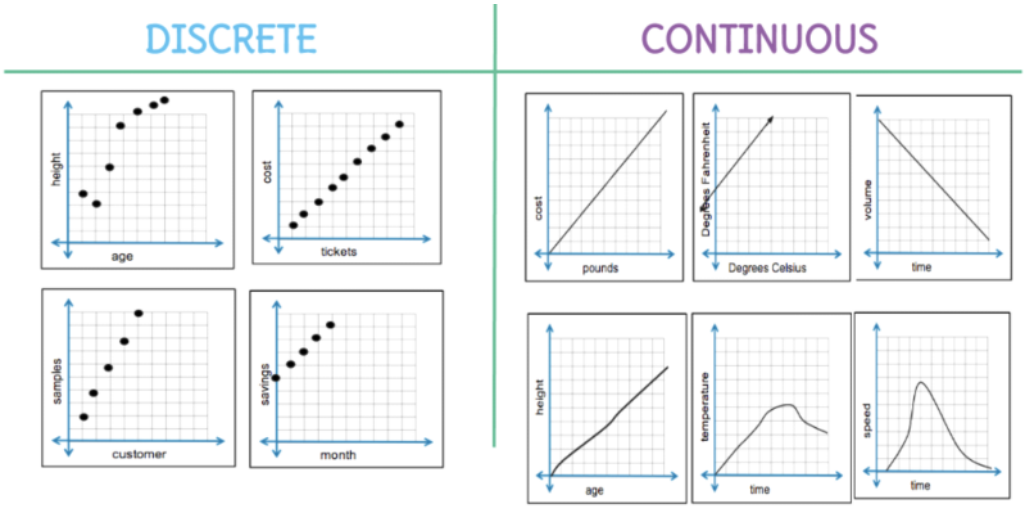
0. Introductory Lecture

Wednesday, January 24, 2018 8:56 AM

What is Discrete Mathematics?

- Study of discrete (as opposed to continuous) objects
- Calculus is continuous



- Example of discrete objects
  - Integers
  - Steps taken by a computer program
  - Distinct paths to travel from point A to point B on a map along a road network
  - Ways to pick a wining set of numbers in a lottery

Kinds of Problems Solved Using Discrete Mathematics

- Number of valid passwords
- Number of valid websites
- Probability of winning a lottery
- Link between two computers in a network
- Identify spam e-mails
- Shortest path
- Prove there are infinitely many prime numbers
- Numbers of steps need to do a sorting
- Prove the correctness of algorithms

Goals of a Course in Discrete Mathematics

- Mathematical Reasoning
- Combinatorial Analysis
- Discrete Structures

1.1 Propositional Logic

Wednesday, January 24, 2018 9:11 AM

Propositions

- Definition
  - A proposition is a declarative sentence that is either true or false.
- Examples of propositions
  - The Moon is made of green cheese.
  - Paris is the capital of Europe.
  - Toronto is the capital of Canada.
  - $1 + 0 = 1$
  - $0 + 0 = 2$
- Examples that are not propositions
  - Sit down!
  - What time is it?
  - $x + 1 = 2$
  - $x + y = z$

Constructing Propositions

- Propositional Variables: $p, q, r, s, \ldots$
- The proposition that is always true is denoted by T
- The proposition that is always false is denoted by F.

Compound Propositions

- Definition
  - Propositions constructed from logical connectives and other propositions
- Negation ¬
  - The negation of a proposition p is denoted by $\neg p$
  - Truth table

p	$\neg p$
T	F
F	T

  - Example
    - If p denotes "The earth is round."
    - Then $\neg p$ denotes "It is not the case that the earth is round,"
    - Or more simply "The earth is not round."
- Conjunction ∧

- The conjunction of propositions p and q is denoted by $p \wedge q$
- Truth Table

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

- Example
  - If p denotes "I am at home." and q denotes "It is raining."
  - Then $p \wedge q$ denotes "I am at home and it is raining."

- Disjunction $\vee$
  - The disjunction of propositions p and q is denoted by $p \vee q$
  - Truth Table

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

  - Example
    - If p denotes "I am at home." and q denotes "It is raining."
    - Then $p \vee q$ denotes "I am at home or it is raining."

- Inclusive Or vs Exclusive Or
  - "Inclusive Or"
    - In the sentence "Students who have taken CS202 or Math120 may take this class," we assume that students need to have taken one of the prerequisites, but may have taken both.
    - This is the meaning of disjunction.
    - For $p \vee q$ to be true, either one or both of p and q must be true.
  - "Exclusive Or"
    - When reading the sentence "Soup or salad comes with this entrée," we do not expect to be able to get both soup and salad.
    - This is the meaning of Exclusive Or (XOR).
    - In $p \oplus q$, one of p and q must be true, but not both.
    - The truth table for $\oplus$ is:

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

- Implication $\rightarrow$
  - If p and q are propositions, then $p \rightarrow q$ is a conditional statement or implication which is read as "if p, then q "
  - Truth Table

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

  - Example
    - If p denotes "I am at home." and q denotes "It is raining."
    - Then $p \rightarrow q$ denotes "If I am at home then it is raining."
  - In $p \rightarrow q$, p is the hypothesis (antecedent or premise) and q is the conclusion (or consequence).
- Biconditional $\leftrightarrow$
  - If p and q are propositions, then we can form the biconditional proposition $p \leftrightarrow q$, read as "p if and only if q."
  - Truth Table

