Bézout coefficients of 3 and 7.
  - Hence, -2 is an inverse of 3 modulo 7.
  - Also every integer congruent to -2 modulo 7 is an inverse of 3 modulo 7, i.e., 5, -9, 12, etc.

Scanned with CamScanner

Finding Inverses

Example: Find an inverse of 101 modulo 4620.

Solution: First use the Euclidian algorithm to show that gcd(101,4620) = 1.

$4620 = 45 \cdot 101 + 75$
$101 = 1 \cdot 75 + 26$
$75 = 2 \cdot 26 + 23$
$26 = 1 \cdot 23 + 3$
$23 = 7 \cdot 3 + 2$
$3 = 1 \cdot 2 + 1$
$2 = 2 \cdot 1$

Since the last nonzero remainder is 1, gcd(101,4260) = 1

Working Backwards:

$1 = 3 - 1 \cdot 2$
$1 = 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3$
$1 = -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23$
$1 = 8 \cdot 26 - 9 \cdot (75 - 2 \cdot 26) = 26 \cdot 26 - 9 \cdot 75$
$1 = 26 \cdot (101 - 1 \cdot 75) - 9 \cdot 75$
$= 26 \cdot 101 - 35 \cdot 75$
$1 = 26 \cdot 101 - 35 \cdot (42620 - 45 \cdot 101)$
$= -35 \cdot 42620 + 1601 \cdot 101$

Bézout coefficients : -35 and 1601

1601 is an inverse of 101 modulo 42620

Scanned with CamScanner

Using Inverses to Solve Congruences

- We can solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides by $\bar{a}$.

Example: What are the solutions of the congruence $3x \equiv 4 \pmod{7}$.

Solution: We found that -2 is an inverse of 3 modulo 7 (two slides back). We multiply both sides of the congruence by -2 giving

$$-2 \cdot 3x \equiv -2 \cdot 4 \pmod{7}.$$

Because $-6 \equiv 1 \pmod{7}$ and $-8 \equiv 6 \pmod{7}$, it follows that if x is a solution, then $x \equiv -8 \equiv 6 \pmod{7}$

We need to determine if every x with $x \equiv 6 \pmod{7}$ is a solution. Assume that $x \equiv 6 \pmod{7}$. By Theorem 5 of Section 4.1, it follows that $3x \equiv 3 \cdot 6 = 18 \equiv 4 \pmod{7}$ which shows that all such x satisfy the congruence.

The solutions are the integers x such that $x \equiv 6 \pmod{7}$, namely, $6, 13, 20 \ldots$ and $-1, -8, -15, \ldots$

Scanned with CamScanner

The Chinese Remainder Theorem

Theorem 2: (*The Chinese Remainder Theorem*) Let $m_1, m_2, \ldots, m_n$ be pairwise relatively prime positive integers greater than one and $a_1, a_2, \ldots, a_n$ arbitrary integers. Then the system

$$x \equiv a_1 \pmod{m_1}$$
$$x \equiv a_2 \pmod{m_2}$$
$$\vdots$$
$$x \equiv a_n \pmod{m_n}$$

has a unique solution modulo $m = m_1 m_2 \cdots m_n$.

(That is, there is a solution x with $0 \leq x < m$ and all other solutions are congruent modulo m to this solution.)

- **Proof**: We'll show that a solution exists by describing a way to construct the solution. Showing that the solution is unique modulo m is Exercise 30.

continued →

Scanned with CamScanner

The Chinese Remainder Theorem

To construct a solution first let $M_k = m/m_k$ for $k = 1,2,\ldots,n$ and $m = m_1 m_2 \cdots m_n$.

Since $\gcd(m_k, M_k) = 1$, by Theorem 1, there is an integer y_k, an inverse of M_k modulo m_k, such that

$$M_k y_k \equiv 1 \pmod{m_k}.$$

Form the sum

$$x = a_1 M_1 y_1 + a_2 M_2 y_2 + \cdots + a_n M_n y_n \pmod{m}$$

Note that because $M_j \equiv 0 \pmod{m_k}$ whenever $j \neq k$, all terms except the kth term in this sum are congruent to 0 modulo m_k.

Because $M_k y_k \equiv 1 \pmod{m_k}$, we see that $x \equiv a_k M_k y_k \equiv a_k \pmod{m_k}$, for $k = 1,2,\ldots,n$.

Hence, x is a simultaneous solution to the n congruences.

$$x \equiv a_1 \pmod{m_1}$$
$$x \equiv a_2 \pmod{m_2}$$
$$\vdots$$
$$x \equiv a_n \pmod{m_n}$$

Scanned with CamScanner

The Chinese Remainder Theorem

Example: Consider the 3 congruences from Sun-Tsu's problem:

$x \equiv 2 \pmod 3$, $x \equiv 3 \pmod 5$, $x \equiv 2 \pmod 7$.

- Let $m = 3 \cdot 5 \cdot 7 = 105$, $M_1 = m/3 = 35$, $M_3 = m/5 = 21$, $M_3 = m/7 = 15$.
- We see that
  - 2 is an inverse of $M_1 = 35$ modulo 3 since $35 \cdot 2 \equiv 2 \cdot 2 \equiv 1 \pmod 3$
  - 1 is an inverse of $M_2 = 21$ modulo 5 since $21 \equiv 1 \pmod 5$
  - 1 is an inverse of $M_3 = 15$ modulo 7 since $15 \equiv 1 \pmod 7$
- Hence,

$$x = a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3 \pmod m$$
$$= 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 = 233 \equiv 23 \pmod{105}$$

- We have shown that 23 is the smallest positive integer that is a simultaneous solution. Check it!

Scanned with CamScanner

The Chinese Remainder Theorem

Word Problem:

Jessica breeds rabbits. She's not sure exactly how many she has today, but as she was moving them about this morning, she noticed some things. When she fed them, in groups of 5, she had 4 left over. When she bathed them, in groups of 8, she had a group of 6 left over. She took them outside to romp in groups of 9, but then the last group consisted of only 8. She's positive that there are fewer than 250 rabbits - but how many does she have?

Solution:

We have the following congurences

$x \equiv 4 \pmod 5$,

$x \equiv 6 \pmod 8$,

$x \equiv 8 \pmod 9$.

Scanned with CamScanner

Cryptography

Scanned with CamScanner

Section Summary

- Classical Cryptography
- Cryptosystems
- Public Key Cryptography
- RSA Cryptosystem
- Fermat's Little theorem

Scanned with CamScanner

Caesar Cipher

Julius Caesar created secret messages by shifting each letter three letters forward in the alphabet (sending the last three letters to the first three letters.) For example, the letter B is replaced by E and the letter X is replaced by A. This process of making a message secret is an example of *encryption*.

Here is how the encryption process works:

- Replace each letter by an integer from $\mathbf{Z}_{26}$, that is an integer from 0 to 25 representing one less than its position in the alphabet.
- The encryption function is $f(p) = (p + 3)$ **mod** 26. It replaces each integer p in the set $\{0,1,2,\ldots,25\}$ by $f(p)$ in the set $\{0,1,2,\ldots,25\}$.
- Replace each integer p by the letter with the position $p + 1$ in the alphabet.

Example: Encrypt the message "MEET YOU IN THE PARK" using the Caesar cipher.

Solution: 12 4 4 19 24 14 20 8 13 19 7 4 15 0 17 10.

Now replace each of these numbers p by $f(p) = (p + 3)$ **mod** 26.

15 7 7 22 1 17 23 11 16 22 10 7 18 3 20 13.

Translating the numbers back to letters produces the encrypted message

"PHHW BRX LQ WKH SDUN."

Scanned with CamScanner

Caesar Cipher

- To recover the original message, use $f^{-1}(p) = (p-3)$ **mod** 26. So, each letter in the coded message is shifted back three letters in the alphabet, with the first three letters sent to the last three letters. This process of recovering the original message from the encrypted message is called *decryption*.
- The Caesar cipher is one of a family of ciphers called *shift ciphers*. Letters can be shifted by an integer k, with 3 being just one possibility. The encryption function is

  $$f(p) = (p + k) \textbf{ mod } 26$$

  and the decryption function is

  $$f^{-1}(p) = (p-k) \textbf{ mod } 26$$

  The integer k is called a *key*.

Scanned with CamScanner

Shift Cipher

Example 1: Encrypt the message "STOP GLOBAL WARMING" using the shift cipher with $k = 11$.

Solution: Replace each letter with the corresponding element of $\mathbf{Z}_{26}$.

18 19 14 15 6 11 14 1 0 11 22 0 17 12 8 13 6.

Apply the shift $f(p) = (p + 11)$ **mod** 26, yielding

3 4 25 0 17 22 25 12 11 22 7 11 2 23 19 24 17.

Translating the numbers back to letters produces the ciphertext

"DEZA RWZMLW HLCXTYR."

Scanned with CamScanner

Shift Cipher

Example 2: Decrypt the message "LEWLYPLUJL PZ H NYLHA ALHJOLY" that was encrypted using the shift cipher with $k = 7$.

Solution: Replace each letter with the corresponding element of $\mathbf{Z}_{26}$.

11 4 22 11 24 15 11 20 9 11 15 25 7 13 24 11 7 0 0 11 7 9 14 11 24.

Shift each of the numbers by $-k = -7$ modulo 26, yielding

4 23 15 4 17 8 4 13 2 4 8 18 0 6 17 4 0 19 19 4 0 2 7 4 17.

Translating the numbers back to letters produces the decrypted message

"EXPERIENCE IS A GREAT TEACHER."

Scanned with CamScanner

Number Theory in Cryptography

Terminology: Two parties **Alice** and **Bob** want to communicate securely s.t. a third party **Eve** who intercepts messages cannot learn the content of the messages.

Symmetric Cryptosystems: Alice and Bob share a secret. Only they know a secret key K that is used to encrypt and decrypt messages. Given a message M, Alice encodes it (possibly with padding) into m, and then sends the ciphertext $encrypt(m, K)$ to Bob. Then Bob uses K to decrypt it and obtains $decrypt(encrypt(m, K), K) = m$.

Example: AES.

Public Key Cryptosystems: Alice and Bob do a-priori **not** share a secret. How can they establish a shared secret when others are listening to their messages?

Idea: Have a two-part key, i.e., a key pair. A public key that is used to encrypt messages, and a secret key to decrypt them. Alice uses Bob's public key to encrypt a message (everyone can do that). Only Bob can decrypt the message with his secret key.

Scanned with CamScanner

Description of RSA: Key generation

- Choose two distinct prime numbers p and q. Numbers p and q should be chosen at random, and be of similar bit-length. Prime integers can be efficiently found using a primality test.
- Let $n = pq$ and $k = (p-1)(q-1)$. (In particular, $k = |Z_n^*|$).
- Choose an integer e such that $1 < e < k$ and $gcd(e, k) = 1$; i.e., e and k are coprime.
  e (for encryption) is released as the public key exponent.
  (e must not be very small.)
- Let d be the multiplicative inverse of e modulo k, i.e., $de \equiv 1 (\mod k)$. (Computed using the extended Euclidean algorithm.) d (for decryption) is the private key and kept secret.

The public key is (n, e) and the private key is (n, d).

Scanned with CamScanner

RSA: Encryption and Decryption

Alice transmits her public key (n, e) to Bob and keeps the private key secret.

Encryption: Bob then wishes to send message M to Alice. He first turns M into an integer m, such that $0 \leq m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c corresponding to

$$c \equiv m^e \mod n$$

This can be done quickly using the method of exponentiation by squaring. Bob then transmits c to Alice.

Decryption: Alice can recover m from c by using her private key exponent d via computing

$$m \equiv c^d \mod n$$

Given m, she can recover the original message M by reversing the padding scheme.

Scanned with CamScanner

Using RSA

Given pubKey = <e, n> and privKey = <d, n>

If Message = m

Then:

encryption: $c = m^e \bmod n$, $m < n$

decryption: $m = c^d \bmod n$

signature: $s = md \bmod n$, $m < n$

verification: $m = se \bmod n$

Scanned with CamScanner

Example of RSA (1)

Choose p = 7 and q = 17.

Compute n = p*q= 119.

Compute f(n)=(p-1)(q-1)=96.

Select e = 5, (a relatively prime to f(n).)

Compute d = _77_such that e*d=1 mod f(n).

- Public key: <5,119>
- Private key: <77,119>
- Message = 19
- Encryption: 19^5 mod 119 = 66
- Decryption: 66^{77} mod 119 = 19

Scanned with CamScanner

Example of RSA (2)

p = 7, q = 11, n = 77

Alice chooses e = 17, making d = 53

Bob wants to send Alice secret message

HELLO (07 04 11 11 14)

– 07^{17} mod 77 = 28; 04^{17} mod 77 = 16

– 11^{17} mod 77 = 44; – 11^{17} mod 77 = 44

– 14^{17} mod 77 = 42

- Bob sends 28 16 44 44 42

Scanned with CamScanner

Example of RSA (3)

Alice receives **28 16 44 44 42**

Alice uses private key, d = 53, to decrypt message:

– 28^{53} mod 77 = 07; 16^{53} mod 77 = 04

– 44^{53} mod 77 = 11; 44^{53} mod 77 = 11

– 42^{53} mod 77 = 14

• Alice translates **07 04 11 11 14** to ***HELLO***

No one else could read it, as only Alice knows her private key (needed for decryption)

Scanned with CamScanner

Fermat's Little Theorem

Pierre de Fermat
(1601-1665)

Theorem 3: (*Fermat's Little Theorem*) If p is prime and a is an integer not divisible by p, then $a^{p-1} \equiv 1 \pmod{p}$

Furthermore, for every integer a we have $a^p \equiv a \pmod{p}$

(*proof outlined in Exercise 19*)

Fermat's little theorem is useful in computing the remainders modulo p of large powers of integers.

Example: Find 7^{222} **mod** 11.

By Fermat's little theorem, we know that $7^{10} \equiv 1 \pmod{11}$, and so $(7^{10})^k \equiv 1 \pmod{11}$, for every positive integer k. Therefore,

$$7^{222} = 7^{22 \cdot 10 + 2} = (7^{10})^{22} 7^2 \equiv (1)^{22} \cdot 49 \equiv 5 \pmod{11}.$$

Hence, 7^{222} **mod** $11 = 5$.

Scanned with CamScanner

The Foundations: Logic and Proofs

Scanned with CamScanner

Proofs

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

- Ingredients:
  - hypotheses of the theorem
  - axioms assumed to be true
  - previously proven theorems
  - rules of inference

You get:
truth of the statement being proved

Scanned with CamScanner

Usefulness

- Computer Science
  - Verifying that computer programs are correct.
  - Establishing that operating systems are secure.
  - Making inferences in artificial intelligence.
  - Showing that system specifications are consistent.

- Mathematics
  - Defining Formalism.
  - Providing specification in a common language.
  - Justification for the results.

Scanned with CamScanner

Definitions

1. An integer n is even if, and only if, $n = 2k$ for some integer k.
2. An integer n is odd if, and only if, $n = 2k + 1$ for some integer k.
3. An integer n is prime if, and only if, $n > 1$ and for all positive integers r and s, if $n = r \cdot s$, then $r = 1$ or $s = 1$.
4. An integer $n > 1$ is composite if, and only if, $n = r \cdot s$ for some positive integers r and s with $r \neq 1$ and $s \neq 1$.
5. A real number r is rational if, and only if, $r = \frac{a}{b}$ for some integers a and b with $b \neq 0$.
6. If n and d are integers and $d \neq 0$, then d divides n, written $d|n$ if, and only if, $n = d.k$ for some integers k.
7. An integer n is called a perfect square if, and only if, $n = k^2$ for some integer k.

Scanned with CamScanner

Types of Proofs

- **Proving conditional Statements**
  - Direct Proofs
  - Indirect Proofs
    - **Proof by Contraposition**
    - **Proofs by Contradiction**
- **Proving Non-conditional Statements**
  - Indirect Proofs
  - If-And-Only-If Proof
  - Constructive Versus Non-constructive Proofs
  - Existence Proofs; Existence and Uniqueness Proofs
  - Disproofs (Counterexample, Contradiction, Existence Statement)
  - Proofs Involving Sets
- **Mathematical Induction**

Scanned with CamScanner

Direct Proofs

- p →q
  - first step is the assumption that *p is true*
  - subsequent steps constructed using rules of inference.
  - final step showing that *q must also be true*

showing that if *p is true,*
then q must also be true,
so that the combination
p true and q false never occurs

Outline for Direct Proof

Proposition If P, then Q.

Proof. Suppose P.

⋮

Therefore Q. ■

Scanned with CamScanner

Activity Time

Prove that the sum of two odd integers is even.

Scanned with CamScanner

Prove that the sum of two odd integers is even.

Let **m** and **n** be two odd integers. Then by definition of odd numbers

$m = 2k + 1$ for some $k \in Z$

$n = 2l + 1$ for some $l \in Z$

Now $m + n = (2k + 1) + (2l + 1)$

$= 2k + 2l + 2$

$= 2(k + l + 1)$

$= 2r$ where $r = (k + l + 1) \in Z$

Hence $m + n$ is even.

Scanned with CamScanner

EXERCISE:

Prove that if n is any even integer, then $(-1)^n = 1$

SOLUTION:

Suppose n is an even integer. Then $n = 2k$ for some integer k.

Now

$$
\begin{aligned}
(-1)^n &= (-1)^{2k} \\
&= [(-1)^2]^k \\
&= (1)^k \\
&= 1 \qquad \text{(proved)}
\end{aligned}
$$

Scanned with CamScanner

EXERCISE:

Prove that the product of an even integer and an odd integer is even.

SOLUTION:

Suppose m is an even integer and n is an odd integer. Then

$m = 2k$ for some integer k

and $n = 2l + 1$ for some integer l

Now

$$m \cdot n = 2k \cdot (2l + 1)$$
$$= 2 \cdot k\,(2l + 1)$$
$$= 2 \cdot r \quad \text{where } r = k(2l + 1) \text{ is an integer}$$

Hence $m \cdot n$ is even. (Proved)

Scanned with CamScanner

EXERCISE:

Prove that the square of an even integer is even.