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

  - 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."
- Example

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

Converse, Contrapositive, and Inverse

- From $p \rightarrow q$ we can form new conditional statements .
  - $q \rightarrow p$ is the converse of $p \rightarrow q$
  - $\neg q \rightarrow \neg p$ is the contrapositive of $p \rightarrow q$
  - $\neg p \rightarrow \neg q$ is the inverse of $p \rightarrow q$
- Example
  - "If it is raining, then I will not go to town."
  - p: "It is raining"

    - q: "I am going to town"
    - Sufficient Condition
        - It raining is a sufficient condition for my not going to town.
    - Necessary Condition
        - My not going to town is a necessary condition for it raining.
    - 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.
- Truth Table

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

Truth Table for Compound Propositions

- Construction of a truth table:
    - Rows
        - Need a row for every possible combination of values for the atomic propositions.
    - Columns
        - Need a column for the compound proposition (usually at far right)
        - Need a column for the truth value of each expression that occurs in the compound proposition as it is built up.
        - This includes the atomic propositions
- Precedence of Logical Operators

Operator	Precedence
¬	1
∧	2
∨	3
→	4
↔	5

- Example: $p \vee q \rightarrow \neg r$

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

T	F	T	T	F	F
F	T	T	T	F	F
F	F	T	F	F	T
T	T	F	T	T	T
T	F	F	T	T	T
F	T	F	T	T	T
F	F	F	F	T	T

1.2 Applications of Propositional Logic

Friday, January 26, 2018 9:10 AM

Translating English Sentences

- Steps to convert an English sentence to a statement in propositional logic
  - Identify atomic propositions and represent using propositional variables.
  - Determine appropriate logical connectives
- Example: "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."
  - $p \vee q \rightarrow \neg r$
- Example: "You can get an extra piece of pie if you have completed your homework or if you are extremely hungry"
  - p: You have completed your homework
  - q: You are extremely hungry
  - r: You can get an extra piece of pie
  - $p \vee q \rightarrow r$

System Specifications

- System and Software engineers take requirements in English and express them in a precise specification language based on logic.
- Example: "The automated reply cannot be sent when the file system is full"
  - p: "The automated reply can be sent"
  - q: "The file system is full."
  - $q \rightarrow \neg\, p$

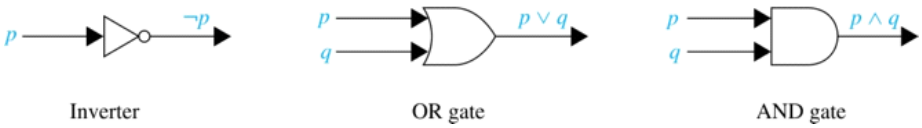
Logic Puzzles

- An island has two kinds of inhabitants, knights, who always tell the truth, and knaves, who always lie.
- You go to the island and meet A and B.
  - A says "B is a knight."
  - B says "The two of us are of opposite types."
- What are the types of A and B?
  - Let p and q be the statements that A is a knight and B is a knight, respectively.
  - So, then $\neg p$ represents the proposition that A is a knave and $\neg q$ that B is a knave.
  - If A is a knight, then p is true. Since knights tell the truth, q must also be true. Then $(p \wedge \neg q) \vee (\neg p \wedge q)$ would have to be true, but it is not. So, A is not a knight and therefore $\neg p$ must be true.

- If A is a knave, then B must not be a knight since knaves always lie. So, then both $\neg p$ and $\neg q$ hold since both are knaves.

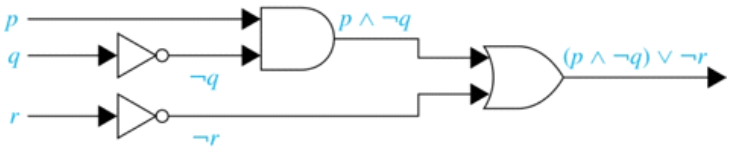
Logic Circuits

- Electronic circuits; each input/output signal can be viewed as a 0 or 1.
  - 0 represents False
  - 1 represents True
- Complicated circuits are constructed from three basic circuits called gates.



Inverter OR gate AND gate

  - The inverter (NOT gate) takes an input bit and produces the negation of that bit.
  - The OR gate takes two input bits and produces the value equivalent to the disjunction of the two bits.
  - The AND gate takes two input bits and produces the value equivalent to the conjunction of the two bits.
- More complicated digital circuits can be constructed by combining these basic circuits to produce the desired output given the input signals by building a circuit for each piece of the output expression and then combining them.



1.3 Propositional Equivalences

Friday, January 26, 2018 9:29 AM

Tautologies, Contradictions, and Contingencies

- Tautology
  - A tautology is a proposition which is always true.
  - Example: $p \vee \neg p$
- Contradiction
  - A contradiction is a proposition which is always false.
  - Example: $p \wedge \neg p$
- Contingency
  - A contingency is a proposition which is neither a tautology nor a contradiction
  - Example: p

Logically Equivalent

- Two compound propositions p and q are logically equivalent if $p \leftrightarrow q$ is a tautology.
- We write this as $p \Leftrightarrow q$ or as $p \equiv q$ where p and q are compound propositions.
- Two compound propositions p and q are equivalent if and only if the columns in a truth table giving their truth values agree.
- Example

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

De Morgan's Laws

- $\neg(p \wedge q) \equiv \neg p \vee \neg q$
- $\neg(p \vee q) \equiv \neg p \wedge \neg q$
- Truth Table

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

Key Logical Equivalences

- Identity Laws

- $p \wedge T \equiv p$
- $p \vee F \equiv p$

- Domination Laws
  - $p \vee T \equiv T$
  - $p \wedge F \equiv F$
- Idempotent Laws
  - $p \vee p \equiv p$
  - $p \wedge p \equiv p$
- Double Negation Law
  - $\neg(\neg p) \equiv p$
- Negation Laws
  - $p \vee \neg p \equiv T$
  - $p \wedge \neg p \equiv F$
- Commutative Laws
  - $p \wedge q \equiv q \wedge p$
  - $p \vee q = q \vee p$
- Associative Laws
  - $(p \vee q) \vee r = p \vee (q \vee r)$
  - $(p \wedge q) \wedge r = p \wedge (q \wedge r)$
- Distributive Laws
  - $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
  - $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
- Absorption Laws
  - $p \vee (p \wedge q) \equiv p$
  - $p \wedge (p \vee q) \equiv p$
- Logical Equivalences Involving Conditional Statements
  - $p \to q \equiv \neg p \vee q$
  - $p \to q \equiv \neg q \to \neg p$
  - $p \vee q \equiv \neg p \to q$
  - $p \wedge q \equiv \neg(p \to \neg q)$
  - $\neg(p \to q) \equiv p \wedge \neg q$
  - $(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$
  - $(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$
  - $(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$
  - $(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$
- Logical Equivalences Involving Biconditional Statements

- $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
- $p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
- $p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
- $\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Constructing New Logical Equivalences

- We can show that two expressions are logically equivalent by developing a series of logically equivalent statements.
- To prove that $A \equiv B$ we produce a series of equivalences beginning with A and ending with B.
  - $A \equiv A_1 \equiv A_2 \equiv \cdots \equiv A_n \equiv B$
- Keep in mind that whenever a proposition (represented by a propositional variable) occurs in the equivalences listed earlier, it may be replaced by an arbitrarily complex compound proposition.

Propositional Satisfiability

- A compound proposition is satisfiable if there is an assignment of truth values to its variables that make it true.
- When no such assignments exist, the compound proposition is unsatisfiable.
- A compound proposition is unsatisfiable if and only if its negation is a tautology.

Questions on Propositional Satisfiability

- $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$
  - $p \vee \neg q = T \Rightarrow \text{set } p = T$
  - $q \vee \neg r = T \Rightarrow \text{set } q = T$
  - $r \vee \neg p = T \Rightarrow \text{set } r = T$
  - One solution: $p = q = r = T$
- $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$
  - Not satisfiable.
  - Check each possible assignment of truth values to the propositional variables and none will make the proposition true.

Notation

- $\bigvee_{j=1}^{n} p_j \equiv p_1 \vee p_2 \vee \cdots \vee p_n$
- $\bigwedge_{j=1}^{n} p_j \equiv p_1 \wedge p_2 \wedge \cdots \wedge p_n$

Sudoku

- A Sudoku puzzle is represented by a 9×9 grid made up of nine 3×3 subgrids, known as blocks.
- Some of the 81 cells of the puzzle are assigned one of the numbers 1,2, …, 9.

- The puzzle is solved by assigning numbers to each blank cell so that every row, column and block contains each of the nine possible numbers.
- Example

	2	9				4		
			5			1		
	4							
				4	2			
6							7	
5								
7			3					5
	1			9				
							6	

- Encoding as a Satisfiability Problem
  - Let $p(i, j, n)$ denote the proposition that is true when the number n is in the cell in the ith row and the jth column.
  - There are $9 \times 9 \times 9 = 729$ such propositions.
  - In the sample puzzle $p(5,1,6)$ is true, but $p(5, j, 6)$ is false for $j = 2,3, \ldots, 9$
  - For each cell with a given value, assert $p(i, j, n)$, when the cell in row i and column j has the given value.
  - Assert that every row contains every number.
    - $\bigwedge_{i=1}^{9} \bigwedge_{n=1}^{9} \bigvee_{j=1}^{9} p(i, j, n)$
  - Assert that every column contains every number.
    - $\bigwedge_{j=1}^{9} \bigwedge_{n=1}^{9} \bigvee_{i=1}^{9} p(i, j, n)$
  - Assert that each of the 3×3 blocks contain every number.
    - $\bigwedge_{r=0}^{2} \bigwedge_{s=0}^{2} \bigwedge_{n=1}^{9} \bigwedge_{i=1}^{3} \bigvee_{j=1}^{3} p(3r + i, 3s + j, n)$
  - Assert that no cell contains more than one number. Take the conjunction over all values of n, n', i, and j, where each variable ranges from 1 to 9 and $n \neq n'$ of
    - $p(i, j, n) \rightarrow \neg p(i, j, n')$
- Solving Satisfiability Problems
  - To solve a Sudoku puzzle, we need to find an assignment of truth values to the 729 variables of the form $p(i, j, n)$ that makes the conjunction of the assertions

true.

- Those variables that are assigned to yield a solution to the puzzle.
- A truth table can always be used to determine the satisfiability of a compound proposition.
- But this is too complex even for modern computers for large problems.
- There has been much work on developing efficient methods for solving satisfiability problems as many practical problems can be translated into satisfiability problems.

1.4 Predicates and Quantifiers

Monday, January 29, 2018 9:25 AM

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: exists and for all
- 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).
- 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 be the integers.
- Find these truth values:
  - $R(2, -1, 5) = F$
  - $R(3, 4, 7) = T$
  - $R(x, 3, z) \Rightarrow$ Not a Proposition
- Now let "x is the least number" be denoted by $Q(x)$, with $U = \{0, 1, 2, 3, 5\}$.