SOLUTION:

Suppose n is an even integer. Then $n = 2k$

Now

$$\begin{aligned} \text{square of } n = n^2 &= (2\cdot k)^2 \\ &= 4k^2 \\ &= 2\cdot(2k^2) \\ &= 2\cdot p \text{ where } p = 2k^2 \in Z \end{aligned}$$

Hence, n^2 is even. (proved)

Scanned with CamScanner

proved that if *n is an odd integer, then* n^2 *is an odd integer*

- We assume that the hypothesis of this conditional statement is true, namely, we assume that *n is odd*.
- By the definition of an odd integer, it follows that $n = 2k + 1$, *where k is some integer.*
- *Square both sides* $n^2 = (2k + 1)^2$
  - $4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1.$
- Consequently, we have proved that if *n is an odd integer, then* n^2 *is an odd integer*

Scanned with CamScanner

EXERCISE:

Prove that if n is an odd integer, then $n^3 + n$ is even.

SOLUTION:

Let n be an odd integer, then $n = 2k + 1$ for some $k \in Z$

Now $n^3 + n = n(n^2 + 1)$

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

$$= (2k + 1)(4k^2 + 4k + 1 + 1)$$

$$= (2k + 1)(4k^2 + 4k + 2)$$

$$= (2k + 1)\, 2.\, (2k^2 + 2k + 1)$$

$$= 2 \cdot (2k + 1)(2k^2 + 2k + 1) \qquad k \in Z$$

= an even integer

Scanned with CamScanner

Proposition If x is an even integer, then $x^2 - 6x + 5$ is odd.

Proof. Suppose x is an even integer.

Then $x = 2a$ for some $a \in \mathbb{Z}$, by definition of an even integer.

So $x^2 - 6x + 5 = (2a)^2 - 6(2a) + 5 = 4a^2 - 12a + 5 = 4a^2 - 12a + 4 + 1 = 2(2a^2 - 6a + 2) + 1$.

Therefore we have $x^2 - 6x + 5 = 2b + 1$, where $b = 2a^2 - 6a + 2 \in \mathbb{Z}$.

Consequently $x^2 - 6x + 5$ is odd, by definition of an odd number.

Scanned with CamScanner

EXERCISE:

Prove that, if the sum of any two integers is even, then so is their difference.

SOLUTION:

Suppose m and n are integers so that m + n is even. Then by definition of even numbers

$$m + n = 2k \qquad \text{for some integer } k$$

$$\Rightarrow \quad m = 2k - n \quad \ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Now $m - n = (2k - n) - n$ using (1)

$$= 2k - 2n$$

$$= 2(k - n) = 2r \qquad \text{where } r = k - n \text{ is an integer}$$

Hence m - n is even.

Scanned with CamScanner

EXERCISE:

Prove that the sum of any two rational numbers is rational.

SOLUTION:

Suppose r and s are rational numbers.

Then by definition of rational

$$r = \frac{a}{b} \quad \text{and} \quad s = \frac{c}{d}$$

for some integers a, b, c, d with $b \neq 0$ and $d \neq 0$

Now

$$r + s = \frac{a}{b} + \frac{c}{d}$$

$$= \frac{ad + bc}{bd}$$

$$= \frac{p}{q}$$

where $p = ad + bc \in Z$ and $q = bd \in Z$ and $q \neq 0$

Hence r + s is rational.

Scanned with CamScanner

EXERCISE:

Given any two distinct rational numbers r and s with $r < s$. Prove that there is a rational number x such that $r < x < s$.

SOLUTION:

Given two distinct rational numbers r and s such that

$$r < s \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Adding r to both sides of (1), we get

$$r + r < r + s$$

$$2r < r + s$$

$$\Rightarrow \qquad r < \frac{r+s}{2} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Next adding s to both sides of (1), we get

$$r + s < s + s$$

$$\Rightarrow \qquad r + s < 2s$$

$$\Rightarrow \qquad \frac{r+s}{2} < s \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

Scanned with CamScanner

Combining (2) and (3), we may write

$$r < \frac{r+s}{2} < s \quad \text{.....(4)}$$

Since the sum of two rationals is rational, therefore $r + s$ is rational. Also the quotient of a rational by a non-zero rational, is rational, therefore $\frac{r+s}{2}$ is rational and by (4) it lies between r & s. Hence, we have found a rational number such that $r < x < s$. (proved)

Scanned with CamScanner

EXERCISE:

Prove that the sum of any three consecutive integers is divisible by 3.

PROOF:

Let n, n + 1 and n + 2 be three consecutive integers.

Now

$$n + (n + 1) + (n + 2) = 3n + 3$$
$$= 3(n + 1)$$
$$= 3 \cdot k \quad \text{where } k = (n+1) \in Z$$

Hence, the sum of three consecutive integers is divisible by 3.

Scanned with CamScanner

Activity Time

Give a direct proof that if *m and n are both perfect squares, then nm is also a perfect square.*

Scanned with CamScanner

Proof

- We assume that the hypothesis of this conditional statement is true, namely, we assume that m and n are both perfect squares.
- By the definition of a perfect square, It follows that there are integers s and t such that $m = s^2$ and $n = t^2$.
- Multiplying both m and n to get s^2t^2.
- Hence, $mn = s^2t^2 = (ss)(tt) = (st)(st) = (st)^2$, using commutativity and associativity of multiplication.
- By the definition of perfect square, it follows that mn is also a perfect square, because it is the square of st, which is an integer.
- We have proved that if m and n are both perfect squares, then mn is also a perfect square.

Scanned with CamScanner

Activity Time

Give a direct proof that if *n is an integer and n is odd, then* $3n + 2$ *is odd.*

Scanned with CamScanner

Indirect Proofs

- Direct proof begin with the premises, continue with a sequence of deductions, and end with the conclusion.
- Attempts at direct proofs often reach dead ends
- Proofs that **do not** start with the premises and end with the conclusion, are called **indirect proofs**

Scanned with CamScanner

PROOF BY CONTRAPOSITION:

A proof by contraposition is based on the logical equivalence between a statement and its contrapositive. Therefore, the implication $p \rightarrow q$ can be proved by showing that its contrapositive $\sim q \rightarrow \sim p$ is true. The contrapositive is usually proved directly.

The method of proof by contrapositive may be summarized as:

1. Express the statement in the form if p then q.
2. Rewrite this statement in the contrapositive form
   if not q then not p.
3. Prove the contrapositive by a direct proof.

Outline for Contrapositive Proof

Proposition If P, then Q.

Proof. Suppose $\sim Q$.

$\vdots$

Therefore $\sim P$. ■

Scanned with CamScanner

Prove that if *n is an integer and 3n + 2 is odd, then n is odd.*

PROOF:

The contrapositive of the given conditional statement is

" if n is even then 3n + 2 is even"

Suppose n is even, then

$n = 2k$ for some $k \in Z$

Now $3n + 2 = 3(2k) + 2$

$= 2.(3k + 1)$

$= 2.r$ where $r = (3k + 1) \in Z$

Hence 3n + 2 is even. We conclude that the given statement is true since its contrapositive is true.

Scanned with CamScanner

EXERCISE:

Prove that for all integers n, if n^2 is even then n is even.

PROOF:

The contrapositive of the given statement is:

"if n is not even (odd) then n^2 is not even (odd)"

We prove this contrapositive statement directly.

Suppose n is odd. Then $n = 2k + 1$ for some $k \in Z$

Now $n^2 = (2k+1)^2 = 4k^2 + 4k + 1$

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

$= 2\cdot r + 1$ where $r = 2k^2 + 2k \in Z$

Hence n^2 is odd. Thus the contrapositive statement is true and so the given statement is true.

Scanned with CamScanner

EXERCISE:

Prove that if n is an integer and $n^3 + 5$ is odd, then n is even.

PROOF:

Suppose n is an odd integer. Since, a product of two odd integers is odd, therefore $n^2 = n.n$ is odd; and $n^3 = n^2.n$ is odd.

Since a sum of two odd integers is even therefore $n^2 + 5$ is even.

Thus we have prove that if n is odd then $n^3 + 5$ is even.

Since this is the contrapositive of the given conditional statement, so the given statement is true.

Scanned with CamScanner

EXERCISE:

Prove that if n^2 is not divisible by 25, then n is not divisible by 5.

SOLUTION:

The contra positive statement is:

"if n is divisible by 5, then n^2 is divisible by 25"

Suppose n is divisible by 5. Then by definition of divisibility

$$n = 5 \cdot k \qquad \text{for some integer } k$$

Squaring both sides

$$n^2 = 25 \cdot k^2 \qquad \text{where } k^2 \in Z$$

n^2 is divisible by 25

Scanned with CamScanner

Proofs by Contradiction

A proof by contradiction is based on the fact that either a statement is true or it is false but not both. Hence the supposition, that the statement to be proved is false, leads logically to a contradiction, impossibility or absurdity, then the supposition must be false. Accordingly, the given statement must be true.

The method of proof by contradiction may be summarized as follows:

1. *Suppose the statement to be proved is false.*
2. *Show that this supposition leads logically to a contradiction.*
3. *Conclude that the statement to be proved is true.*

Scanned with CamScanner

Basic Idea

- Assume that the statement we want to prove is *false, and then show* that this assumption leads to nonsense!

Outline for Proof by Contradiction

Proposition P.

Proof. Suppose $\sim P$.

$\vdots$

Therefore $C \wedge \sim C$. ■

We are then led to conclude that we were wrong to assume the statement was false, so the statement must be true.

Scanned with CamScanner

THEOREM:

There is no greatest integer.

PROOF:

Suppose there is a greatest integer N. Then $n \leq N$ for every integer n.

Let $M = N + 1$

Now M is an integer since it is a sum of integers.

Also $M > N$ since $M = N + 1$

Thus M is an integer that is greater than the greatest integer, which is a contradiction. Hence our supposition is not true and so there is no greatest integer.

Scanned with CamScanner

EXERCISE:

Give a proof by contradiction for the statement:

"If n^2 is an even integer then n is an even integer."

PROOF:

Suppose n^2 is an even integer and n is not even, so that n is odd.

Hence $n = 2k + 1$ for some integer k.

Now
$$\begin{aligned} n^2 &= (2k+1)^2 \\ &= 4k^2 + 4k + 1 \\ &= 2\cdot(2k^2 + 2k) + 1 \\ &= 2r + 1 \qquad \text{where } r = (2k^2 + 2k) \in Z \end{aligned}$$

This shows that n^2 is odd, which is a contradiction to our supposition that n^2 is even. Hence the given statement is true.

Scanned with CamScanner

EXERCISE:

Prove that if n is an integer and $n^3 + 5$ is odd, then n is even using contradiction method.

SOLUTION:

Suppose that $n^3 + 5$ is odd and n is not even (odd). Since n is odd and the product of two odd numbers is odd, it follows that n^2 is odd and $n^3 = n^2 . n$ is odd. Further, since the difference of two odd numbers is even, it follows that

$$5 = (n^3 + 5) - n^3$$

is even. But this is a contradiction. Therefore, the supposition that $n^3 + 5$ and n are both odd is wrong and so the given statement is true.

Scanned with CamScanner

THEOREM:

The sum of any rational number and any irrational number is irrational.

PROOF:

We suppose that the negation of the statement is true. That is, we suppose that there is a rational number r and an irrational number s such that r + s is rational. By definition of ration

$$r = \frac{a}{b} \quad \ldots\ldots\ldots\ldots\ldots(1) \qquad \text{and} \qquad r + s = \frac{c}{d} \quad \ldots\ldots\ldots\ldots\ldots(2)$$

for some integers a, b, c and d with $b \neq 0$ and $d \neq 0$.

Using (1) in (2), we get

$$\frac{a}{b} + s = \frac{c}{d}$$

$$\Rightarrow \qquad s = \frac{c}{d} - \frac{a}{b}$$

$$s = \frac{bc - ad}{bd} \qquad (bd \neq 0)$$

Now bc - ad and bd are both integers, since products and difference of integers are integers. Hence s is a quotient of two integers bc-ad and bd with $bd \neq 0$. So by definition of rational, s is rational.

This contradicts the supposition that s is irrational. Hence the supposition is false and the theorem is true.

Scanned with CamScanner

EXERCISE:

Prove that $\sqrt{2}$ is irrational.

PROOF:

Suppose $\sqrt{2}$ is rational. Then there are integers m and n with no common factors so

$$\sqrt{2} = \frac{m}{n}$$

that

Squaring both sides gives

$$2 = \frac{m^2}{n^2}$$

Or $m^2 = 2n^2$ .....(1)

This implies that m^2 is even (by definition of even). It follows that m is even. Hence

$m = 2k$ for some integer k (2)

Substituting (2) in (1), we get

$$(2k)^2 = 2n^2$$
$$\Rightarrow \quad 4k^2 = 2n^2$$
$$\Rightarrow \quad n^2 = 2k^2$$

This implies that n^2 is even, and so n is even. But we also know that m is even. Hence both m and n have a common factor 2. But this contradicts the supposition that m and n have no common factors. Hence our supposition is false and so the theorem is true.

Scanned with CamScanner

PROOF BY COUNTER EXAMPLE

Disprove the statement by giving a counter example.
For all real numbers a and b, if $a < b$ then $a^2 < b^2$.

SOLUTION:

Suppose $a = -5$ and $b = -2$
then clearly $-5 < -2$

But $a^2 = (-5)^2 = 25$ and $b^2 = (-2)^2 = 4$

But $25 > 4$

This disproves the given statement.

Scanned with CamScanner

EXERCISE:

Prove or give counter example to disprove the statement.

For all integers n, $n^2 - n + 11$ is a prime number.

SOLUTION:

The statement is not true

For n = 11

we have , $n^2 - n + 11 = (11)^2 - 11 + 11$

$= (11)^2$

$= (11)(11)$

$= 121$

which is obviously not a prime number.

Scanned with CamScanner

Mathematical Induction

Shoaib Raza

Scanned with CamScanner

Conjecture: The sum of the first n odd natural numbers equals n^2.

n	sum of the first n odd natural numbers	n^2
1	$1 =$	1
2	$1+3 =$	4
3	$1+3+5 =$	9
4	$1+3+5+7 =$	16
5	$1+3+5+7+9 =$	25
$\vdots$	$\vdots$	$\vdots$
n	$1+3+5+7+9+11+\cdots+(2n-1) =$	n^2
$\vdots$	$\vdots$	$\vdots$

Scanned with CamScanner

An infinite ladder

- Suppose that we have an infinite ladder, and we want to know whether we can reach every step on this ladder.

- We know two things:
  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.

Scanned with CamScanner

FIGURE 1 **Climbing an Infinite Ladder.**

Scanned with CamScanner

Mathematical Induction

- Mathematical statements assert that a property is true for all positive integers.

- Proofs using mathematical induction have two parts.
  - First, they show that the statement holds for the positive integer 1 (base case).
  - Second, they show that if the statement holds for a positive integer then it must also hold for the next larger integer. (inductive case)

- The method can be extended to prove statements about more general well-founded structures, such as trees; this generalization, known as structural induction, is used in mathematical logic and computer science.

Scanned with CamScanner

NOTE

- It is extremely important to note that mathematical induction can be used only to prove results obtained in some other way.
- It is *not a tool for discovering formulae or* theorems.
- Mathematicians sometimes find proofs by mathematical induction unsatisfying because they do not provide insights as to why theorems are true.
- You can prove a theorem by mathematical induction even if you do not have the slightest idea why it is true!

Scanned with CamScanner

FIGURE 2 Illustrating How Mathematical Induction Works Using Dominoes.

Scanned with CamScanner

The Simple Idea Behind Mathematical Induction

Statements are lined up like dominoes.

(1) Suppose the first statement falls (i.e. is proved true);

(2) Suppose the kth falling always causes the $(k+1)^{th}$ to fall;

Then all must fall (i.e. all statements are proved true).

Scanned with CamScanner

PRINCIPLE OF MATHEMATICAL INDUCTION:

Let P(n) be a propositional function defined for all positive integers n. P(n) is true for every positive integer n if

1.Basis Step:

The proposition P(1) is true.

2.Inductive Step:

If P(k) is true then P(k + 1) is true for all integers $k \geq 1$.

i.e. $\forall k \quad p(k) \rightarrow P(k+1)$

Scanned with CamScanner

EXAMPLE:

Use Mathematical Induction to prove that

$$1+2+3+\cdots+n=\frac{n(n+1)}{2} \quad \text{for all integers } n \geq 1$$

SOLUTION:

Let $$P(n): 1+2+3+\cdots+n=\frac{n(n+1)}{2}$$

1.Basis Step:

P(1) is true.

For n = 1, left hand side of P(1) is the sum of all the successive integers starting at 1 and ending at 1, so LHS = 1 and RHS is

$$R.H.S=\frac{1(1+1)}{2}=\frac{2}{2}=1$$

so the proposition is true for n = 1.

2. Inductive Step: Suppose P(k) is true for, some integers $k \geq 1$.

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

Scanned with CamScanner

To prove P(k + 1) is true. That is,

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

Consider L.H.S. of (2)

$$\begin{aligned}
1+2+3+\cdots+(k+1) &= 1+2+3+\cdots+k+(k+1) \\
&= \frac{k(k+1)}{2}+(k+1) \qquad \text{using (1)} \\
&= (k+1)\left[\frac{k}{2}+1\right] \\
&= (k+1)\left[\frac{k+2}{2}\right] \\
&= \frac{(k+1)(k+2)}{2} = \text{RHS of (2)}
\end{aligned}$$

Hence by principle of Mathematical Induction the given result true for all integers greater or equal to 1.

Scanned with CamScanner

EXERCISE:

Use mathematical induction to prove that
$1+3+5+\ldots+(2n-1) = n^2$ for all integers $n \geq 1$.