- Find these truth values:
  - $Q(0) = T$
  - $Q(5) = F$
  - $Q(6) \Rightarrow$ undefined
- What is $Q(0)$ is U is the integers? $Q(0) = F$

Compound Expressions

- Connectives from propositional logic carry over to predicate logic.
- If $P(x)$ denotes "$x > 0$," find these truth values:
  - $P(3) \vee P(-1) = T \vee F = T$
  - $P(3) \vee P(-1) = T \wedge F = F$
  - $P(3) \rightarrow P(-1) = T \rightarrow F = F$
  - $P(3) \rightarrow \neg P(-1) = T \rightarrow T = T$
- 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

- 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 every x in the domain.
  - $\exists x\, P(x)$ asserts $P(x)$ is true for some x in the domain.
- The quantifiers are said to bind the variable x in these expressions.

Universal Quantifier

- $\forall x\, P(x)$ is read as "For all x, $P(x)$" or "For every x, $P(x)$"
- If $P(x)$ denotes "$x > 0$" and U is the integers, then $\forall x\, P(x)$ is false.
- If $P(x)$ denotes "$x > 0$" and U is the positive integers, then $\forall x\, P(x)$ is true.
- 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 $x, P(x)$."

- 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.
- If $P(x)$ denotes "$x < 0$" and U is the positive integers, then $\exists x\ P(x)$ is false.
- If $P(x)$ denotes "x is even" and U is the integers, then $\exists x\ P(x)$ is true.

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.

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.

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 $\big(\forall x\ P(x)\big) \vee Q(x)$
- $\forall x\ \big(P(x) \vee Q(x)\big)$ means something different.

Translating from English to Logic

- Every student in this class has taken a course in Java.
  - Solution 1
    - If $U =$ every student in the class
    - Let $J(x) \coloneqq x$ has taken a course in Java
    - $\forall x\ J(x)$
  - Solution 2
    - If $U =$ every student
    - Let $C(x) \coloneqq x$ is a student in the class

        - Let $J(x) \coloneqq x$ has taken a course in Java
        - Let $\forall x \left(C(x) \rightarrow J(x)\right)$
- Some but not all students in this class has taken a course in Java.
    - Let $C(x) \coloneqq x$ is a student in the class
    - Let $J(x) \coloneqq x$ has taken a course in Java
    - Some but not all
        - $\exists x\, J(x) \wedge \neg \forall x\, J(x)$
        - $\equiv \exists x\, J(x) \wedge \exists x\, \neg J(x)$
    - Solution
        - $\exists x\big(C(x) \wedge J(x)\big) \wedge \neg \forall x\big(C(x) \rightarrow J(x)\big)$
        - $\equiv \exists x\big(C(x) \wedge J(x)\big) \wedge \neg \forall x\, \big(C(x) \rightarrow J(x)\big)$
        - $\equiv \exists x\big(C(x) \wedge J(x)\big) \wedge \forall x\, \neg\big(C(x) \rightarrow J(x)\big)$
        - $\equiv \exists x\big(C(x) \wedge J(x)\big) \wedge \exists x\, \big(C(x) \wedge \neg J(x)\big)$

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

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 Java."
    - This implies that "There is a student in your class who has not taken Java."
    - Symbolically $\neg \forall x\, J(x)$ and $\exists x\, \neg J(x)$ are equivalent
- 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 Java"
    - Symbolically $\neg \exists x\, J(x)$ and $\forall x\, \neg J(x)$ are equivalent

Equivalent Statements

- $\forall x\big(P(x) \wedge Q(x)\big) \equiv \forall x\, P(x) \wedge \forall x\, Q(x)$
- $\forall x\big(P(x) \vee Q(x)\big) \neq \forall x\, P(x) \vee \forall x\, Q(x)$
  - Let $U = \mathbb{N} = \{0,1,2,3 \dots\}$
  - Let $P(x) \coloneqq x$ is even
  - Let $Q(x) \coloneqq x$ is odd
  - $\forall x\big(P(x) \vee Q(x)\big)$: every natural number is even or odd
  - $\forall x\, P(x) \vee \forall x\, Q(x)$: every natural number is even or every natural number is odd
- $\forall x\, P(x) \equiv \forall z\, P(z)$

Lewis Carroll Example

- 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."
- Define
  - $U \coloneqq$ all creatures
  - $L(x) \coloneqq x$ is a lion
  - $F(x) \coloneqq x$ is fierce
  - $C(x) \coloneqq x$ drinks coffee
- Translation
  - $\forall x\, \big(L(x) \rightarrow F(x)\big)$
  - $\exists x\, \big(L(x) \wedge \neg C(x)\big)$
  - $\exists x\big(F(x) \wedge \neg C(x)\big)$

Some Predicate Calculus Definitions

- An assertion involving predicates and quantifiers is valid if it is true
  - for all domains
  - every propositional function substituted for the predicates in the assertion.
- Example: $\forall x\, \neg S(x) \leftrightarrow \neg \exists x\, S(x)$
- An assertion involving predicates is satisfiable if it is true
  - for some domains
  - some propositional functions that can be substituted for the predicates in the assertion.
- Otherwise it is unsatisfiable.
- Example: $\forall x\, \big(F(x) \leftrightarrow T(x)\big)$ not valid but satisfiable
- Example: $\forall x\big(F(x) \wedge \neg F(x)\big)$ unsatisfiable

1.5 Nested Quantifiers

Wednesday, January 31, 2018 9:28 AM

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 cannot actually be carried out.

Order of Quantifiers

- 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.
- Let $Q(x, y)$ be the statement "$x + y = 0$."
  - Assume that U is the real numbers.
  - Then $\forall x \exists y\, Q(x, y)$ is true, but $\exists y \forall x\, Q(x, y)$ is false.

Questions on Order of Quantifiers

- Example 1
  - Let U be the real numbers,
  - Define $P(x, y)$: $x \cdot y = 0$
  - $\forall x \forall y P(x, y) = F$
  - $\forall x \exists y P(x, y) = T$
  - $\exists x \forall y P(x, y) = T$
  - $\exists x \exists y P(x, y) = T$

  Example 2
  - Let U be the positive real numbers,
  - Define $P(x, y)$: $x/y = 1$
  - $\forall x \forall y P(x, y) = F$
  - $\forall x \exists y P(x, y) = T$
  - $\exists x \forall y P(x, y) = F$
  - $\exists x \exists y P(x, y) = T$

Translating Nested Quantifiers into English

- Example 1
  - Translate the statement $\forall x \left(C(x) \vee \exists y \left(C(y) \wedge F(x, y)\right)\right)$
    - 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.
- Example 2
  - Translate the statement
    - $\exists x \forall y \forall z \left(\left(F(x, y) \wedge F(x, z) \wedge (y \neq z)\right) \rightarrow \neg F(y, z)\right)$
  - Solution
    - There is a student none of whose friends are also friends with each other.
- Example 3
  - Translate the statement $\forall x \left(B(x) \vee \exists y \left(B(y) \wedge S(y, x)\right)\right)$
    - Where $B(x)$ is "x is a barber,"
    - And $S(y, x)$ is "y shaves x"
    - And the domain for both x and y consists of all people in Jonesville

Translating Mathematical Statements into Predicate Logic

- Example 1
  - Every barber in Jonesville shaves those and only those who don't shave themselves
  - Solution: $\forall x \left(B(x) \rightarrow \forall y \left(S(x,y) \leftrightarrow \neg S(y,y)\right)\right)$
- Example 2
  - The sum of two positive integers is always positive
  - Solution: $\forall x \forall y \left(x > 0 \wedge y > 0\right) \Rightarrow x + y > 0$
  - Negation: $\exists x \exists y \left((x > 0) \wedge (y > 0) \wedge \neg(x + y > 0)\right)$
- Example 3
  - Every natural number can be represented as the sum of four squares
  - Solution: $\forall x \exists a \exists b \exists c \exists d \left(x = a^2 + b^2 + c^2 + d^2\right)$
  - Negation: $\exists x \forall a \forall b \forall c \forall d \left(x \neq a^2 + b^2 + c^2 + d^2\right)$
- Example 4
  - Translate the epsilon-delta definition for the limit of a function: $\lim_{x \to c} f(x) = L$
  - Solution: $\forall \varepsilon \left(\varepsilon > 0 \rightarrow \exists \delta \forall x \left(0 < |x - c| < \delta \rightarrow \left|f(x) - L\right| < \varepsilon\right)\right)$
  - Negation: $\exists \varepsilon \left(\varepsilon > 0 \wedge \forall \delta \exists x \left(0 < |x - c| < \delta \wedge \neg\left(\left|f(x) - L\right| < \varepsilon\right)\right)\right)$

1.6 Rules of Inference

Friday, February 2, 2018 9:08 AM

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:

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

- We will see shortly that this is a valid argument

Arguments in Propositional Logic

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

Rules of Inference for Propositional Logic:

- Modus Ponens
  - Equation

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

- Corresponding Tautology:
  - $\big(p \wedge (p \to q)\big) \to q$
- 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 Tollens
  - Equation

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

  - Corresponding Tautology:
    - $(\neg q \wedge (p \to q)) \to \neg p$
  - 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
  - Equation

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

  - Corresponding Tautology:
    - $((p \to q) \wedge (q \to r)) \to (p \to r)$
  - 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."
- Disjunctive Syllogism

- Equation

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

- 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
  - Equation

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

  - 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."
- Simplification
  - Equation

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

  - 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."
- Conjunction
  - Equation

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

- Corresponding Tautology:
  - $\big((p) \wedge (q)\big) \rightarrow (p \wedge q)$
- 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
  - Equation

$$\frac{\neg p \vee r}{p \vee q}$$
$$\therefore q \vee r$$

  - 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 study 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.

Valid Arguments

Example 1

- From the single proposition $p \wedge \big(p \rightarrow q\big)$
- Show that q is a conclusion.

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

Example 2

- With these hypotheses:
  - "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."
- Using the inference rules, construct a valid argument for the conclusion:
  - "We will be home by sunset."
- Choose propositional variables:
  - p: "It is sunny this afternoon."
  - r: "We will go swimming."
  - t: "We will be home by sunset."
  - q: "It is colder than yesterday."
  - s: "We will take a canoe trip."
- Translation into propositional logic:
  - Hypotheses: $\neg p \wedge q, r \rightarrow p, \neg r \rightarrow s, s \rightarrow t$
  - Conclusion: t
- Argument

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)

Handling Quantified Statements

- Universal Instantiation (UI)
  - Example:
    - Our domain consists of all dogs and Fido is a dog.
    - "All dogs are cuddly."
    - "Therefore, Fido is cuddly."
- Universal Generalization (UG)
  - Used often implicitly in Mathematical Proofs.

- Existential Instantiation (EI)
  - Example:
    - "There is someone who got an A in the course."
    - "Let's call her a and say that a got an A"
- Existential Generalization (EG)
  - Example:
    - "Michelle got an A in the class."
    - "Therefore, someone got an A in the class."