SOLUTION:

Let P(n) be the equation $1+3+5+\ldots+(2n-1) = n^2$

1. **Basis Step:**

   P(1) is true
   For n = 1, L.H.S of P(1) = 1and
   R.H.S = $1^2 = 1$
   Hence the equation is true for n = 1

2. **Inductive Step:**

   Suppose P(k) is true for some integer $k \geq 1$. That is,
   $1 + 3 + 5 + \ldots + (2k - 1) = k^2$ .....(1)

To prove P(k+1) is true; i.e.,
$1 + 3 + 5 + \ldots + [2(k+1)-1] = (k+1)^2$ .....(2)

Consider L.H.S. of (2)

$$\begin{aligned}
1+3+5+\cdots+[2(k+1)-1] &= 1+3+5+\cdots+(2k+1) \\
&= 1+3+5+\cdots+(2k-1)+(2k+1) \\
&= k^2+(2k+1) \qquad \text{using (1)} \\
&= (k+1)^2 \\
&= \text{R.H.S. of (2)}
\end{aligned}$$

Thus P(k+1) is also true. Hence by mathematical induction, the given equation is true for all integers $n \geq 1$.

Scanned with CamScanner

Exercise (cont.)

Proof.

1. $P(n)$: $2^0 + 2^1 + 2^2 + \cdots + 2^n = 2^{n+1} - 1$
2. Basis step $P(0)$: $2^0 = 1 = 2^{0+1} - 1$.
3. Inductive step:
   Inductive hypothesis $P(k)$: $2^0 + 2^1 + 2^2 + \cdots + 2^k = 2^{k+1} - 1$
   Let's prove $P(k+1)$:

$$\begin{aligned} 2^0 + 2^1 + 2^2 + \cdots + 2^k + 2^{k+1} &= 2^{k+1} - 1 + 2^{k+1} && \text{(by IH)} \\ &= 2(2^{k+1}) - 1 && \text{(by arithmetic)} \\ &= 2^{k+2} - 1 && \text{(by arithmetic)} \end{aligned}$$

Scanned with CamScanner

Counting

Chapter 6

Mr. Shoaib Raza

Scanned with CamScanner

Chapter Summary

- The Basics of Counting
- The Pigeonhole Principle
- Permutations and Combinations
- Binomial Coefficients and Identities
- Generalized Permutations and Combinations

Scanned with CamScanner

The Basics of Counting

Section 6.1

Scanned with CamScanner

COMBINATORICS

- Combinatorics is the mathematics of counting and arranging objects. Counting of objects with certain properties (enumeration) is required to solve many different types of problem.
- Applications, include topics as diverse as codes, circuit design and algorithm complexity [and gambling]

Scanned with CamScanner

Counting

- Enumeration, the counting of objects with certain properties, is an important part of combinatorics.
- We must count objects to solve many different types of problems. For example, counting is used to:

1. Determine number of ordered or unordered arrangement of objects.
2. Generate all the arrangements of a specified kind which is important in computer simulations.
3. Compute probabilities of events.
4. Analyze the chance of winning games, lotteries etc.
5. Determine the complexity of algorithms.

Scanned with CamScanner

Section Summary

- The Sum Rule
- The Product Rule
- The Subtraction Rule
- The Division Rule
- Examples, Examples, and Examples
- Tree Diagrams

Scanned with CamScanner

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 to do the second task, 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.

Scanned with CamScanner

The Sum Rule in terms of sets.

- The sum rule can be phrased in terms of sets.

  $|A \cup B| = |A| + |B|$ as long as A and B are disjoint sets.

- Or more generally,

$$|A_1 \cup A_2 \cup \cdots \cup A_m| = |A_1| + |A_2| + \cdots + |A_m|$$

when $A_i \cap A_j = \emptyset$ for all i, j.

- The case where the sets have elements in common will be discussed when we consider the subtraction rule and taken up fully in Chapter 8.

Scanned with CamScanner

Basic Counting Principles: The Sum Rule

Example:

Suppose there are 7 different optional courses in Computer Science and 3 different optional courses in Mathematics. How many ways student can choose a course.

Solution: By the sum rule it follows that there are $7 + 3 = 10$ choices for a student who wants to take one optional course.

Scanned with CamScanner

Basic Counting Principles: The Sum Rule

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.

Scanned with CamScanner

Basic Counting Principles: The Sum Rule

Example: A student can choose a computer project from one of the three lists. The three lists contain 23, 15 and 19 possible projects, respectively. How many possible projects are there to choose from?

Solution: The student can choose a project from the first list in 23 ways, from the second list in 15 ways, and from the third list in 19 ways. Hence, there are

$23 + 15 + 19 = 57$ projects to choose from.

Scanned with CamScanner

Basic Counting Principles: The Product Rule

The Product Rule: A procedure can be broken down into a sequence of two tasks. There are n_1 ways to do the first task and n_2 ways to do the second task. Then there are $n_1 \cdot n_2$ ways to do the procedure.

Scanned with CamScanner

Product Rule in Terms of Sets

- If $A_1, A_2, \ldots, A_m$ are finite sets, then the number of elements in the Cartesian product of these sets is the product of the number of elements of each set.
- The task of choosing an element in the Cartesian product $A_1 \times A_2 \times \cdots \times A_m$ is done by choosing an element in A_1, an element in A_2 , ..., and an element in A_m.
- By the product rule, it follows that:
  $|A_1 \times A_2 \times \cdots \times A_m| = |A_1| \cdot |A_2| \cdot \cdots \cdot |A_m|$.

Scanned with CamScanner

The Product Rule

Example: How many ways a student can choose one optional course each from computer science and mathematics courses if there are 7 different optional courses in Computer Science and 3 different optional courses in Mathematics.

Solution:

A student who wants to take one optional course of each subject, there are $7 \times 3 = 21$ choices.

Scanned with CamScanner

The Product Rule

Example: The chairs of an auditorium are to be labeled with two characters, a letter followed by a digit. What is the largest number of chairs that can be labeled differently?

Solution:

The procedure of labeling a chair consists of two events, namely,

Assigning one of the 26 letters: A, B, C, ..., Z and

Assigning one of the 10 digits: 0, 1, 2, ..., 9

By product rule, there are $26 \times 10 = 260$ different ways that a chair can be labeled by both a letter and a digit.

Scanned with CamScanner

The Product Rule

Example: Find the number n of ways that an organization consisting of 15 members can elect a president, treasurer, and secretary. (assuming no person is elected to more than one position)

Solution:

The president can be elected in 15 different ways; following this, the treasurer can be elected in 14 different ways; and following this, the secretary can be elected in 13 different ways. Thus, by product rule, there are

$$n = 15 \times 14 \times 13 = 2730$$

different ways in which the organization can elect the officers.

Scanned with CamScanner

The Product Rule

Example: There are four bus lines between A and B; and three bus lines between B and C.

Find the number of ways a person can travel:

a) By bus from A to C by way of B;

b) Round trip by bus from A to C by way of B;

c) Round trip by bus from A to C by way of B, if the person does not want to use a bus line more than once.

Scanned with CamScanner

The Product Rule

a) By bus from A to C by way of B;

Solution:

$$A \xrightarrow{4} B \xrightarrow{3} C$$

There are 4 ways to go from A to B and 3 ways to go from B to C; hence there are $4 \times 3 = 12$ ways to go from A to C by way of B.

Scanned with CamScanner

The Product Rule

b) Round trip by bus from A to C by way of B;

Solution:

The person will travel from A to B to C to B to A for the round trip. i.e. (A →B →C →B →A)

$$A \xrightarrow{4} B \xrightarrow{3} C \xrightarrow{3} B \xrightarrow{4} A$$

The person can travel 4 ways from A to B and 3 way from B to C and back.

Thus there are 4 ×3 ×3 × 4 = 144 ways to travel the round trip.

Scanned with CamScanner

The Product Rule

c) Round trip by bus from A to C by way of B, if the person does not want to use a bus line more than once.

Solution:

$$A \xrightarrow{4} B \xrightarrow{3} C \xrightarrow{2} B \xrightarrow{3} A$$

The person can travel 4 ways from A to B and 3 ways from B to C, but only 2 ways from C to B and 3 ways from B to A, since bus line cannot be used more than once. Hence there are

$4 \times 3 \times 2 \times 3 = 72$ ways

to travel the round trip without using a bus line more than once.

Scanned with CamScanner

The Product Rule

Example: A bit string is a sequence of 0's and 1's.How many bit string are there of length 4?

Solution:

Each bit (binary digit) is either **0** or **1**.

Hence, there are **2** ways to choose each bit. Since we have to choose four bits therefore,

$$2 \times 2 \times 2 \times 2 = 2^4 = 16$$

the product rule shows, there are a total of different bit strings of length four.

Scanned with CamScanner

The Product Rule

Example: How many bit strings of length 8:

(i) begin with a 1?

(ii) begin and end with a 1?

Solution:

(i) If the first bit (left most bit) is a 1, then it can be filled in only one way. Each of the remaining seven positions in the bit string can be filled in 2 ways (i.e., either by 0 or 1). Hence, there are

$$1 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^7 = 128$$

different bit strings of length 8 that begin with a 1.

Scanned with CamScanner

The Product Rule

(ii) begin and end with a 1?

Solution:

If the first and last bit in an 8 bit string is a 1, then only the intermediate six bits can be filled in 2 ways, i.e. by a 0 or 1. Hence there are

$$1 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 1 = 2^6 = 64$$

different bit strings of length 8 that begin and end with a 1.

Scanned with CamScanner

The Product Rule

Example: Suppose that an automobile license plate has three letters followed by three digits.

(a) How many different license plates are possible?

Solution:

Each of the three letters can be written in 26 different ways, and each of the three digits can be written in 10 different ways.

Hence, by the product rule, there is a total of

$26 \times 26 \times 26 \times 10 \times 10 \times 10 = 17{,}576{,}000$

different License plates possible.

Scanned with CamScanner

The Product Rule

(b) How many license plates could begin with A and end on 0?

Solution:

The first and last place can be filled in one way only, while each of second and third place can be filled in 26 ways and each of fourth and fifth place can be filled in 10 ways.

Number of license plates that begin with A and end in 0 are

$$1 \times 26 \times 26 \times 10 \times 10 \times 1 = 67600$$

Scanned with CamScanner

The Product Rule

(c) How many license plates begin with PQR.

Solution:

Number of license plates that begin with PQR are

$$1 \times 1 \times 1 \times 10 \times 10 \times 10 = 1000 \text{ ways.}$$

Scanned with CamScanner

The Product Rule

(d) How many license plates are possible in which all the letters and digits are distinct?

Solution:

The first letter place can be filled in 26 ways. Since, the second letter place should contain a different letter than the first, so it can be filled in 25 ways. Similarly, the third letter place can be filled in 24 ways. And the digits can be respectively filled in 10, 9, and 8 ways.

Hence;

number of license plates in which all the letters and digits are distinct are

$$26 \times 25 \times 24 \times 10 \times 9 \times 8 = 11,232,000$$

Scanned with CamScanner

The Product Rule

(e) How many license plates could begin with AB and have all three letters and digits distinct.

Solution:

The first two letters places are fixed (to be filled with A and B), so there is only one way to fill them. The third letter place should contain a letter different from A & B, so there

are 24 ways to fill it.

The three digit positions can be filled in 10 and 8 ways to have distinct digits. Hence, desired number of license plates are

$$1 \times 1 \times 24 \times 10 \times 9 \times 8 = 17280$$

Scanned with CamScanner

Telephone Numbering Plan

Example: The *North American numbering plan* (*NANP*) specifies that a telephone number consists of 10 digits, consisting of a three-digit area code, a three-digit office code, and a four-digit station code. There are some restrictions on the digits.

- Let X denote a digit from 0 through 9.
- Let N denote a digit from 2 through 9.
- Let Y denote a digit that is 0 or 1.
- In the old plan (in use in the 1960s) the format was *NYX-NNX-XXXX*.
- In the new plan, the format is *NXX-NXX-XXXX*.

How many different telephone numbers are possible under the old plan and the new plan?

Solution: Use the Product Rule.

- There are $8 \cdot 2 \cdot 10 = 160$ area codes with the format *NYX*.
- There are $8 \cdot 10 \cdot 10 = 800$ area codes with the format *NXX*.
- There are $8 \cdot 8 \cdot 10 = 640$ office codes with the format *NNX*.
- There are $10 \cdot 10 \cdot 10 \cdot 10 = 10{,}000$ station codes with the format *XXXX*.

Number of old plan telephone numbers: $160 \cdot 640 \cdot 10{,}000 = 1{,}024{,}000{,}000$.

Number of new plan telephone numbers: $800 \cdot 800 \cdot 10{,}000 = 6{,}400{,}000{,}000$.

Scanned with CamScanner

NUMBER OF ITERATIONS OF A NESTED LOOP

Example: Determine how many times the inner loop will be iterated when the following algorithm is implemented and run

For i: = 1 to 4

For j : = 1 to 3

[Statement in body of inner loop. None contain branching statements that lead out of the inner loop.]

next j

next i

<u>Solution:</u>

The outer loop is iterated four times, and during each iteration of the outer loop, there are three iterations of the inner loop.

Hence, by product rules the total number of iterations of inner loop is $4 \cdot 3 = 12$

Scanned with CamScanner

Example: Determine how many times the inner loop will be iterated when the following algorithm is implemented and run.

```
for     i = 5 to 50
for     j: = 10 to 20
```

[Statement in body of inner loop. None contain branching statements that lead out of the inner loop.]

```
        next j
        next i
```

Solution:

The outer loop is iterated 50 - 5 + 1 = 46 times and during each iteration of the outer loop there are 20 - 10 + 1 = 11 iterations of the inner loop. Hence by product rule, the total number of iterations of the inner loop is 46×11 = 506.

Scanned with CamScanner

Example: Determine how many times the inner loop will be iterated when the following algorithm is implemented and run.

```
for     i: = 1 to 4
for     j: = 1 to i
        [Statements in body of inner loop. None contain branching statements that lead outside the loop.]
next j
next i
```

Solution:

The outer loop is iterated 4 times, but during each iteration of the outer loop, the inner loop iterates different number of times.

For first iteration of outer loop, inner loop iterates 1 times.

For second iteration of outer loop, inner loop iterates 2 times.

For third iteration of outer loop, inner loop iterates 3 times.

For fourth iteration of outer loop, inner loop iterates 4 times.

Hence, total number of iterations of inner loop = 1 + 2 + 3 + 4 = 10.

Scanned with CamScanner

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.

Solution:

- First consider variable names one character in length. Since such names consist of a single letter, there are 26 variable names of length 1.
- Next, consider variable names two characters in length. Since the first character is a letter, there are 26 ways to choose it. The second character is a digit, there are 10 ways to choose it. Hence, to construct variable name of two characters in length, there are $26 \times 10 = 260$ ways.
- Finally, by sum rule, there are $26 + 260 = 286$ possible variable names in the programming language.

Scanned with CamScanner

Combining the Sum and Product Rule

Example: A computer access code word consists of from one to three letters of English alphabets with repetitions allowed. How many different code words are possible.

Solution:

Number of code words of length 1 = 26^1

Number of code words of length 2 = 26^2

Number of code words of length 3 = 26^3

Hence, the total number of code words =

$$26^1 + 26^2 + 26^3 = 18{,}278$$

Scanned with CamScanner

Counting Passwords

- Combining the sum and product rule allows us to solve more complex problems.

Example: Each user on a computer system has a password, which is six to eight characters long, where each character is an uppercase letter or a digit. Each password must contain 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 6, 7, and 8.

- By the sum rule $P = P_6 + P_7 + P_8$.

Finding P_6 directly is difficult. To find P_6 it is easier to find the number of strings of uppercase letters and digits that are six characters long, including those with no digits, and subtract from this the number of strings with no digits. By the product rule, the number of strings of six characters is 36^6 ,and the number of strings with no digits is 26^6

Scanned with CamScanner

Counting Passwords(Continued)

- To find each of P6, P7, and P8 , we find the number of passwords of the specified length composed of letters and digits and subtract the number composed only of letters. We find that:
- $P_6 = 36^6 - 26^6 = 2{,}176{,}782{,}336 - 308{,}915{,}776 = 1{,}867{,}866{,}560.$
- $P_7 = 36^7 - 26^7 = 78{,}364{,}164{,}096 - 8{,}031{,}810{,}176 = 70{,}332{,}353{,}920.$
- $P_8 = 36^8 - 26^8 = 2{,}821{,}109{,}907{,}456 - 208{,}827{,}064{,}576 = 2{,}612{,}282{,}842{,}880.$
- Consequently, $P = P_6 + P_7 + P_8 = 2{,}684{,}483{,}063{,}360.$

Scanned with CamScanner

Internet Addresses

- Version 4 of the Internet Protocol (IPv4) uses 32 bits.

Bit Number	0	1	2	3	4	8	16	24	31
Class A	0	netid				hostid			
Class B	1	0	netid				hostid		
Class C	1	1	0	netid				hostid	
Class D	1	1	1	0	Multicast Address				
Class E	1	1	1	1	0	Address			

- **Class A Addresses**: used for the largest networks, a 0,followed by a 7-bit netid and a 24-bit hostid.
- **Class B Addresses**: used for the medium-sized networks, a 10,followed by a 14-bit netid and a 16-bit hostid.
- **Class C Addresses**: used for the smallest networks, a 110,followed by a 21-bit netid and a 8-bit hostid.
  - Neither Class D nor Class E addresses are assigned as the address of a computer on the internet. Only Classes A, B, and C are available.
  - 1111111 is not available as the netid of a Class A network.
  - Hostids consisting of all 0s and all 1s are not available in any network.

Scanned with CamScanner

Counting Internet Addresses

Example: How many different IPv4 addresses are available for computers on the internet?

Solution: Use both the sum and the product rule. Let x be the number of available addresses, and let x_A, x_B, and x_C denote the number of addresses for the respective classes.

- To find, x_A: $2^7 - 1 = 127$ netids. $2^{24} - 2 = 16{,}777{,}214$ hostids.
  $$x_A = 127 \cdot 16{,}777{,}214 = 2{,}130{,}706{,}178.$$
- To find, x_B: $2^{14} = 16{,}384$ netids. $2^{16} - 2 = 16{,}534$ hostids.
  $$x_B = 16{,}384 \cdot 16{,}534 = 1{,}073{,}709{,}056.$$
- To find, x_C: $2^{21} = 2{,}097{,}152$ netids. $2^8 - 2 = 254$ hostids.
  $$x_C = 2{,}097{,}152 \cdot 254 = 532{,}676{,}608.$$
- Hence, the total number of available IPv4 addresses is

$$\begin{aligned} x &= x_A + x_B + x_C \\ &= 2{,}130{,}706{,}178 + 1{,}073{,}709{,}056 + 532{,}676{,}608 \\ &= 3{,}737{,}091{,}842. \end{aligned}$$

Not Enough Today !!
The newer IPv6 protocol solves the problem of too few addresses.

Scanned with CamScanner

Basic Counting Principles: Subtraction Rule

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.

- Also known as, the *principle of inclusion-exclusion*:

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

Scanned with CamScanner

Counting Bit Strings

Example: How many bit strings of length eight start either with a 1 bit or end with the two bits 00?

Solution: Use the subtraction rule.

- Number of bit strings of length eight that start with a 1 bit: $2^7 = 128$
- Number of bit strings of length eight that end with bits 00: $2^6 = 64$
- Number of bit strings of length eight that start with a 1 bit and end with bits 00 : $2^5 = 32$

Hence, the number is $128 + 64 - 32 = 160$.

Scanned with CamScanner

Counting Functions

Counting Functions: How many functions are there from a set with m elements to a set with n elements?

Solution: Since a function represents a choice of one of the n elements of the codomain for each of the m elements in the domain, the product rule tells us that there are $n \cdot n \cdots n = n^m$ such functions.

Counting One-to-One Functions: How many one-to-one functions are there from a set with m elements to one with n elements?

Solution: Suppose the elements in the domain are $a_1, a_2, \ldots, a_m$. There are n ways to choose the value of a_1 and $n-1$ ways to choose a_2, etc. The product rule tells us that there are $n(n-1)(n-2)\cdots(n-m+1)$ such functions.

Scanned with CamScanner

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 shirts that the campus book store needs to stock to have one of each size and color available?

Scanned with CamScanner

Tree Diagrams

- **Solution**: Draw the tree diagram.

W = white, R = red, G = green, B = black

- The store must stock 17 T-shirts.

Scanned with CamScanner

The Pigeonhole Principle

Section 6.2

Scanned with CamScanner

Section Summary

- The Pigeonhole Principle
- The Generalized Pigeonhole Principle

Scanned with CamScanner

The Pigeonhole Principle

Pigeonhole Principle: If k is a positive integer and $k + 1$ objects are placed into k boxes, then at least one box contains two or more objects.

Proof: We use a proof by contraposition. Suppose none of the k boxes has more than one object. Then the total number of objects would be at most k. This contradicts the statement that we have $k + 1$ objects.

Scanned with CamScanner

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.

(a) (b) (c)

Scanned with CamScanner

The Pigeonhole Principle

Corollary 1: A function f from a set with $k + 1$ elements to a set with k elements is not one-to-one.

Proof: Use the pigeonhole principle.

- Create a box for each element y in the codomain of f.
- Put in the box for y all of the elements x from the domain such that $f(x) = y$.
- Because there are $k + 1$ elements and only k boxes, at least one box has two or more elements.

Hence, f can't be one-to-one. ◀

Scanned with CamScanner

The Generalized Pigeonhole Principle

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.

Proof: We use a proof by contraposition. Suppose that none of the boxes contains more than $\lceil N/k \rceil - 1$ objects. Then the total number of objects is at most

$$k\left(\left\lceil \frac{N}{k} \right\rceil - 1\right) < k\left(\left(\frac{N}{k} + 1\right) - 1\right) = N,$$

where the inequality $\lceil N/k \rceil < \lceil N/k \rceil + 1$ has been used. This is a contradiction because there are a total of n objects. ◀

Scanned with CamScanner

Pigeonhole Principle

Example: Among any group of 367 people, there must be at least two with the same birthday, because there are only 366 possible birthdays. $\lceil 367/366 \rceil = 2$

Example: Among 100 people there are at least $\lceil 100/12 \rceil = 9$ who were born in the same month.

Example: In any set of 27 English , must be at least two that begin with the same letter, since there are 26 letters in the English alphabet. $\lceil 27/26 \rceil = 2$

Scanned with CamScanner

The Generalized Pigeonhole Principle

Example: 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 guarantee that at least six students receive the same grade is the smallest integer N such that $\lceil N/K \rceil = \lceil N/5 \rceil = 6$. The smallest such integer is

$$N = K(\lceil N/K \rceil - 1) + 1 = 5(6-1)+1 = 5 \cdot 5 + 1 = 26.$$

Thus 26 is the minimum number of students needed to be sure that at least 6 students will receive the same grades.

Scanned with CamScanner

Permutations and Combinations

Section 6.3

Scanned with CamScanner

Section Summary

- Permutations
- Combinations
- Combinatorial Proofs

Scanned with CamScanner

Permutations

Definition: 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*.

Example: Let $S = \{1,2,3\}$.

- The ordered arrangement 3,1,2 is a permutation of S.
- The ordered arrangement 3,2 is a 2-permutation of S.

- The number of r-permutations of a set with n elements is denoted by $P(n,r)$.
- The 2-permutations of $S = \{1,2,3\}$ are 1,2; 1,3; 2,1; 2,3; 3,1; and 3,2. Hence, $P(3,2) = 6$.

Scanned with CamScanner

A 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

$$P(n, r) = n(n-1)(n-2)\cdots(n-r+1)$$

r-permutations of a set with n distinct elements.

Proof: Use the product rule. The first element can be chosen in n ways. The second in $n-1$ ways, and so on until there are $(n-(r-1))$ ways to choose the last element.

- Note that $P(n,0) = 1$, since there is only one way to order zero elements.

Corollary 1: If n and r are integers with $1 \leq r \leq n$, then

$$P(n, r) = \frac{n!}{(n-r)!}$$

Scanned with CamScanner

Solving Counting Problems by Counting Permutations

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

Solution:

$$P(100,3) = 100 \cdot 99 \cdot 98 = 970{,}200$$

Scanned with CamScanner

Solving Counting Problems by Counting Permutations (*continued*)

- **Example**: Suppose that there are eight runners in a race. The winner receives a gold medal, the second place finisher receives a silver medal, and the third-place finisher receives a bronze medal. How many different ways are there to award these medals, if all possible outcomes of the race can occur and there are no ties?
- **Solution**: The number of different ways to award the medals is the number of 3-permutations of a set with eight elements. Hence, there are

$$P(8, 3) = 8 \cdot 7 \cdot 6 = 336$$

possible ways to award the medals.

Scanned with CamScanner

Solving Counting Problems by Counting Permutations (*continued*)

Example: Suppose that a saleswoman has to visit eight different cities. She must begin her trip in a specified city, but she can visit the other seven cities in any order she wishes. How many possible orders can the saleswoman use when visiting these cities?

Solution: The first city is chosen, and the rest are ordered arbitrarily. Hence the orders are:

$$P(7,7) = 7! = 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 5040$$

If she wants to find the tour with the shortest path that visits all the cities, she must consider 5040 paths!

Scanned with CamScanner

Solving Counting Problems by Counting Permutations (*continued*)

Example: How many permutations of the letters *ABCDEFGH* contain the string *ABC* ?

Solution: We solve this problem by counting the permutations of six objects, *ABC*, *D*, *E*, *F*, *G*, and *H*.

$$P(6,6) = 6! = 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 720$$

Scanned with CamScanner

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 used and is called a *binomial coefficient*. (*We will see the notation again in the binomial theorem in Section*. 6.4)

Scanned with CamScanner

Combinations

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 $\{c, d\}$.

Scanned with CamScanner

Combinations

Theorem 2: The number of r-combinations of a set with n elements, where $n \geq r \geq 0$, equals

$$C(n,r) = \frac{n!}{(n-r)!r!}.$$

Proof: By the product rule $P(n, r) = C(n,r) \cdot P(r,r)$. Therefore,

$$C(n,r) = \frac{P(n,r)}{P(r,r)} = \frac{n!/(n-r)!}{r!/(r-r)!} = \frac{n!}{(n-r)!r!}.$$

Scanned with CamScanner

Combinations

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:

$$C(52,5) = \frac{52!}{5!47!}$$
$$= \frac{52\cdot 51\cdot 50\cdot 49\cdot 48}{5\cdot 4\cdot 3\cdot 2\cdot 1} = 26\cdot 17\cdot 10\cdot 49\cdot 12 = 2,598,960$$

- The different ways to select 47 cards from 52 is

$$C(52,47) = \frac{52!}{47!5!} = C(52,5) = 2,598,960.$$

This is a special case of a general result. →

Scanned with CamScanner

Combinations

Corollary 2: Let n and r be nonnegative integers with $r \leq n$. Then $C(n, r) = C(n, n - r)$.

Proof: From Theorem 2, it follows that

$$C(n, r) = \frac{n!}{(n-r)!r!}$$

and

$$C(n, n-r) = \frac{n!}{(n-r)![n-(n-r)]!} = \frac{n!}{(n-r)!r!} .$$

Hence, $C(n, r) = C(n, n - r)$. ◀

This result can be proved without using algebraic manipulation. →

Scanned with CamScanner

Combinatorial Proofs

- **Definition 1**: A *combinatorial proof* of an identity is a proof that uses one of the following methods.
  - A *double counting proof* uses counting arguments to prove that both sides of an identity count the same objects, but in different ways.
  - A *bijective proof* shows that there is a bijection between the sets of objects counted by the two sides of the identity.

Scanned with CamScanner

Combinatorial Proofs

- Here are two combinatorial proofs that

$$C(n, r) = C(n, n - r)$$

when r and n are nonnegative integers with $r < n$:

- *Bijective Proof*: Suppose that S is a set with n elements. The function that maps a subset A of S to $\bar{A}$ is a bijection between the subsets of S with r elements and the subsets with $n - r$ elements. Since there is a bijection between the two sets, they must have the same number of elements. ◀
- *Double Counting Proof*: By definition the number of subsets of S with r elements is $C(n, r)$. Each subset A of S can also be described by specifying which elements are not in A, i.e., those which are in $\bar{A}$. Since the complement of a subset of S with r elements has $n - r$ elements, there are also $C(n, n - r)$ subsets of S with r elements. ◀

Scanned with CamScanner

Combinations

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

Solution: By Theorem 2, the number of combinations is

$$C(10,5) = \frac{10!}{5!5!} = 252.$$

Example: A group of 30 people have been trained as astronauts to go on the first mission to Mars. How many ways are there to select a crew of six people to go on this mission?

Solution: By Theorem 2, the number of possible crews is

$$C(30,6) = \frac{30!}{6!24!} = \frac{30\cdot 29\cdot 28\cdot 27\cdot 26\cdot 25}{6\cdot 5\cdot 4\cdot 3\cdot 2\cdot 1} = 593,775\,.$$

Scanned with CamScanner

Binomial Coefficients and Identities

Section 6.4

Scanned with CamScanner

Section Summary

- The Binomial Theorem
- Pascal's Identity and Triangle

Scanned with CamScanner

Binomial Theorem

Binomial Theorem: Let x and y be variables, and n a nonnegative integer. Then:

$$(x+y)^n = \sum_{j=0}^{n} \binom{n}{j} x^{n-j} y^j = \binom{n}{0} x^n + \binom{n}{1} x^{n-1} y + \cdots + \binom{n}{n-1} x y^{n-1} + \binom{n}{n} y^n.$$

Proof: We use combinatorial reasoning . The terms in the expansion of $(x + y)^n$ are of the form $x^{n-j}y^j$ for $j = 0,1,2,\ldots,n$. To form the term $x^{n-j}y^j$, it is necessary to choose $n-j$ xs from the n sums. Therefore, the coefficient of $x^{n-j}y^j$ is $\binom{n}{n-j}$ which equals $\binom{n}{j}$. ◀

Scanned with CamScanner

Powers of Binomial Expressions

Definition: A *binomial* expression is the sum of two terms, such as $x + y$. (More generally, these terms can be products of constants and variables.)

- We can use counting principles to find the coefficients in the expansion of $(x + y)^n$ where n is a positive integer.
- To illustrate this idea, we first look at the process of expanding $(x + y)^3$.
- $(x + y)\ (x + y)\ (x + y)$ expands into a sum of terms that are the product of a term from each of the three sums.
- Terms of the form x^3, x^2y, $x\,y^2$, y^3 arise. The question is what are the coefficients?
  - To obtain x^3, an x must be chosen from each of the sums. There is only one way to do this. So, the coefficient of x^3 is 1.
  - To obtain x^2y, an x must be chosen from two of the sums and a y from the other. There are $\binom{3}{2}$ ways to do this and so the coefficient of x^2y is 3.
  - To obtain xy^2, an x must be chosen from of the sums and a y from the other two . There are $\binom{3}{1}$ ways to do this and so the coefficient of xy^2 is 3.
  - To obtain y^3, a y must be chosen from each of the sums. There is only one way to do this. So, the coefficient of y^3 is 1.
- We have used a counting argument to show that $(x + y)^3 = x^3 + 3x^2y + 3x\,y^2 + y^3$.
- Next we present the binomial theorem gives the coefficients of the terms in the expansion of $(x + y)^n$.

Scanned with CamScanner

Using the Binomial Theorem

Example:

What is the expansion of $(x+y)^4$?

Solution: From the binomial theorem it follows that

$$(x+y)^4 = \sum_{j=0}^{4} \binom{4}{j} x^{4-j} y^j$$

$$= \binom{4}{0} x^4 + \binom{4}{1} x^3 y + \binom{4}{2} x^2 y^2 + \binom{4}{3} x y^3 + \binom{4}{4} y^4$$

$$= x^4 + 4x^3 y + 6x^2 y^2 + 4xy^3 + y^4.$$

Scanned with CamScanner

Using the Binomial Theorem

What is the coefficient of $x^{12}y^{13}$ in the expansion of $(x+y)^{25}$?

Solution: From the binomial theorem it follows that this coefficient is

$$\binom{25}{13} = \frac{25!}{13!\,12!} = 5{,}200{,}300.$$

Scanned with CamScanner

Using the Binomial Theorem

Example: What is the coefficient of $x^{12}y^{13}$ in the expansion of $(2x - 3y)^{25}$?

Solution: We view the expression as $(2x + (-3y))^{25}$. By the binomial theorem

$$(2x + (-3y))^{25} = \sum_{j=0}^{25} \binom{25}{j} (2x)^{25-j}(-3y)^j.$$

Consequently, the coefficient of $x^{12}y^{13}$ in the expansion is obtained when $j = 13$.

$$\binom{25}{13} 2^{12}(-3)^{13} = -\frac{25!}{13!12!}2^{12}3^{13}.$$

Scanned with CamScanner

A Useful Identity

Corollary 1: With $n \geq 0$, $$\sum_{k=0}^{n} \binom{n}{k} = 2^n.$$

Proof (*using binomial theorem*): With $x = 1$ and $y = 1$, from the binomial theorem we see that:

$$2^n = (1+1)^n = \sum_{k=0}^{n} \binom{n}{k} 1^k 1^{(n-k)} = \sum_{k=0}^{n} \binom{n}{k}. \quad \blacktriangleleft$$

Proof (*combinatorial*): Consider the subsets of a set with n elements. There are $\binom{n}{0}$ subsets with zero elements, $\binom{n}{1}$ with one element, $\binom{n}{2}$ with two elements, ..., and $\binom{n}{n}$ with n elements. Therefore the total is $$\sum_{k=0}^{n} \binom{n}{k}.$$

Since, we know that a set with n elements has 2^n subsets, we conclude:

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n. \quad \blacktriangleleft$$

Scanned with CamScanner

Blaise Pascal
(1623-1662)

Pascal's Identity

Pascal's Identity: If n and k are integers with $n \geq k \geq 0$, then

$$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}.$$

Proof (*combinatorial*): Let T be a set where $|T| = n + 1$, $a \in T$, and $S = T - \{a\}$. There are $\binom{n+1}{k}$ subsets of T containing k elements. Each of these subsets either:

- contains a with $k - 1$ other elements, or
- contains k elements of S and not a.

There are

- $\binom{n}{k-1}$ subsets of k elements that contain a, since there are $\binom{n}{k-1}$ subsets of $k - 1$ elements of S,
- $\binom{n}{k}$ subsets of k elements of T that do not contain a, because there are $\binom{n}{k}$ subsets of k elements of S.

Hence,

$$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}. \quad \blacktriangleleft$$

See Exercise 19 *for an algebraic proof.*

Scanned with CamScanner

Pascal's Triangle

The nth row in the triangle consists of the binomial coefficients $\binom{n}{k}$, $k = 0,1,\ldots,n$.

$$\binom{0}{0}$$

$$\binom{1}{0}\ \binom{1}{1}$$

$$\binom{2}{0}\ \binom{2}{1}\ \binom{2}{2}$$

$$\binom{3}{0}\ \binom{3}{1}\ \binom{3}{2}\ \binom{3}{3}$$

$$\binom{4}{0}\ \binom{4}{1}\ \binom{4}{2}\ \binom{4}{3}\ \binom{4}{4}$$

$$\binom{5}{0}\ \binom{5}{1}\ \binom{5}{2}\ \binom{5}{3}\ \binom{5}{4}\ \binom{5}{5}$$

$$\binom{6}{0}\ \binom{6}{1}\ \binom{6}{2}\ \binom{6}{3}\ \binom{6}{4}\ \binom{6}{5}\ \binom{6}{6}$$

$$\binom{7}{0}\ \binom{7}{1}\ \binom{7}{2}\ \binom{7}{3}\ \binom{7}{4}\ \binom{7}{5}\ \binom{7}{6}\ \binom{7}{7}$$

$$\binom{8}{0}\ \binom{8}{1}\ \binom{8}{2}\ \binom{8}{3}\ \binom{8}{4}\ \binom{8}{5}\ \binom{8}{6}\ \binom{8}{7}\ \binom{8}{8}$$

$$\cdots$$

(a)

By Pascal's identity:

$$\binom{6}{4} + \binom{6}{5} = \binom{7}{5}$$

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

1 8 28 56 70 56 28 8 1

$\cdots$

(b)

By Pascal's identity, adding two adjacent bionomial coefficients results is the binomial coefficient in the next row between these two coefficients.