Using Rules of Inference

- Example 1
  - Using the rules of inference, construct a valid argument to show that
    - "John Smith has two legs"
  - is a consequence of the premises
    - "Every man has two legs."
    - "John Smith is a man."
  - Notation and domain
    - Let $M(x)$ denote "x is a man"
    - $L(x)$ "x has two legs"
    - Let John Smith be a member of the domain.
  - Argument

Step	Reason
1. $\forall x(M(x) \rightarrow L(x))$	Premise
2. $M(J) \rightarrow L(J)$	UI from (1)
3. $M(J)$	Premise
4. $L(J)$	Modus Ponens using (2) and (3)

- Example 2
  - Use the rules of inference to construct a valid argument showing that the conclusion
    - "Someone who passed the first exam has not read the book."

    follows from the premises

    "A student in this class has not read the book."

    "Everyone in this class passed the first exam."
  - Notation
    - Let C(x) denote "x is in this class."
    - B(x) denote "x has read the book."
    - P(x) denote "x passed the first exam."
  - First we translate the premises and conclusion into symbolic form.

$$\frac{\exists x(C(x) \wedge \neg B(x)) \quad \forall x(C(x) \rightarrow P(x))}{\therefore\ \exists x(P(x) \wedge \neg B(x))}$$

- Argument

Step	Reason
1. $\exists x(C(x) \wedge \neg B(x))$	Premise
2. $C(a) \wedge \neg B(a)$	EI from (1)
3. $C(a)$	Simplification from (2)
4. $\forall x(C(x) \rightarrow P(x))$	Premise
5. $C(a) \rightarrow P(a)$	UI from (4)
6. $P(a)$	MP from (3) and (5)
7. $\neg B(a)$	Simplification from (2)
8. $P(a) \wedge \neg B(a)$	Conj from (6) and (7)
9. $\exists x(P(x) \wedge \neg B(x))$	EG from (8)

Returning to the Socrates Example

- Premises and conclusion

$$\frac{\forall x(Man(x) \rightarrow Mortal(x)) \quad Man(Socrates)}{\therefore\ \ Mortal(Socrates)}$$

- Argument

Step	Reason
1. $\forall x(Man(x) \rightarrow Mortal(x))$	Premise
2. $Man(Socrates) \rightarrow Mortal(Socrates)$	UI from (1)
3. $Man(Socrates)$	Premise
4. $Mortal(Socrates)$	MP from (2) and (3)

The Barber Example

- Show that from the statements
- "Every barber in Jonesville shaves those and only those who don't shave themselves." and "There is a barber in Jonesville"
- We can derive a contradiction
  - $\forall x\left(B(x) \rightarrow \forall y\left(S(x,y) \leftrightarrow \neg S(y,y)\right)\right)$
  - $\exists x\ B(x)$
  - $\forall c\ B(c)$
  - $B(c) \rightarrow \forall y\left(S(c,y) \leftrightarrow \neg S(y,y)\right)$
  - $\forall y\ \left(S(c,y) \leftrightarrow \neg S(y,y)\right)$
  - $S(c,c) \leftrightarrow \neg S(c,c)$

- Thus, we have a contradiction

Lewis Carroll

- The first three are called premises and the third is called the conclusion
  - "All hummingbirds are richly colored."
  - "No large birds live on honey."
  - "Birds that do not live on honey are dull in color."
  - "Hummingbirds are small."
- Notation
  - $H(x) \coloneqq x$ is a hummingbird
  - $C(x) \coloneqq x$ is richly colored
  - $L(x) \coloneqq x$ is large
  - $Ho(x) \coloneqq x$ lives on honey
- Here is one way to translate these statements to predicate logic
  - $\forall x \left(H(x) \to C(x)\right)$
  - $\forall x \left(L(x) \to \neg Ho(x)\right)$
  - $\forall x \left(\neg Ho(x) \to \neg C(x)\right)$
  - $\forall x \left(H(x) \to \neg L(x)\right)$
- Let c be an arbilirary element of the universe

(1) $\forall x \left(H(x) \to C(x)\right)$

(2) $\forall x \left(L(x) \to \neg Ho(x)\right)$

(3) $\forall x \left(\neg Ho(x) \to \neg C(x)\right)$

(4) $H(c) \to C(c) \equiv \neg H(c) \vee C(c)$

(5) $L(c) \to \neg Ho(c) \equiv \neg L(c) \vee \neg Ho(c)$

(6) $\neg Ho(c) \to \neg C(c) \equiv Ho(c) \vee \neg C(c)$

(7) By resolution of (4) and (6), $\neg H(c) \vee Ho(c)$

(8) By resolution of (5) and (7), $\neg H(c) \vee \neg L(c) \equiv H(c) \to \neg L(c)$

(9) By (8), $\forall x\left(H(x) \to \neg L(x)\right)$

1.7 Introduction to Proofs

Monday, February 5, 2018 9:05 AM

Proofs of Mathematical Statements

- A proof is a valid argument that establishes the truth of a statement.
- In math, CS, and other disciplines, informal proofs which are generally shorter, are generally used.
    - More than one rule of inference are often used in a step.
    - Steps may be skipped.
    - The rules of inference used are not explicitly stated.
    - Easier for to understand and to explain to people.
    - But it is also easier to introduce errors.

Definitions

- A theorem is a statement that can be shown to be true using:
    - definitions
    - other theorems
    - axioms (statements which are given as true)
    - rules of inference
- A lemma is a 'helping theorem' or a result which is needed to prove a theorem.
- A corollary is a result which follows directly from a theorem.
- Less important theorems are sometimes called propositions.
- A conjecture is a statement that is being proposed to be true.
- Once a proof of a conjecture is found, it becomes a theorem, it may turn out to be false.