Scanned with CamScanner

Graphs

Chapter 10

Scanned with CamScanner

Chapter Summary

- Graphs and Graph Models
- Graph Terminology and Special Types of Graphs
- Representing Graphs and Graph Isomorphism
- Connectivity
- Euler and Hamiltonian Graphs
- Shortest-Path Problems
- Planar Graphs
- Graph Coloring

Scanned with CamScanner

Graphs and Graph Models

Section 10.1

Scanned with CamScanner

Section Summary

- Introduction to Graphs
- Graph Taxonomy
- Graph Models

Scanned with CamScanner

Graphs

Definition: A *graph* $G = (V, E)$ consists of a nonempty set V of *vertices* (or *nodes*) and a set E of *edges*. Each edge has either one or two vertices associated with it, called its *endpoints*. An edge is said to *connect* its endpoints.

Example:

This is a graph with four vertices and five edges.

Remarks:

- The graphs we study here are unrelated to graphs of functions studied in Chapter 2.
- We have a lot of freedom when we draw a picture of a graph. All that matters is the connections made by the edges, not the particular geometry depicted. For example, the lengths of edges, whether edges cross, how vertices are depicted, and so on, do not matter
- A graph with an infinite vertex set is called an *infinite graph*. A graph with a finite vertex set is called a *finite graph*. We (following the text) restrict our attention to finite graphs.

Scanned with CamScanner

Some Terminology

- In a *simple graph* each edge connects two different vertices and no two edges connect the same pair of vertices.
- *Multigraphs* may have multiple edges connecting the same two vertices. When m different edges connect the vertices u and v, we say that $\{u,v\}$ is an edge of *multiplicity* m.
- An edge that connects a vertex to itself is called a *loop*.
- A *pseudograph* may include loops, as well as multiple edges connecting the same pair of vertices.

Example:
This pseudograph has both multiple edges and a loop.

Remark: There is no standard terminology for graph theory. So, it is crucial that you understand the terminology being used whenever you read material about graphs.

Scanned with CamScanner

Directed Graphs

Definition: An *directed graph* (or *digraph*) $G = (V, E)$ consists of a nonempty set V of *vertices* (or *nodes*) and a set E of *directed edges* (or *arcs*). Each edge is associated with an ordered pair of vertices. The directed edge associated with the ordered pair (u,v) is said to *start at u* and *end at v*.

Remark:

- Graphs where the end points of an edge are not ordered are said to be *undirected graphs*.

Scanned with CamScanner

Some Terminology (*continued*)

- A *simple directed graph* has no loops and no multiple edges.

Example:
This is a directed graph with three vertices and four edges.

- A *directed multigraph* may have multiple directed edges. When there are m directed edges from the vertex u to the vertex v, we say that (u,v) is an edge of *multiplicity* m.

Example:
In this directed multigraph the multiplicity of (a,b) is 1 and the multiplicity of (b,c) is 2.

Scanned with CamScanner

Graph Terminology: Summary

- To understand the structure of a graph and to build a graph model, we ask these questions:
  - Are the edges of the graph undirected or directed (or both)?
  - If the edges are undirected, are multiple edges present that connect the same pair of vertices? If the edges are directed, are multiple directed edges present?
  - Are loops present?

TABLE 1 Graph Terminology.

Type	*Edges*	*Multiple Edges Allowed?*	*Loops Allowed?*
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudograph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixed graph	Directed and undirected	Yes	Yes

Scanned with CamScanner

Other Applications of Graphs

- We will illustrate how graph theory can be used in models of:
  - Social networks
  - Communications networks
  - Information networks
  - Software design
  - Transportation networks
  - Biological networks
- It's a challenge to find a subject to which graph theory has not yet been applied. Can you find an area without applications of graph theory?

Scanned with CamScanner

Graph Models: Computer Networks

- When we build a graph model, we use the appropriate type of graph to capture the important features of the application.
- We illustrate this process using graph models of different types of computer networks. In all these graph models, the vertices represent data centers and the edges represent communication links.
- To model a computer network where we are only concerned whether two data centers are connected by a communications link, we use a simple graph. This is the appropriate type of graph when we only care whether two data centers are directly linked (and not how many links there may be) and all communications links work in both directions.

Scanned with CamScanner

Graph Models: Computer Networks (*continued*)

- To model a computer network where we care about the number of links between data centers, we use a multigraph.

Scanned with CamScanner

Graph Models: Computer Networks

- To model a computer network with diagnostic links at data centers, we use a pseudograph, as loops are needed.

Scanned with CamScanner

Graph Models: Computer Networks

- To model a network with multiple one-way links, we use a directed multigraph. Note that we could use a directed graph without multiple edges if we only care whether there is at least one link from a data center to another data center.

Scanned with CamScanner

Graph Models: Social Networks

- Graphs can be used to model social structures based on different kinds of relationships between people or groups.
- In a *social network*, vertices represent individuals or organizations and edges represent relationships between them.
- Useful graph models of social networks include:
  - *friendship graphs* - undirected graphs where two people are connected if they are friends (in the real world, on Facebook, or in a particular virtual world, and so on.)
  - *collaboration graphs* - undirected graphs where two people are connected if they collaborate in a specific way
  - *influence graphs* - directed graphs where there is an edge from one person to another if the first person can influence the second person

Scanned with CamScanner

Graph Models: Social Networks

Example: A friendship graph where two people are connected if they are Facebook friends.

Scanned with CamScanner

Graph Models: Social Networks

Example: An influence graph

Scanned with CamScanner

Applications to Information Networks

- Graphs can be used to model different types of networks that link different types of information.
- In a *web graph*, web pages are represented by vertices and links are represented by directed edges.
  - A web graph models the web at a particular time.
  - We will explain how the web graph is used by search engines in Section 11.4.
- In a *citation network*:
  - Research papers in a particular discipline are represented by vertices.
  - When a paper cites a second paper as a reference, there is an edge from the vertex representing this paper to the vertex representing the second paper.

Scanned with CamScanner

Transportation Graphs

- Graph models are extensively used in the study of transportation networks.
- Airline networks can be modeled using directed multigraphs where
  - airports are represented by vertices
  - each flight is represented by a directed edge from the vertex representing the departure airport to the vertex representing the destination airport
- Road networks can be modeled using graphs where
  - vertices represent intersections and edges represent roads.
  - undirected edges represent two-way roads and directed edges represent one-way roads.

Scanned with CamScanner

Software Design Applications

- Graph models are extensively used in software design. We will introduce two such models here; one representing the dependency between the modules of a software application and the other representing restrictions in the execution of statements in computer programs.
- When a top-down approach is used to design software, the system is divided into modules, each performing a specific task.
- We use a *module dependency graph* to represent the dependency between these modules. These dependencies need to be understood before coding can be done.

Scanned with CamScanner

Software Design Applications

- In a module dependency graph vertices represent software modules and there is an edge from one module to another if the second module depends on the first.

Example: The dependencies between the seven modules in the design of a web browser are represented by this module dependency graph.

Scanned with CamScanner

Software Design Applications

- We can use a directed graph called a *precedence graph* to represent which statements must have already been executed before we execute each statement.
  - Vertices represent statements in a computer program
  - There is a directed edge from a vertex to a second vertex if the second vertex cannot be executed before the first

Example: This precedence graph shows which statements must already have been executed before we can execute each of the six statements in the program.

S_1	$a := 0$
S_2	$b := 1$
S_3	$c := a + 1$
S_4	$d := b + a$
S_5	$e := d + 1$
S_6	$e := c + d$

Scanned with CamScanner

Biological Applications

- Graph models are used extensively in many areas of the biological science. We will describe two such models, one to ecology and the other to molecular biology.
- *Niche overlap graphs* model competition between species in an ecosystem
  - Vertices represent species and an edge connects two vertices when they represent species who compete for food resources.

Example: This is the niche overlap graph for a forest ecosystem with nine species.

Scanned with CamScanner

Biological Applications

- We can model the interaction of proteins in a cell using a *protein interaction network*.
- In a *protein interaction graph*, vertices represent proteins and vertices are connected by an edge if the proteins they represent interact.
- Protein interaction graphs can be huge and can contain more than 100,000 vertices, each representing a different protein, and more than 1,000,000 edges, each representing an interaction between proteins
- Protein interaction graphs are often split into smaller graphs, called *modules*, which represent the interactions between proteins involved in a particular function.

Scanned with CamScanner

Biological Applications

Example: This is a module of the protein interaction graph of proteins that degrade RNA in a human cell.

Scanned with CamScanner

Graph Terminology and Special Types of Graphs

Section 10.2

Scanned with CamScanner

Section Summary

- Basic Terminology
- Some Special Types of Graphs
- Bipartite Graphs
- Bipartite Graphs and Matchings
- Some Applications of Special Types of Graphs
- New Graphs from Old

Scanned with CamScanner

Basic Terminology

Definition 1. Two vertices u, v in an undirected graph G are called *adjacent* (or *neighbors*) in G if there is an edge e between u and v. Such an edge e is called *incident with* the vertices u and v and e is said to *connect* u and v.

Definition 2. The set of all neighbors of a vertex v of $G = (V, E)$, denoted by $N(v)$, is called the *neighborhood* of v. If A is a subset of V, we denote by $N(A)$ the set of all vertices in G that are adjacent to at least one vertex in A. So, $N(A) = \bigcup_{v \in A} N(v)$.

Definition 3. The *degree of a vertex in a undirected graph* is the number of edges incident with it, except that a loop at a vertex contributes two to the degree of that vertex. The degree of the vertex v is denoted by $\deg(v)$.

Scanned with CamScanner

Degrees and Neighborhoods of Vertices

Example: What are the degrees and neighborhoods of the vertices in the graphs G and H?

Solution:

G: $\deg(a) = 2$, $\deg(b) = \deg(c) = \deg(f) = 4$, $\deg(d) = 1$, $\deg(e) = 3$, $\deg(g) = 0$.

$N(a) = \{b, f\}$, $N(b) = \{a, c, e, f\}$, $N(c) = \{b, d, e, f\}$, $N(d) = \{c\}$,

$N(e) = \{b, c, f\}$, $N(f) = \{a, b, c, e\}$, $N(g) = \varnothing$.

H: $\deg(a) = 4$, $\deg(b) = \deg(e) = 6$, $\deg(c) = 1$, $\deg(d) = 5$.

$N(a) = \{b, d, e\}$, $N(b) = \{a, b, c, d, e\}$, $N(c) = \{b\}$, $N(d) = \{a, b, e\}$,

$N(e) = \{a, b, d\}$.

Scanned with CamScanner

Handshaking Theorem

Theorem: If G is any graph, then the sum of the degrees of all the vertices of G equals twice the number of edges of G.

Specifically, if the vertices of G are **v1**, **v2**, ..., **vn**, where n is a positive integer, then

the total degree of G = deg(v1) + deg(v2) + ... + deg(vn)
= 2 . (the number of edges of G)

PROOF:

- Each edge "e" of G connects its end points vi and vj. This edge, therefore contributes 1 to the degree of vi and 1 to the degree of vj.
- If "e" is a loop, then it is counted twice in computing the degree of the vertex on which it is incident.
- Accordingly, each edge of G contributes 2 to the total degree of G.

Thus, the total degree of G = 2. (the number of edges of G)

COROLLARY: The total degree of G is an even number

Scanned with CamScanner

Degree of Vertices

Theorem: An undirected graph has an even number of vertices of odd degree.

***Proof*:** Let V_1 be the vertices of even degree and V_2 be the vertices of odd degree in an undirected graph $G = (V, E)$ with m edges. Then

$$\text{even} \rightarrow 2m = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v).$$

must be even since $\deg(v)$ is even for each $v \in V_1$

This sum must be even because $2m$ is even and the sum of the degrees of the vertices of even degrees is also even. Because this is the sum of the degrees of all vertices of odd degree in the graph, there must be an even number of such vertices.

Scanned with CamScanner

Handshaking Theorem

We now give two examples illustrating the usefulness of the handshaking theorem.

Example: How many edges are there in a graph with 10 vertices of degree six?

Solution: Because the sum of the degrees of the vertices is $6 \cdot 10 = 60$, the handshaking theorem tells us that $2m = 60$. So the number of edges $m = 30$.

Example: If a graph has 5 vertices, can each vertex have degree 3?

Solution: This is not possible by the handshaking theorem, because the sum of the degrees of the vertices $3 \cdot 5 = 15$ is odd.

Scanned with CamScanner

Directed Graphs

Recall the definition of a directed graph.

Definition: An *directed graph* $G = (V, E)$ consists of V, a nonempty set of *vertices* (or *nodes*), and E, a set of *directed edges* or *arcs*. Each edge is an ordered pair of vertices. The directed edge (u,v) is said to start at u and end at v.

Definition: Let (u,v) be an edge in G. Then u is the *initial vertex* of this edge and is *adjacent to* v and v is the *terminal* (or *end*) *vertex* of this edge and is *adjacent from* u. The initial and terminal vertices of a loop are the same.

Scanned with CamScanner

Directed Graphs (*continued*)

Definition: The *in-degree of a vertex v*, denoted $deg^-(v)$, is the number of edges which terminate at v. The *out-degree of v*, denoted $deg^+(v)$, is the number of edges with v as their initial vertex. Note that a loop at a vertex contributes 1 to both the in-degree and the out-degree of the vertex.

Example: In the graph G we have

$\deg^-(a) = 2$, $\deg^-(b) = 2$, $\deg^-(c) = 3$, $\deg^-(d) = 2$, $\deg^-(e) = 3$, $\deg^-(f) = 0$.

$\deg^+(a) = 4$, $\deg^+(b) = 1$, $\deg^+(c) = 2$, $\deg^+(d) = 2$, $\deg^+(e) = 3$, $\deg^+(f) = 0$.

Scanned with CamScanner

Directed Graphs (*continued*)

Theorem 3: Let $G = (V, E)$ be a graph with directed edges. Then:

$$|E| = \sum_{v \in V} deg^{-}(v) = \sum_{v \in V} deg^{+}(v).$$

Proof: The first sum counts the number of outgoing edges over all vertices and the second sum counts the number of incoming edges over all vertices. It follows that both sums equal the number of edges in the graph.

Scanned with CamScanner

Special Types of Simple Graphs: Complete Graphs

A *complete graph on n vertices*, denoted by K_n, is the simple graph that contains exactly one edge between each pair of distinct vertices.

Scanned with CamScanner

Special Types of Simple Graphs: Cycles and Wheels

A *cycle* C_n for $n \geq 3$ consists of n vertices $v_1, v_2, \cdots, v_n$, and edges $\{v_1, v_2\}, \{v_2, v_3\}, \cdots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$.

A *wheel* W_n is obtained by adding an additional vertex to a cycle C_n for $n \geq 3$ and connecting this new vertex to each of the n vertices in C_n by new edges.

Scanned with CamScanner

Special Types of Graphs and Computer Network Architecture

Various special graphs play an important role in the design of computer networks.

(a) (b) (c)

- Some local area networks use a *star topology*, which is a complete bipartite graph $K_{1,n}$, as shown in (a). All devices are connected to a central control device.
- Other local networks are based on a *ring topology*, where each device is connected to exactly two others using C_n, as illustrated in (b). Messages may be sent around the ring.
- Others, as illustrated in (c), use a W_n – based topology, combining the features of a star topology and a ring topology.

Scanned with CamScanner

Special Types of Simple Graphs: *n*-Cubes

An *n-dimensional hypercube*, or *n-cube*, $\boldsymbol{Q_n}$, is a graph with 2^n vertices representing all bit strings of length n, where there is an edge between two vertices that differ in exactly one bit position.

Scanned with CamScanner

Special Types of Graphs and Computer Network Architecture

- Various special graphs also play a role in parallel processing where processors need to be interconnected as one processor may need the output generated by another.
- The n-dimensional hypercube, or n-cube, Qn, is a common way to connect processors in parallel, e.g., Intel Hypercube.
- Another common method is the mesh network, illustrated here for 16 processors.

Scanned with CamScanner

Bipartite Graphs

Definition: A simple graph G is bipartite if V can be partitioned into two disjoint subsets V_1 and V_2 such that every edge connects a vertex in V_1 and a vertex in V_2. In other words, there are no edges which connect two vertices in V_1 or in V_2.

- It is not hard to show that an equivalent definition of a bipartite graph is a graph where it is possible to color the vertices red or blue so that no two adjacent vertices are the same color.

Scanned with CamScanner

Bipartite Graphs (*continued*)

Example: Show that C_3 is not bipartite.

Solution: If we divide the vertex set of C_3 into two nonempty sets, one of the two must contain two vertices. But in C_3 every vertex is connected to every other vertex. Therefore, the two vertices in the same partition are connected. Hence, C_3 is not bipartite.

Scanned with CamScanner

Bipartite Graphs (*continued*)

Example: Show that C_6 is bipartite.

Solution: We can partition the vertex set into

$V_1 = \{v_1, v_3, v_5\}$ and $V_2 = \{v_2, v_4, v_6\}$ so that every edge of C_6 connects a vertex in V_1 and V_2 .

Scanned with CamScanner

Bipartite Graphs

G is bipartite

H is not bipartite since if we color a red, then the adjacent vertices f and b must both be blue.

Scanned with CamScanner

Bipartite Graphs and Matchings

- Bipartite graphs are used to model applications that involve matching the elements of one set to elements in another, for example:
- *Job assignments* - vertices represent the jobs and the employees, edges link employees with those jobs they have been trained to do. A common goal is to match jobs to employees so that the most jobs are done.

Scanned with CamScanner

Bipartite Graphs and Matchings

Marriage/dating - vertices represent men & women and edges link a man & woman if they are acceptable to each other as partners.

Example:

Match.com an online dating website receive applications from 5 boys (**Andy, Bobby, Colin, Dan and Elliott**) and 5 girls (**Rita, Sarah, Tracey, Uma and Victoria**). After filling in their details the match.com algorithms work out the following:

- Andy appears compatible with all 5 girls.

- Bobby appears compatible with Uma.

- Colin appears compatible with Sarah.

- Dan appears compatible with Tracey and Victoria.

- Elliott does not appear compatible with any of the girls.

Draw a bipartite graph to model this situation.

Scanned with CamScanner

Bipartite Graphs and Matchings

Scanned with CamScanner

Complete Bipartite Graphs

Definition: A *complete bipartite graph* $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets V_1 of size m and V_2 of size n such that there is an edge from every vertex in V_1 to every vertex in V_2.

Example: We display four complete bipartite graphs here.

$K_{2,3}$

$K_{3,3}$

$K_{3,5}$

$K_{2,6}$

Scanned with CamScanner

New Graphs from Old

Definition: A *subgraph of a graph* $G = (V,E)$ is a graph (W,F), where $W \subset V$ and $F \subset E$. A subgraph H of G is a proper subgraph of G if $H \neq G$.

Example: Here we show K_5 and one of its subgraphs.

Definition: Let $G = (V, E)$ be a simple graph. The *subgraph induced* by a subset W of the vertex set V is the graph (W,F), where the edge set F contains an edge in E if and only if both endpoints are in W.

Example: Here we show K_5 and the subgraph induced by $W = \{a,b,c,e\}$.

Scanned with CamScanner

New Graphs from Old (*continued*)

Definition: The *union* of two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. The union of G_1 and G_2 is denoted by $G_1 \cup G_2$.

Example:

(a) (b)

Scanned with CamScanner

Representations of Graphs

Section 10.3

Scanned with CamScanner

Section Summary

- Adjacency Lists
- Adjacency Matrices
- Incidence Matrices

Scanned with CamScanner

Representing Graphs: Adjacency Lists

Definition: An *adjacency list* can be used to represent a graph with no multiple edges by specifying the vertices that are adjacent to each vertex of the graph.

Example:

TABLE 1 An Adjacency List for a Simple Graph.

Vertex	*Adjacent Vertices*
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

Scanned with CamScanner

Representing Graphs: Adjacency Lists

Example:

TABLE 2 An Adjacency List for a Directed Graph.

Initial Vertex	*Terminal Vertices*
a	b, c, d, e
b	b, d
c	a, c, e
d	
e	b, c, d

Scanned with CamScanner

Representation of Graphs: Adjacency Matrices

Definition: Suppose that $G = (V, E)$ is a simple graph where $|V| = n$. Arbitrarily list the vertices of G as $v_1, v_2, \ldots, v_n$. The *adjacency matrix* $\mathbf{A}_G$ of G, with respect to the listing of vertices, is the $n \times n$ zero-one matrix with 1 as its (i, j)th entry when v_i and v_j are adjacent, and 0 as its (i, j)th entry when they are not adjacent.

- In other words, if the graphs adjacency matrix is $\mathbf{A}_G = [a_{ij}]$, then

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

Scanned with CamScanner

Adjacency Matrices (*continued*)

Example:

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

The ordering of vertices is a, b, c, d.

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

The ordering of vertices is a, b, c, d.

When a graph is sparse, that is, it has few edges relatively to the total number of possible edges, it is much more efficient to represent the graph using an adjacency list than an adjacency matrix. But for a dense graph, which includes a high percentage of possible edges, an adjacency matrix is preferable.

Note: The adjacency matrix of a simple graph is symmetric, i.e., $a_{ij} = a_{ji}$
Also, since there are no loops, each diagonal entry a_{ij} for $i = 1, 2, 3, \ldots, n$, is 0.

Scanned with CamScanner

Adjacency Matrices (*continued*)

- Adjacency matrices can also be used to represent graphs with loops and multiple edges.
- A loop at the vertex vi is represented by a 1 at the (i, j)th position of the matrix.
- When multiple edges connect the same pair of vertices vi and vj, (or if multiple loops are present at the same vertex), the (i, j)th entry equals the number of edges connecting the pair of vertices.
- Example: We give the adjacency matrix of the pseudograph shown here using the ordering of vertices a, b, c, d.

$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}$$

Scanned with CamScanner

Adjacency Matrices (*continued*)

- Adjacency matrices can also be used to represent directed graphs. The matrix for a directed graph $G = (V, E)$ has a 1 in its (i, j)th position if there is an edge from v_i to v_j, where $v_1, v_2, \ldots v_n$ is a list of the vertices.
  - In other words, if the graphs adjacency matrix is $\mathbf{A}_G = [a_{ij}]$, then

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

  - The adjacency matrix for a directed graph does not have to be symmetric, because there may not be an edge from v_i to v_j, when there is an edge from v_j to v_i.
  - To represent directed multigraphs, the value of a_{ij} is the number of edges connecting v_i to v_j.

Scanned with CamScanner

Representation of Graphs: Incidence Matrices

Definition: Let $G = (V, E)$ be an undirected graph with vertices where $v_1, v_2, \ldots v_n$ and edges $e_1, e_2, \ldots e_m$. The incidence matrix with respect to the ordering of V and E is the $n \times m$ matrix $\mathbf{M} = [m_{ij}]$, where

$$m_{ij} = \begin{cases} 1 & \text{when edge } e_j \text{ is incident with } v_i, \\ 0 & \text{otherwise.} \end{cases}$$

Scanned with CamScanner

Incidence Matrices (*continued*)

Example: Simple Graph and Incidence Matrix

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

The rows going from top to bottom represent v_1 through v_5 and the columns going from left to right represent e_1 through e_6.

Example: Pseudograph and Incidence Matrix

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

The rows going from top to bottom represent v_1 through v_5 and the columns going from left to right represent e_1 through e_8.

Scanned with CamScanner

Connectivity

Section 10.4

Scanned with CamScanner

Section Summary

- Paths
- Connectedness in Undirected Graphs
- Connectedness in Directed Graphs
- Counting Paths between Vertices

Scanned with CamScanner

Paths

Informal Definition: A *path* is a sequence of edges that begins at a vertex of a graph and travels from vertex to vertex along edges of the graph. As the path travels along its edges, it visits the vertices along this path, that is, the endpoints of these.

Applications: Numerous problems can be modeled with paths formed by traveling along edges of graphs such as:

- determining whether a message can be sent between two computers.
- efficiently planning routes for mail delivery.

Scanned with CamScanner

Paths

Definition: Let n be a nonnegative integer and G an undirected graph. A *path* of *length n* from u to v in G is a sequence of n edges $e_1, \ldots, e_n$ of G for which there exists a sequence $x_0 = u, x_1, \ldots, x_{n-1}, x_n = v$ of vertices such that e_i has, for $i = 1, \ldots, n$, the endpoints x_{i-1} and x_i.

- When the graph is simple, we denote this path by its vertex sequence $x_0, x_1, \ldots, x_n$(since listing the vertices uniquely determines the path).
- The path is a *circuit* if it begins and ends at the same vertex ($u = v$) and has length greater than zero.
- The path or circuit is said to *pass through* the vertices $x_1, x_2, \ldots, x_{n-1}$ and *traverse* the edges $e_1, \ldots, e_n$.
- A path or circuit is *simple* if it does not contain the same edge more than once.

Scanned with CamScanner

Paths (*continued*)

Example: In the simple graph here:

- *a*, *d*, *c*, *f*, *e* is a simple path of length 4.
- *d*, *e*, *c*, *a* is not a path because *e* is not connected to *c*.
- *b*, *c*, *f*, *e*, *b* is a circuit of length 4.
- *a*, *b*, *e*, *d*, *a*, *b* is a path of length 5, but it is not a simple path.

Scanned with CamScanner

Connectedness in Undirected Graphs

Definition: An undirected graph is called *connected* if there is a path between every pair of vertices. An undirected graph that is not *connected* is called *disconnected*. We say that we *disconnect* a graph when we remove vertices or edges, or both, to produce a disconnected subgraph.

Example: G_1 is connected because there is a path between any pair of its vertices, as can be easily seen. However G_2 is not connected because there is no path between vertices a and f, for example.

G_1

G_2

Scanned with CamScanner

Euler and Hamiltonian Graphs

Section 10.5

Scanned with CamScanner

Section Summary

- Euler Paths and Circuits
- Hamilton Paths and Circuits
- Applications of Hamilton Circuits

Scanned with CamScanner

Euler Paths and Circuits

Leonard Euler
(1707-1783)

- The town of Königsberg, Prussia (now Kalingrad, Russia) was divided into four sections by the branches of the Pregel river. In the 18th century seven bridges connected these regions.
- People wondered whether whether it was possible to follow a path that crosses each bridge exactly once and returns to the starting point.
- The Swiss mathematician Leonard Euler proved that no such path exists. This result is often considered to be the first theorem ever proved in graph theory.

The 7 Bridges of Königsberg

Multigraph Model of the Bridges of Königsberg

Scanned with CamScanner

Euler Paths and Circuits

Definition: An *Euler circuit* in a graph G is a simple circuit containing every edge of G. An *Euler path* in G is a simple path containing every edge of G.

Example: Which of the undirected graphs G_1, G_2, and G_3 has a Euler circuit? Of those that do not, which has an Euler path?

Solution: The graph G_1 has an Euler circuit (e.g., a, e, c, d, e, b, a). But, as can easily be verified by inspection, neither G_2 nor G_3 has an Euler circuit. Note that G_3 has an Euler path (e.g., a, c, d, e, b, d, a, b), but there is no Euler path in G_2, which can be verified by inspection.

Scanned with CamScanner

Necessary Conditions for Euler Circuits and Paths

- An Euler circuit begins with a vertex a and continues with an edge incident with a, say $\{a, b\}$. The edge $\{a, b\}$ contributes one to $\deg(a)$.
- Each time the circuit passes through a vertex it contributes two to the vertex's degree.
- Finally, the circuit terminates where it started, contributing one to $\deg(a)$. Therefore $\deg(a)$ must be even.
- We conclude that the degree of every other vertex must also be even.
- By the same reasoning, we see that the initial vertex and the final vertex of an Euler path have odd degree, while every other vertex has even degree. So, a graph with an Euler path has exactly two vertices of odd degree.
- In the next slide we will show that these necessary conditions are also sufficient conditions.

Scanned with CamScanner

Sufficient Conditions for Euler Circuits and Paths

Suppose that G is a connected multigraph with ≥ 2 vertices, all of even degree. Let $x_0 = a$ be a vertex of even degree. Choose an edge $\{x_0, x_1\}$ incident with a and proceed to build a simple path $\{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_n\}$ by adding edges one by one until another edge can not be added.

We illustrate this idea in the graph G here. We begin at a and choose the edges $\{a, f\}$, $\{f, c\}$, $\{c, b\}$, and $\{b, a\}$ in succession.

- The path begins at a with an edge of the form $\{a, x\}$; we show that it must terminate at a with an edge of the form $\{y, a\}$. Since each vertex has an even degree, there must be an even number of edges incident with this vertex. Hence, every time we enter a vertex other than a, we can leave it. Therefore, the path can only end at a.
- If all of the edges have been used, an Euler circuit has been constructed. Otherwise, consider the subgraph H obtained from G by deleting the edges already used.

In the example H consists of the vertices c, d, e.

Scanned with CamScanner

Sufficient Conditions for Euler Circuits and Paths

- Because G is connected, H must have at least one vertex in common with the circuit that has been deleted.

In the example, the vertex is *c*.

- Every vertex in H must have even degree because all the vertices in *G* have even degree and for each vertex, pairs of edges incident with this vertex have been deleted. Beginning with the shared vertex construct a path ending in the same vertex (as was done before). Then splice this new circuit into the original circuit.

In the example, we end up with the circuit *a, f, c, d, e, c, b, a*.

- Continue this process until all edges have been used. This produces an Euler circuit. Since every edge is included and no edge is included more than once.
- Similar reasoning can be used to show that a graph with exactly two vertices of odd degree must have an Euler path connecting these two vertices of odd degree

Scanned with CamScanner

Algorithm for Constructing an Euler Circuits

In our proof we developed this algorithms for constructing a Euler circuit in a graph with no vertices of odd degree.

procedure *Euler*(G: connected multigraph with all vertices of even degree)
circuit := a circuit in G beginning at an arbitrarily chosen vertex with edges successively added to form a path that returns to this vertex.
H := G with the edges of this circuit removed
while H has edges
subciruit := a circuit in H beginning at a vertex in H that also is an endpoint of an edge in circuit.
H := H with edges of *subciruit* and all isolated vertices removed
circuit := *circuit* with *subcircuit* inserted at the appropriate vertex.
return *circuit*{*circuit* is an Euler circuit}

Scanned with CamScanner

Necessary and Sufficient Conditions for Euler Circuits and Paths (*continued*)

Theorem: A connected multigraph with at least two vertices has an Euler circuit if and only if each of its vertices has an even degree and it has an Euler path if and only if it has exactly two vertices of odd degree.

Example: Two of the vertices in the multigraph model of the Königsberg bridge problem have odd degree. Hence, there is no Euler circuit in this multigraph and it is impossible to start at a given point, cross each bridge exactly once, and return to the starting point.

Scanned with CamScanner

Euler Circuits and Paths

Example:

G_1 contains exactly two vertices of odd degree (b and d). Hence it has an Euler path, e.g., d, a, b, c, d, b.

G_2 has exactly two vertices of odd degree (b and d). Hence it has an Euler path, e.g., $b, a, g, f, e, d, c, g, b, c, f, d$.

G_3 has six vertices of odd degree. Hence, it does not have an Euler path.

Scanned with CamScanner

Applications of Euler Paths and Circuits

- Euler paths and circuits can be used to solve many practical problems such as finding a path or circuit that traverses each
  - street in a neighborhood,
  - road in a transportation network,
  - connection in a utility grid,
  - link in a communications network.
- Other applications are found in the
  - layout of circuits,
  - network multicasting,
  - molecular biology, where Euler paths are used in the sequencing of DNA.

Scanned with CamScanner

Hamilton Paths and Circuits

William Rowan Hamilton (1805- 1865)

- Euler paths and circuits contained every edge only once.
  Now we look at paths and circuits that contain every vertex exactly once.
- William Hamilton invented the *Icosian puzzle* in 1857. It consisted of a wooden dodecahedron (with 12 regular pentagons as faces), illustrated in (a), with a peg at each vertex, labeled with the names of different cities. String was used to used to plot a circuit visiting 20 cities exactly once
- The graph form of the puzzle is given in (b).

(a)

(b)

- The solution (a Hamilton circuit) is given here.

Scanned with CamScanner

Hamilton Paths and Circuits

Definition: A simple path in a graph G that passes through every vertex exactly once is called a *Hamilton path*, and a simple circuit in a graph G that passes through every vertex exactly once is called a *Hamilton circuit*.

That is, a simple path $x_0, x_1, \ldots, x_{n-1}, x_n$ in the graph $G = (V, E)$ is called a Hamilton path if $V = \{x_0, x_1, \ldots, x_{n-1}, x_n\}$ and $x_i \neq x_j$ for $0 \leq i < j \leq n$, and the simple circuit $x_0, x_1, \ldots, x_{n-1}, x_n, x_0$ (with $n > 0$) is a Hamilton circuit if $x_0, x_1, \ldots, x_{n-1}, x_n$ is a Hamilton path.

Scanned with CamScanner

Hamilton Paths and Circuits

Example: Which of these simple graphs has a Hamilton circuit or, if not, a Hamilton path?

Solution:

G_1 does not have a Hamilton circuit (Why?), but does have a Hamilton path : a, b ,e , c, d.

G_2 has a Hamilton circuit: a, b, c, d, e, a.

G_3 has a Hamilton circuit: a,b,e,d,c,a

Scanned with CamScanner

Necessary Conditions for Hamilton Circuits

Gabriel Andrew Dirac
(1925-1984)

- Unlike for an Euler circuit, no simple necessary and sufficient conditions are known for the existence of a Hamiton circuit.
- However, there are some useful necessary conditions. We describe two of these now.

Dirac's Theorem: If G is a simple graph with $n \geq 3$ vertices such that the degree of every vertex in G is $\geq n/2$, then G has a Hamilton circuit.

Ore's Theorem: If G is a simple graph with $n \geq 3$ vertices such that $\deg(u) + \deg(v) \geq n$ for every pair of nonadjacent vertices, then G has a Hamilton circuit.

Øysten Ore
(1899-1968)

Scanned with CamScanner

Applications of Hamilton Paths and Circuits

- Applications that ask for a path or a circuit that visits each intersection of a city, each place pipelines intersect in a utility grid, or each node in a communications network exactly once, can be solved by finding a Hamilton path in the appropriate graph.
- The famous *traveling salesperson problem* (*TSP*) asks for the shortest route a traveling salesperson should take to visit a set of cities. This problem reduces to finding a Hamilton circuit such that the total sum of the weights of its edges is as small as possible.
- A family of binary codes, known as *Gray codes*, which minimize the effect of transmission errors, correspond to Hamilton circuits in the n-cube Q_n.

Scanned with CamScanner

ISOMORPHISM OF GRAPHS

Section 10.3

Scanned with CamScanner

Isomorphism of Graphs

Definition: The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic* if there is a one-to-one and onto function f from V_1 to V_2 with the property that a and b are adjacent in G_1 if and only if $f(a)$ and $f(b)$ are adjacent in G_2 , for all a and b in V_1 . Such a function f is called an *isomorphism*. Two simple graphs that are not isomorphic are called *nonisomorphic*.