Forms of Theorems

- Many theorems assert that a property holds for all elements in a domain, such as the integers, the real numbers, or some of the discrete structures that we will study in this class.
- Often the universal quantifier (needed for a precise statement of a theorem) is omitted by standard mathematical convention.
- For example, the statement:
    - "If $x > y$, where x and y are positive real numbers, then $x^2 > y^2$"
    - really means
    - "For all positive real numbers x and y, if $x > y$, then $x^2 > y^2$."

Proving Theorems

- Many theorems have the form: $\forall x\big(P(x) \rightarrow Q(x)\big)$
- To prove them, we show that $P(c) \rightarrow Q(c)$
  where c is an arbitrary element of the domain,
- By universal generalization the truth of the original formula follows.
- So, we must prove something of the form: $p \rightarrow q$

Proving Conditional Statements: $p \rightarrow q$

- Trivial Proof
  - If we know q is true, then $p \rightarrow q$ is true as well.
  - "If it is raining then $1 = 1$.
- Vacuous Proof
  - If we know p is false then $p \rightarrow q$ is true as well.
  - "If I am both rich and poor then $2 + 2 = 5$."
- Even though these examples seem silly, both trivial and vacuous proofs
  are often used in mathematical induction, as we will see in Chapter 5
- Direct Proof
  - Assume that p is true.
  - Use rules of inference, axioms, and logical equivalences
    to show that q must also be true.
- Example 1 of Direct Proof
  - Give a direct proof of the theorem "If n is an odd integer, then n^2 is odd."
  - Assume that n is odd. Then $n = 2k + 1$ for an integer k.
  - Squaring both sides of the equation, we get:
    - $n^2 = (2k + 1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1 = 2r + 1$,
    - where $r = 2k\text{^}2 + 2k$, an integer.
  - We have proved that if n is an odd integer, then n^2 is an odd integer.
- Example 2 of Direct Proof
  - Prove that the sum of two rational numbers is rational.
  - Assume x and y are two rational numbers.
  - Then there must be integers p, q, r, s such that
  - $x = \frac{p}{q}, y = \frac{r}{s}$ and $s \neq 0, p \neq 0$
  - $x + y = \frac{p}{q} + \frac{r}{s} = \frac{ps + qr}{qs}$, where $q, s \neq 0$, and $\big(ps + qr\big), qs$ are integers
  - Hence, $x + y$ is rational
- Proof by Contraposition
  - Assume $\neg q$ and show $\neg p$ is true also.
  - This is sometimes called an indirect proof method.

- If we give a direct proof of $\neg q \rightarrow \neg p$ then we have a proof of $p \rightarrow q$.

- Example of Proof by Contraposition
  - Prove that for an integer n, if n^2 is odd, then n is odd.
  - Use proof by contraposition.
  - Assume n is even (i.e., not odd).
  - Therefore, there exists an integer k such that $n = 2k$.
  - Hence, $n^2 = 4k^2 = 2(2k^2)$, and n^2 is even(i.e., not odd).
  - We have shown that if n is an even integer, then n^2 is even.
  - Therefore by contraposition, for an integer n, if n^2 is odd, then n is odd.

- Proof by Contradiction: (AKA reductio ad absurdum).
  - To prove p, assume $\neg p$ and derive a contradiction such as $p \wedge \neg p$.
  - (an indirect form of proof).
  - Since we have shown that $\neg p \rightarrow F$ is true,
  - it follows that the contrapositive $T \rightarrow p$ also holds.

- Example of Proof by Contradiction
  - Use a proof by contradiction to give a proof that $\sqrt{2}$ is irrational.
  - Towards a contradiction assume that $\sqrt{2}$ is rational
  - Let a, b be such that $\sqrt{2} = \frac{a}{b}, b \neq 0$, and a, b have no common factors
  - $2 = \frac{a^2}{b^2} \Rightarrow 2b^2 = a^2$, so a^2 is even and a is even
  - Let $a \coloneqq 2k$ for some $k \in \mathbb{Z}$, then $a^2 = 4k^2$
  - Then $2b^2 = 4k^2 \Rightarrow b^2 = 2k$, so b^2 is even, and b is also even
  - So 2 divides a and b, which makes a contradiction ■

Theorems that are Biconditional Statements

- To prove a theorem that is a biconditional statement, that is,

  a statement of the form $p \leftrightarrow q$, we show that $p \rightarrow q$ and $q \rightarrow p$ are both true.

- Example
  - Prove the theorem: "If n is an integer, then n is odd if and only if n^2 is odd."
  - We have already shown (previous slides) that both $p \rightarrow q$ and $q \rightarrow p$.
  - Therefore we can conclude p ↔ q.

- Note
  - Sometimes iff is used as an abbreviation for "if an only if," as in "If n is an integer, then n is odd iif n^2 is odd."

Looking Ahead

- If direct methods of proof do not work:
- We may need a clever use of a proof by contraposition.

- Or a proof by contradiction.
- In the next section, we will see strategies that can be used
  when straightforward approaches do not work.
- In Chapter 5, we will see mathematical induction and related techniques.
- In Chapter 6, we will see combinatorial proofs

1.8 Proof Methods and Strategy

Monday, February 5, 2018 9:34 AM

Proof by Cases

- To prove a conditional statement of the form:
  - $(p_1 \vee p_2 \vee \cdots \vee p_n) \rightarrow q$
- Use the tautology
  - $(p_1 \vee p_2 \vee \cdots \vee p_n) \rightarrow q$