Scanned with CamScanner

Isomorphism of Graphs

- It is difficult to determine whether two simple graphs are isomorphic using brute force because there are $n!$ possible one-to-one correspondences between the vertex sets of two simple graphs with n vertices.
- The best algorithms for determining weather two graphs are isomorphic have exponential worst case complexity in terms of the number of vertices of the graphs.
- Sometimes it is not hard to show that two graphs are not isomorphic. We can do so by finding a property, preserved by isomorphism, that only one of the two graphs has. Such a property is called *graph invariant*.
- There are many different useful graph invariants that can be used to distinguish nonisomorphic graphs, such as the number of vertices, number of edges, and degree sequence (list of the degrees of the vertices in nonincreasing order). We will encounter others in later sections of this chapter.

Scanned with CamScanner

ISOMORPHIC INVARIANT

- A property P is called an isomorphic invariant if, and only if, given any graphs G and G',
- If G has property P and G' is isomorphic to G, then G' has property P.

THEOREM OF ISOMORPHIC INVARIANT

Each of the following properties is an invariant for graph isomorphism, where n, m and k are all non-negative integers, if the graph:

1. has n vertices.
2. has m edges.
3. has a vertex of degree k.
4. has m vertices of degree k.
5. has a circuit of length k.
6. has a simple circuit of length k.
7. has m simple circuits of length k.
8. is connected.
9. has an Euler circuit.
10. has a Hamiltonian circuit.

Scanned with CamScanner

Isomorphism of Graphs (*cont.*)

Example: Show that the graphs $G = (V, E)$ and $H = (W, F)$ are isomorphic.

Solution: The function f with $f(u_1) = v_1, f(u_2) = v_4, f(u_3) = v_3$, and $f(u_4)$ correspondence between V and W. Note that adjacent vertices in G are u_1 and u_2, u_1 and u_3, u_2 and u_4, and u_3 and u_4. Each of the pairs $f(u_1) = v_1$ and $f(u_2) = v_4$, $f(u_1) = v_1$ and $f(u_3) = v_3$, $f(u_2) = v_4$ and $f(u_4) = v_2$, and $f(u_3) = v_3$ and $f(u_4) = v_2$ consists of two adjacent vertices in H.

Scanned with CamScanner

Isomorphism of Graphs

Example: Show that the graphs $G = (V, E)$ and $G' = (W, F)$ are isomorphic.

Solution: The function f with

f(v1) = v1, f(v2) = v3, f(v3) = v5, f(v4) = v2 and f(v5) = v4 is a one-to-one correspondence between V and W.

Note that adjacent vertices in G are v1 and v2, v2 and v3, v3 and v4, v4 and v5 and v5 and v1.

Each of the pairs f(v1) = v1 and f(v2) = v3, f(v2) = v3 and f(v3) = v5 , f(v3) = v5 and f(v4) = v2 , f(v4) = v2 and f(v5) = v4 and f(v5) = v4 and f(v1) = v1 consists of two adjacent vertices in H.

Scanned with CamScanner

Isomorphism of Graphs

EXAMPLE: Determine whether the graph G and G' given below are isomorphic.

SOLUTION:

As both the graphs have the same number of vertices. But the graph G has 7 edges and the graph G' has only 6 edges. Therefore the two graphs are not isomorphic.

Note: As the edges of both the graphs G and G' are not same then how the one-one correspondence is possible ,that the reason the graphs G and G' are not isomorphic.

Scanned with CamScanner

Isomorphism of Graphs

EXAMPLE: Determine whether the graph G and G' given below are isomorphic.

SOLUTION:

Both the graphs have 5 vertices and 7 edges. The vertex q of G' has degree 5. However G does not have any vertex of degree 5 (so one-one correspondence is not possible). Hence, the two graphs are not isomorphic.

Scanned with CamScanner

Isomorphism of Graphs

EXAMPLE: Determine whether the graph G and G' given below are isomorphic.

SOLUTION:

Clearly the vertices of both the graphs G and G' have the same degree (i.e "2") and having the same number of vertices and edges but isomorphism is not possible. As the graph G' is a connected graph but the graph G is not connected due to have two components (eca and bdf). Therefore the two graphs are non isomorphic.

Scanned with CamScanner

Isomorphism of Graphs

EXAMPLE: Determine whether the graph G and G' given below are isomorphic.

SOLUTION:

Clearly G has six vertices, G' also has six vertices. And the graph G has two simple circuits of length 3; one is abca and the other is defd. But G' does not have any simple circuit of length 3(as one simple circuit in G' is uxwv of length 4).Therefore the two graphs are non-isomorphic.

Note: A simple circuit is a circuit that does not have any other repeated vertex except the first and last.

Scanned with CamScanner

Isomorphism of Graphs

EXAMPLE: Determine whether the graph G and G' given below are isomorphic.

SOLUTION:

Both the graph G and G' have 8 vertices and 12 edges and both are also called regular graph(as each vertex has degree 3).The graph G has two simple circuits of length 5; abcfea(i.e starts and ends at a) and cdhgfc(i.e starts and ends at c). But G' does not have any simple circuit of length 5 (it has simple circuit tyxut,vwxuv of length 4 etc). Therefore the two graphs are non-isomorphic.

Scanned with CamScanner

Isomorphism of Graphs

EXAMPLE: Determine whether the given graph G and H are isomorphic.

Solution:

Solution: The function f with

f(a) = u, f(b) = v, f(c) = y, f(d) = x, f(e) = w and f(f) = z is a one-to-one correspondence between G and H.

Note that adjacent vertices in G are a and b, b and c, c and d, c and f, d and e, e and f and f and a.

Each of the pairs f(a) = u and f(b) = v, f(b) = v and f(c) = y , f(c) = y and f(d) = x , f(c) = y and f(f) = z , f(d) = x and f(e) = w, f(e) = w and f(f) = z and f(f) = z and f(a) = u consists of two adjacent vertices in H.

Scanned with CamScanner

Isomorphism of Graphs (*cont.*)

Example: Determine whether these two graphs are isomorphic.

G

H

Solution: Both graphs have eight vertices and ten edges.
They also both have four vertices of degree two and four of degree three.

However, G and H are not isomorphic. Note that since $deg(a) = 2$ in G, a must correspond to t, u, x, or y in H, because these are the vertices of degree 2. But each of these vertices is adjacent to another vertex of degree two in H, which is not true for a in G.

Alternatively, note that the subgraphs of G and H made up of vertices of degree three and the edges connecting them must be isomorphic.
But the subgraphs, as shown at the right, are not isomorphic.

Scanned with CamScanner

Isomorphism of Graphs

Example: Determine whether these two graphs are isomorphic.

Solution: Both graphs have six vertices and seven edges.

They also both have four vertices of degree two and two of degree three. The subgraphs of G and H consisting of all the vertices of degree two and the edges connecting them are isomorphic. So, it is reasonable to try to find an isomorphism f.

We define an injection f from the vertices of G to the vertices of H that preserves the degree of vertices. We will determine whether it is an isomorphism.

The function f with $f(u_1) = v_6$, $f(u_2) = v_3$, $f(u_3) = v_4$, and $f(u_4) = v_5$, $f(u_5) = v_1$, and $f(u_6) = v_2$ is a one-to-one correspondence between G and H. Showing that this correspondence preserves edges is straightforward, so we will omit the details here. Because f is an isomorphism, it follows that G and H are isomorphic graphs.

Scanned with CamScanner

Algorithms for Graph Isomorphism

- The best algorithms known for determining whether two graphs are isomorphic have exponential worst-case time complexity (in the number of vertices of the graphs).
- However, there are algorithms with linear average-case time complexity.
- You can use a public domain program called NAUTY to determine in less than a second whether two graphs with as many as 100 vertices are isomoprhic.
- Graph isomorphism is a problem of special interest because it is one of a few NP problems not known to be either tractable or NP-complete (see Section 3.3).

Scanned with CamScanner

Applications of Graph Isomorphism

- The question whether graphs are isomorphic plays an important role in applications of graph theory. For example,
  - chemists use molecular graphs to model chemical compounds. Vertices represent atoms and edges represent chemical bonds. When a new compound is synthesized, a database of molecular graphs is checked to determine whether the graph representing the new compound is isomorphic to the graph of a compound that this already known.
  - Electronic circuits are modeled as graphs in which the vertices represent components and the edges represent connections between them. Graph isomorphism is the basis for
    - the verification that a particular layout of a circuit corresponds to the design's original schematics.
    - determining whether a chip from one vendor includes the intellectual property of another vendor.

Scanned with CamScanner

Shortest Paths

Scanned with CamScanner

Weighted Graphs

Graphs that have a number assigned to each edge are called *weighted graphs*.

Scanned with CamScanner

Weighted Graphs

MILES

Scanned with CamScanner

Weighted Graphs

FARES

Scanned with CamScanner

Weighted Graphs

FLIGHT
TIMES

Scanned with CamScanner

Weighted Graphs

- A weighted graph is a graph in which each edge (u, v) has a weight w(u, v). Each weight is a real number.
- Weights can represent distance, cost, time, capacity, etc.
- The length of a path in a weighted graph is the sum of the weights on the edges.
- Dijkstra's Algorithm finds the shortest path between two vertices.

Scanned with CamScanner

Dijkstra's Algorithm

- Dijkstra's algorithm is used in problems relating to finding the shortest path.
- Each node is given a temporary label denoting the length of the shortest path *from* the start node *so far*.
- This label is replaced if another shorter route is found.
- Once it is certain that no other shorter paths can be found, the temporary label becomes a permanent label.
- Eventually all the nodes have permanent labels.
- At this point the shortest path is found by retracing the path backwards.

Scanned with CamScanner

Dijkstra's algorithm: example

Step	N'	D(v)	D(w)	D(x)	D(y)	D(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					

notes:

- construct shortest path tree by tracing predecessor nodes
- ties can exist (can be broken arbitrarily)

Scanned with CamScanner

Dijkstra's algorithm: another example

Step	N'	D(v)	D(w)	D(x)	D(y)	D(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

Scanned with CamScanner

FIGURE 3 **A Weighted Simple Graph.**

What is the length of a shortest path between a and z in the weighted graph shown in Figure 3?

Scanned with CamScanner

Problem: shortest path from a to z

Scanned with CamScanner

The Traveling Salesman Problem

- The **traveling salesman problem** is one of the classical problems in computer science.
- A traveling salesman wants to visit a number of cities and then return to his starting point. Of course he wants to save time and energy, so he wants to determine the **shortest cycle** for his trip.
- We can represent the cities and the distances between them by a weighted, complete, undirected graph.
- The problem then is to find the **shortest cycle (of minimum total weight that visits each vertex exactly one)**.
- Finding the shortest cycle is different than Dijkstra's shortest path.
  It is much harder too, no polynomial time algorithm exists!

Scanned with CamScanner

The Traveling Salesman Problem

- Importance:
  - Variety of scheduling application can be solved as a traveling salesmen problem.
  - Examples:
    - Ordering drill position on a drill press.
    - School bus routing.
  - The problem has theoretical importance because it represents a class of difficult problems known as NP-hard problems.

Scanned with CamScanner

Travelling Salesman problem

FIGURE 5 **The Graph Showing the Distances between Five Cities.**

Scanned with CamScanner

Travelling Salesman problem

Route	*Total Distance (miles)*
Detroit–Toledo–Grand Rapids–Saginaw–Kalamazoo–Detroit	610
Detroit–Toledo–Grand Rapids–Kalamazoo–Saginaw–Detroit	516
Detroit–Toledo–Kalamazoo–Saginaw–Grand Rapids–Detroit	588
Detroit–Toledo–Kalamazoo–Grand Rapids–Saginaw–Detroit	458
Detroit–Toledo–Saginaw–Kalamazoo–Grand Rapids–Detroit	540
Detroit–Toledo–Saginaw–Grand Rapids–Kalamazoo–Detroit	504
Detroit–Saginaw–Toledo–Grand Rapids–Kalamazoo–Detroit	598
Detroit–Saginaw–Toledo–Kalamazoo–Grand Rapids–Detroit	576
Detroit–Saginaw–Kalamazoo–Toledo–Grand Rapids–Detroit	682
Detroit–Saginaw–Grand Rapids–Toledo–Kalamazoo–Detroit	646
Detroit–Grand Rapids–Saginaw–Toledo–Kalamazoo–Detroit	670
Detroit–Grand Rapids–Toledo–Saginaw–Kalamazoo–Detroit	728

Scanned with CamScanner

Planar Graphs

Scanned with CamScanner

The House-and-Utilities Problem

© The McGraw-Hill Companies, Inc. all rights reserved.

Scanned with CamScanner

Planar Graphs

- Consider the previous slide. Is it possible to join the three houses to the three utilities in such a way that none of the connections cross?

Scanned with CamScanner

Planar Graphs

- Phrased another way, this question is equivalent to: Given the complete bipartite graph $K_{3,3}$, can $K_{3,3}$ be drawn in the plane so that no two of its edges cross?

$K_{3,3}$

Scanned with CamScanner

Planar Graphs

- A graph is called *planar* if it can be drawn in the plane without any edges crossing.
- A crossing of edges is the intersection of the lines or arcs representing them at a point other than their common endpoint.
- Such a <u>drawing</u> is called a *planar representation* of the graph.

Scanned with CamScanner

Example

A graph may be planar even if it is usually drawn with crossings, since it may be possible to draw it in another way without crossings.

Scanned with CamScanner

Example

A graph may be planar even if it represents a 3-dimensional object.

© The McGraw-Hill Companies, Inc. all rights reserved.

© The McGraw-Hill Companies, Inc. all rights reserved.

Scanned with CamScanner

Planar Graphs

- We can prove that a particular graph is planar by showing how it can be drawn without any crossings.
- However, not all graphs are planar.
- It may be difficult to show that a graph is nonplanar. We would have to show that there is *no way* to draw the graph without any edges crossing.

Scanned with CamScanner

Regions

- Euler showed that all planar representations of a graph split the plane into the same number of *regions*, including an unbounded region.

Scanned with CamScanner

Regions

- In any planar representation of $K_{3,3}$, vertex v_1 must be connected to both v_4 and v_5, and v_2 also must be connected to both v_4 and v_5.

Scanned with CamScanner

Regions

- The four edges $\{v_1, v_4\}$, $\{v_4, v_2\}$, $\{v_2, v_5\}$, $\{v_5, v_1\}$ form a closed curve that splits the plane into two regions, R_1 and R_2.

Scanned with CamScanner

Regions

- Next, we note that v_3 must be in either R_1 or R_2.
- Assume v_3 is in R_2. Then the edges $\{v_3, v_4\}$ and $\{v_4, v_5\}$ separate R_2 into two subregions, R_{21} and R_{22}.

Scanned with CamScanner

Regions

- Now there is no way to place vertex v_6 without forcing a crossing:
  - If v_6 is in R_1 then $\{v_6, v_3\}$ must cross an edge
  - If v_6 is in R_{21} then $\{v_6, v_2\}$ must cross an edge
  - If v_6 is in R_{22} then $\{v_6, v_1\}$ must cross an edge

Scanned with CamScanner

Regions

- Alternatively, assume v_3 is in R_1. Then the edges $\{v_3, v_4\}$ and $\{v_4, v_5\}$ separate R_1 into two subregions, R_{11} and R_{12}.

Scanned with CamScanner

Regions

- Now there is no way to place vertex v_6 without forcing a crossing:
  - If v_6 is in R_2 then $\{v_6, v_3\}$ must cross an edge
  - If v_6 is in R_{11} then $\{v_6, v_2\}$ must cross an edge
  - If v_6 is in R_{12} then $\{v_6, v_1\}$ must cross an edge

Scanned with CamScanner

Planar Graphs

- Consequently, the graph $K_{3,3}$ must be nonplanar.

$K_{3,3}$

Scanned with CamScanner

Regions

- Euler devised a formula for expressing the relationship between the number of vertices, edges, and regions of a planar graph.
- These *may* help us determine if a graph can be planar or not.

Scanned with CamScanner

Euler's Formula

- Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G. Then $r = e - v + 2$.

# of edges, $e = 6$
# of vertices, $v = 4$
# of regions, $r = e - v + 2 = 4$

Scanned with CamScanner

Euler's Formula (Cont.)

- Corollary 1: If G is a connected planar simple graph with e edges and v vertices where $v \geq 3$, then $e \leq 3v - 6$.
- Is K_5 planar?

K_5

Scanned with CamScanner

Euler's Formula (Cont.)

- K_5 has 5 vertices and 10 edges.
- We see that $v \geq 3$.
- So, if K_5 is planar, it must be true that $e \leq 3v - 6$.
- $3v - 6 = 3*5 - 6 = 15 - 6 = 9$.
- So e must be ≤ 9.
- But $e = 10$.
- So, K_5 is nonplanar.

K_5

Scanned with CamScanner

Euler's Formula (Cont.)

- Corollary 2: If G is a connected planar simple graph, then G must have a vertex of degree not exceeding 5.

If G has one or two vertices, it is true;
thus, we assume that G has at least three vertices.

If the degree of each vertex were at least 6, then by Handshaking Theorem,
$2e \geq 6v$, i.e., $e \geq 3v$,

but this contradicts the inequality from
Corollary 1: $e \leq 3v - 6$.

$$2e = \sum_{v \in V} \deg(v)$$

Scanned with CamScanner

Euler's Formula (Cont.)

- Corollary 3: If a connected planar simple graph has e edges and v vertices with $v \geq 3$ and no circuits of length 3, then $e \leq 2v - 4$.
- Is $K_{3,3}$ planar?

Scanned with CamScanner

Euler's Formula (Cont.)

- $K_{3,3}$ has 6 vertices and 9 edges.
- Obviously, v ≥ 3 and there are no circuits of length 3.
- If $K_{3,3}$ were planar, then $e \leq 2v - 4$ would have to be true.
- $2v - 4 = 2*6 - 4 = 8$
- So e must be ≤ 8.
- But $e = 9$.
- So $K_{3,3}$ is nonplanar.

$K_{3,3}$

Scanned with CamScanner

Trees

Chapter 11

Scanned with CamScanner

Chapter Summary

- Introduction to Trees
- Applications of Trees
- Tree Traversal
- Spanning Trees
- Minimum Spanning

Scanned with CamScanner

Introduction to Trees

Section 11.1

Scanned with CamScanner

Section Summary

- Introduction to Trees
- Rooted Trees
- Trees as Models
- Properties of Trees

Scanned with CamScanner

Trees

Definition: A *tree* is a connected undirected graph with no simple circuits.

Definition: An undirected graph is a tree if and only if there is a unique simple path between any two of its vertices. A tree cannot contain multiple edges or loops.

Definition: An undirected graph is a tree if and only if there is a unique simple path between any two of its vertices.

Scanned with CamScanner

Trees

Example: Which of these graphs are trees?

Solution: G_1 and G_2 are trees - both are connected and have no simple circuits. G_3 is not a tree because e, b, a, d, e is a simple circuit,. G_4 is not a tree because it is not connected.

Scanned with CamScanner

FOREST

Definition: A *forest* is a graph that has no simple circuit, but is not connected. Each of the connected components in a forest is a tree.

This is one graph with three connected components.

Scanned with CamScanner

Applications of Trees

Section 11.2

Scanned with CamScanner

Trees as Models

Arthur Cayley
(1821-1895)

- Trees are used as models in computer science, chemistry, geology, botany, psychology, and many other areas.

- Trees were introduced by the mathematician Cayley in 1857 in his work counting the number of isomers of saturated hydrocarbons. The two isomers of butane are:

Butane

Isobutane

Scanned with CamScanner

Trees as Models

- The organization of a computer file system into directories, subdirectories, and files is naturally represented as a tree.

Scanned with CamScanner

Trees as Models

- Trees are used to represent the structure of organizations.

Scanned with CamScanner

Applications of Trees

- **Game Trees**

Trees can be used to analyze certain types of games such as tic-tac-toe, nim, checkers, and chess.

Scanned with CamScanner

Game Tree for Tic-Tac-Toe

Scanned with CamScanner

Universal Address Systems

FIGURE 1 **The Universal Address System of an Ordered Rooted Tree.**

Scanned with CamScanner

Prefix code

Definition: A code that has the property that the code of a character is never a prefix of the code of another character.

- A prefix code can be represented using a binary tree, where the characters are the labels of the leaves in the tree.
- The edges of the tree are labeled so that an edge leading to a left child is assigned a 0 and an edge leading to a right child is assigned a 1.
- The bit string used to encode a character is the sequence of labels of the edges in the unique path from the root to the leaf that has this character as its label.
- For instance, the tree in Figure 5 represents the encoding of *e* by 0, *a* by 10, *t* by 110, *n* by 1110, and *s* by 1111.

FIGURE 5 A Binary Tree with a Prefix Code.

Scanned with CamScanner

Decision Trees

Definition: A rooted tree where each vertex represents a possible outcome of a decision and the leaves represent the possible solutions of a problem.

- Rooted trees can be used to model problems in which a series of decisions leads to a solution.
- The possible solutions of the problem correspond to the paths to the leaves of this rooted tree.

Scanned with CamScanner

Example : A decision tree that orders the elements of the list *a*, *b*, *c*.

A Decision Tree for Sorting Three Distinct Elements.

Scanned with CamScanner

Rooted Trees

Definition: A *rooted tree* is a tree in which one vertex has been designated as the *root* and every edge is directed away from the root.

- An unrooted tree is converted into different rooted trees when different vertices are chosen as the root.

Scanned with CamScanner

Rooted Tree Terminology

- Terminology for rooted trees is a mix from botany and genealogy (such as this family tree of the Bernoulli family of mathematicians).

Scanned with CamScanner

Rooted Tree Terminology

- If v is a vertex of a rooted tree other than the root, the *parent* of v is the unique vertex u such that there is a directed edge from u to v. When u is a parent of v, v is called a *child* of u. Vertices with the same parent are called *siblings*.

Scanned with CamScanner

Rooted Tree Terminology

- The *ancestors* of a vertex are the vertices in the path from the root to this vertex, excluding the vertex itself and including the root.
- The *descendants* of a vertex *v* are those vertices that have *v* as an ancestor. The subtree rooted at u includes all the descendants of u, and all edges that connect between them.

The whole part forms a path from root to w

The whole part is the subtree rooted at u

Scanned with CamScanner

Rooted Tree Terminology

- A vertex of a rooted tree with no children is called a *leaf*. Vertices that have children are called *internal vertices*.

Scanned with CamScanner

Terminology for Rooted Trees

Example: In the rooted tree T (with root a):

(i) Find the parent of c, the children of g, the siblings of h, the ancestors of e, and the descendants of b.

Solution:

(i) The parent of c is b. The children of g are h, i, and j. The siblings of h are i and j. The ancestors of e are c, b, and a. The descendants of b are c, d, and e.

Scanned with CamScanner

Terminology for Rooted Trees

Example: In the rooted tree T (with root a):

(i) Find all internal vertices and all leaves.

Solution:

(i) The internal vertices are a, b, c, g, h, and j. The leaves are d, e, f, i, k, l, and m.

Scanned with CamScanner

Terminology for Rooted Trees

(i) What is the subtree rooted at g?

Solution:

(i) We display the subtree rooted at g.

Scanned with CamScanner

Level of vertices and height of trees

- When working with trees, we often want to have rooted trees where the sub trees at each vertex contain paths of approximately the same length.
- To make this idea precise we need some definitions:
  - The *level* of a vertex *v* in a rooted tree is the length of the unique path from the root to this vertex.
  - The *height* of a rooted tree is the maximum of the levels of the vertices.

Scanned with CamScanner

Level of vertices and height of trees

Example:

(*i*) Find the level of each vertex in the tree to the right.

(*ii*) What is the height of the tree?

Solution:

(*i*) The root *a* is at level 0.

Vertices *b*, *j*, and *k* are at level 1.

Vertices *c*, *e*, *f*, and *l* are at level 2.

Vertices *d*, *g*, *i*, *m*, and *n* are at level 3.

Vertex *h* is at level 4.

(*ii*) The height is 4, since 4 is the largest level of any vertex.

Scanned with CamScanner

m-ary Rooted Trees

Definition: A rooted tree is called an *m-ary tree* if every internal vertex has no more than *m* children. The tree is called a *full m-ary tree* if every internal vertex has exactly *m* children. An *m*-ary tree with $m = 2$ is called a *binary* tree.

Example: Are the following rooted trees full *m*-ary trees for some positive integer *m*?

Scanned with CamScanner

Solution:

- T_1 is a full binary tree because each of its internal vertices has two children.
- T_2 is a full 3-ary tree because each of its internal vertices has three children.
- In T_3 each internal vertex has five children, so T_3 is a full 5-ary tree.
- T_4 is not a full m-ary tree for any m because some of its internal vertices have two children and others have three children.

Scanned with CamScanner

Balanced *m*-Ary Trees

Definition: A rooted m-ary tree of height h is *balanced* if all leaves are at levels h or $h-1$.

Example: Which of the rooted trees shown below is balanced?

Solution: T_1 and T_3 are balanced, but T_2 is not because it has leaves at levels 2, 3, and 4.

Scanned with CamScanner

Ordered Rooted Trees

Definition: An *ordered rooted tree* is a rooted tree where the children of each internal vertex are ordered.

- We draw ordered rooted trees so that the children of each internal vertex are shown in order from left to right.

Scanned with CamScanner

Binary Trees

Definition: A *binary tree* is an ordered rooted where each internal vertex has at most two children. If an internal vertex of a binary tree has two children, the first is called the *left child* and the second the *right child*. The tree rooted at the left child of a vertex is called the *left subtree* of this vertex, and the tree rooted at the right child of a vertex is called the *right subtree* of this vertex.

Scanned with CamScanner

Example:

Consider the binary tree T.

(*i*) What are the left and right children of d?

(*ii*) What are the left and right sub trees of c?

Solution:

(*i*) The left child of d is f and the right child is g.

(*ii*) The left and right subtrees of c are displayed in (b) and (c).

Scanned with CamScanner

Properties of Trees

- A tree with n vertices has $n - 1$ edges.
- A full m-ary tree with i internal vertices has $n = mi + 1$ vertices.
- A full m-ary tree with:

  (i) *n vertices has* $i = (n - 1)/m$ *internal vertices and* $l = [(m - 1)n + 1]/m$ *leaves,*

  (ii) i internal vertices has $n = mi + 1$ vertices and $l = (m - 1)i + 1$ leaves,

  (iii) l leaves has $n = (ml - 1)/(m - 1)$ vertices and $i = (l - 1)/(m - 1)$ internal vertices.
- There are at most m^h leaves in an m-ary tree of height h.

Scanned with CamScanner

Binary Search Tree

Definition: A binary tree in which the vertices are labeled with items so that a label of a vertex is greater than the labels of all vertices in the left subtree of this vertex and is less than the labels of all vertices in the right subtree of this vertex.

- Searching for items in a list is one of the most important tasks that arises in computer science.
- Our primary goal is to implement a searching algorithm that finds items efficiently when the items are totally ordered. This can be accomplished through the use of a binary search tree.

Scanned with CamScanner

Example : Form a binary search tree for the words mathematics, physics, geography, zoology, meteorology, geology, psychology, and chemistry (using alphabetical order).

Scanned with CamScanner

Tree Traversal

Section 11.3

Scanned with CamScanner

Tree Traversal

- Procedures for systematically visiting every vertex of an ordered tree are called *traversals*.
- The three most commonly used *traversals* are *preorder traversal*, *inorder traversal*, and *postorder traversal*.

Scanned with CamScanner

Preorder Traversal

Definition: Let T be an ordered rooted tree with root r. If T consists only of r, then r is the *preorder traversal* of T. Otherwise, suppose that $T_1, T_2, \ldots, T_n$ are the subtrees of r from left to right in T. The preorder traversal begins by visiting r, and continues by traversing T_1 in preorder, then T_2 in preorder, and so on, until T_n is traversed in preorder.

Scanned with CamScanner

Preorder Traversal (*continued*)

procedure *preorder*
(T: ordered rooted tree)
r := root of T
list r
for each child c of r from left to right
  $T(c)$:= subtree with c as root
  preorder($T(c)$)

Scanned with CamScanner

Inorder Traversal

Definition: Let T be an ordered rooted tree with root r. If T consists only of r, then r is the *inorder traversal* of T. Otherwise, suppose that $T_1, T_2, \ldots, T_n$ are the subtrees of r from left to right in T. The inorder traversal begins by traversing T_1 in inorder, then visiting r, and continues by traversing T_2 in inorder, and so on, until T_n is traversed in inorder.

Scanned with CamScanner

Inorder Traversal (*continued*)

procedure
inorder (T: ordered rooted tree)
r := root of T
if r is a leaf **then** list r
else
  l := first child of r from left to right
  $T(l)$:= subtree with l as its root
  inorder($T(l)$)
  list(r)
  for each child c of r from left to right
    $T(c)$:= subtree with c as root
    inorder($T(c)$)

Scanned with CamScanner

Postorder Traversal

Definition: Let T be an ordered rooted tree with root r. If T consists only of r, then r is the *postorder traversal* of T. Otherwise, suppose that $T_1, T_2, \ldots, T_n$ are the subtrees of r from left to right in T. The postorder traversal begins by traversing T_1 in postorder, then T_2 in postorder, and so on, after T_n is traversed in postorder, r is visited.

Scanned with CamScanner

Post order Traversal (*continued*)

procedure
postordered (*T*: ordered rooted tree)
r := root of *T*
for each child *c* of *r* from left to right
  T(*c*) := subtree with *c* as root
  postorder(*T*(*c*))
list *r*

Scanned with CamScanner

Expression Trees

- Complex expressions can be represented using ordered rooted trees.
- Consider the expression $((x + y) \uparrow 2\ ) + ((x - 4)/3)$.
- A binary tree for the expression can be built from the bottom up, as is illustrated here.

Scanned with CamScanner

Infix Notation

- An inorder traversal of the tree representing an expression produces the original expression when parentheses are included except for unary operations, which now immediately follow their operands.
- We illustrate why parentheses are needed with an example that displays three trees all yield the same infix representation.

Rooted Trees Representing $(x + y)/(x + 3)$, $(x + (y/x)) + 3$, and $x + (y/(x + 3))$.

Scanned with CamScanner

Jan Łukasiewicz
(1878-1956)

Prefix Notation

- When we traverse the rooted tree representation of an expression in preorder, we obtain the *prefix* form of the expression. Expressions in prefix form are said to be in *Polish notation*, named after the Polish logician Jan Łukasiewicz.
- Operators precede their operands in the prefix form of an expression. Parentheses are not needed as the representation is unambiguous.
- The prefix form of $((x + y) \uparrow 2\ ) + ((x - 4)/3)$
  is $+ \uparrow + x\ y\ 2\ /\ - x\ 4\ 3$.
- Prefix expressions are evaluated by working from right to left. When we encounter an operator, we perform the corresponding operation with the two operations to the right.

Scanned with CamScanner

Prefix Notation

- **Example**: We show the steps used to evaluate a particular prefix expression:

$$+ \quad - \quad * \quad 2 \quad 3 \quad 5 \quad / \quad \underbrace{\uparrow \quad 2 \quad 3}_{2 \uparrow 3 = 8} \quad 4$$

$$+ \quad - \quad * \quad 2 \quad 3 \quad 5 \quad \underbrace{/ \quad 8 \quad 4}_{8 / 4 = 2}$$

$$+ \quad - \quad \underbrace{* \quad 2 \quad 3}_{2 * 3 = 6} \quad 5 \quad 2$$

$$+ \quad \underbrace{- \quad 6 \quad 5}_{6 - 5 = 1} \quad 2$$

$$\underbrace{+ \quad 1 \quad 2}_{1 + 2 = 3}$$

Value of expression: 3

Scanned with CamScanner

Postfix Notation

- We obtain the *postfix form* of an expression by traversing its binary trees in postorder. Expressions written in postfix form are said to be in *reverse Polish notation*.
- Parentheses are not needed as the postfix form is unambiguous.
- $x\ y + 2 \uparrow x\ 4 - 3\ / +$ is the postfix form of $((x + y) \uparrow 2\ ) + ((x - 4)/3)$.
- A binary operator follows its two operands. So, to evaluate an expression one works from left to right, carrying out an operation represented by an operator on its preceding operands.

Scanned with CamScanner

Postfix Notation

- **Example**: We show the steps used to evaluate a particular postfix expression.

$$7 \quad \underbrace{2 \quad 3 \quad *}_{2 * 3 = 6} \quad - \quad 4 \quad \uparrow \quad 9 \quad 3 \quad / \quad +$$

$$\underbrace{7 \quad 6 \quad -}_{7 - 6 = 1} \quad 4 \quad \uparrow \quad 9 \quad 3 \quad / \quad +$$

$$\underbrace{1 \quad 4 \quad \uparrow}_{1^4 = 1} \quad 9 \quad 3 \quad / \quad +$$

$$1 \quad \underbrace{9 \quad 3 \quad /}_{9 / 3 = 3} \quad +$$

$$\underbrace{1 \quad 3 \quad +}_{1 + 3 = 4}$$

Value of expression: 4

Scanned with CamScanner

Spanning Trees

Section 11.4

Scanned with CamScanner

Section Summary

- Spanning Trees
- Prim's Algorithm
- Kruskal Algorithm

Scanned with CamScanner

Spanning Trees

Definition: Let G be a simple graph. A spanning tree of G is a subgraph of G that is a tree containing every vertex of G.

Example: Find the spanning tree of the simple graph:

Scanned with CamScanner

Spanning Trees

Solution: The graph is connected, but is not a tree because it contains simple circuits. Remove the edge $\{a, e\}$. Now one simple circuit is gone, but the remaining subgraph still has a simple circuit. Remove the edge $\{e, f\}$ and then the edge $\{c, g\}$ to produce a simple graph with no simple circuits. It is a spanning tree, because it contains every vertex of the original graph.

Scanned with CamScanner

Minimum Spanning

Section 11.5

Scanned with CamScanner

Minimum spanning tree

- A *minimum spanning tree* in a connected weighted graph is a spanning tree that has the smallest possible sum of weights of its edges.
- **Example:** A company plans to build a communications network connecting its five computer centers. Any pair of these centers can be linked with a leased telephone line. Which links should be made to ensure that there is a path between any two computer centers so that the total cost of the network is minimized?

Scanned with CamScanner

Minimum spanning tree

- **Solution:** We can model this problem using the weighted graph shown in Figure 1, where vertices represent computer centers, edges represent possible leased lines, and the weights on edges are the monthly lease rates of the lines represented by the edges. We can solve this problem by finding a spanning tree so that the sum of the weights of the edges of the tree is minimized. Such a spanning tree is called a **minimum spanning tree**.

FIGURE 1 A Weighted Graph Showing Monthly Lease Costs for Lines in a Computer Network.

Scanned with CamScanner

PRIM'S ALGORITHM

ALGORITHM 1 Prim's Algorithm.

procedure *Prim*(G: weighted connected undirected graph with n vertices)
T := a minimum-weight edge
for i := 1 **to** $n - 2$
  e := an edge of minimum weight incident to a vertex in T and not forming a simple circuit in T if added to T
  T := T with e added
return T {T is a minimum spanning tree of G}

Scanned with CamScanner

Minimal spanning tree (MST)

Example: Use Prims algorithm to find a minimal spanning tree for the graph below. Indicate the order in which edges are added to form the tree.

Order of adding the edges:

{a , b}, {b , c}, {c , d}, {d , e}, {e , f}, {f , g}

MST COST = 23

Scanned with CamScanner

KRUSKAL'S ALGORITHM

ALGORITHM 2 Kruskal's Algorithm.

procedure *Kruskal*(G: weighted connected undirected graph with n vertices)
T := empty graph
for i := 1 **to** $n - 1$
  e := any edge in G with smallest weight that does not form a simple circuit
    when added to T
  T := T with e added
return T {T is a minimum spanning tree of G}

Scanned with CamScanner

Minimal spanning tree (MST)

Example: Use Kruskal's algorithm to find a minimal spanning tree for the graph below. Indicate the order in which edges are added to form the tree.

Order of adding the edges:
{a, b}, {e, f}, {e, d}, {c, d}, {g, f}, {b, c}

Scanned with CamScanner