    $\updownarrow$
  - $(p_2 \rightarrow q) \wedge (p_2 \rightarrow q) \wedge \cdots \wedge (p_n \rightarrow q)$
- Each of the implications $p_i \rightarrow q$ is a case.
- Example
  - Let $a@b = max\{a, b\} = a$ if $a \geq b$,
  - otherwise $a@b = max\{a, b\} = b$.
  - Show that for all real numbers a, b, c
    - $(a@b)@c = a@(b@c)$
  - (This means the operation @ is associative.)
  - Let a, b, and c be arbitrary real numbers.
  - Then one of the following 6 cases must hold.
    - $a \geq b \geq c$
    - $a \geq c \geq b$
    - $b \geq a \geq c$
    - $b \geq c \geq a$
    - $c \geq a \geq b$
    - $c \geq b \geq a$

Without Loss of Generality

- Show that if x and y are integers and both $x \cdot y$ and $x + y$ are even,
- then both x and y are even.
- Use a proof by contraposition.
- Suppose x and y are not both even.
- Then, one or both are odd.
- Without loss of generality, assume that x is odd.
- Then $x = 2m + 1$ for some integer m.
- Case 1: y is even.
  - Then $y = 2n$ for some integer n, so
  - $x + y = (2m + 1) + 2n = 2(m + n) + 1$ is odd.

- Case 2: y is odd.
  - Then $y = 2n + 1$ for some integer n, so
  - $x \cdot y = (2m + 1)(2n + 1) = 2(2m \cdot n + m + n) + 1$ is odd.
- We only cover the case where x is odd
- because the case where y is odd is similar.
- The use phrase without loss of generality (WLOG) indicates this.

Existence Proofs

- Proof of theorems of the for $\exists x\, P(x)$.
- Constructive existence proof:
  - Find an explicit value of c, for which $P(c)$ is true.
  - Then $\exists x\, P(x)$ is true by Existential Generalization (EG).
- Example:
  - Show that there is a positive integer that can be written as
  - the sum of cubes of positive integers in two different ways:
  - $1729 = 10^3 + 9^3 = 12^3 + 1^3$
- Nonconstructive existence proof
  - In a nonconstructive existence proof,
  - we assume no c exists which makes $P(c)$ true
  - and derive a contradiction.
- Example
  - Show that there exist irrational numbers x, y such that x^y is rational.
  - We know that $\sqrt{2}$ is irrational.
  - Consider the number $\sqrt{2}^{\sqrt{2}}$.
  - If $\sqrt{2}^{\sqrt{2}}$ is rational
    - we have two irrational numbers x and y with x^y rational
    - namely $x = \sqrt{2}$ and $y = \sqrt{2}$.
  - If $\sqrt{2}^{\sqrt{2}}$ is irrational
    - then we can let $x = \sqrt{2}^{\sqrt{2}}$ and $y = \sqrt{2}$ so that
    - $x^y = \left(\sqrt{2}^{\sqrt{2}}\right)^{\sqrt{2}} = \sqrt{2}^{\sqrt{2}\times\sqrt{2}} = \sqrt{2}^2 = 2.$

Uniqueness Proofs

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

        - We show that an element x with the property exists.
    - Uniqueness
        - We show that if $y \neq x$, then y does not have the property.
- Example
    - Show that if a and b are real numbers and $a \neq 0$, then
    - there is a unique real number r such that $ar + b = 0$.
    - Existence
        - The real number $r = -\frac{b}{a}$ is a solution of $ar + b = 0$
        - because $a\left(-\frac{b}{a}\right) + b = -b + b = 0$.
    - Uniqueness
        - Suppose that s is a real number such that $as + b = 0$.
        - Then $ar + b = as + b$, where $r = -\frac{b}{a}$.
        - Subtracting b from both sides
        - and dividing by a shows that $r = s$.

Additional Proof Methods

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

2.1 Sets

Wednesday, February 7, 2018 9:00 AM

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.
  - $S = \{a, b, c, d\} = \{a, b, c, b, c, d\}$
- Dots (…) may be used to describe a set without listing all of the members when the pattern is clear.
  - $S = \{a, b, c, d, \dots, z\}$

Example of Roster Method

- Set of all vowels in the English alphabet:
  - $V = \{a, e, i, o, u\}$
- Set of all odd positive integers less than 10:
  - $O = \{1,3,5,7,9\}$
- Set of all positive integers less than 100:
  - $S = \{1,2,3, \dots, 99\}$
- Set of all integers less than 0:
  - $S = \{\dots, -3, -2, -1\}$

Some Important Sets

- $\mathbb{N}$ = natural numbers = $\{0,1,2,3 \dots\}$
- $\mathbb{Z}$ = integers = $\{\dots, -3, -2, -1,0,1,2,3, \dots\}$
- $\mathbb{Z}^+$ = positive integers = $\{1,2,3, \dots\}$
- $\mathbb{R}$ = set of real numbers
- $\mathbb{R}^+$ = set of positive real numbers

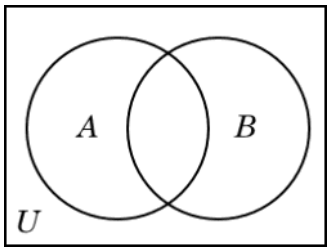
- $\mathbb{C} =$ set of complex numbers.
- $\mathbb{Q} =$ set of rational numbers

Set-Builder Notation

- Specify the property or properties that all members must satisfy:
  - $S = \{x | x \text{ is a positive integer less than } 100\}$
  - $O = \{x | x \text{ is an odd positive integer less than } 10\}$
  - $O = \{x \in \mathbb{Z}^+ | \ x \text{ is odd and } x < 10\}$
- A predicate may be used:
  - $S = \{x | P(x)\}$
- All prime numbers
  - $S = \{x | \text{Prime}(x)\}$
- Positive rational numbers:
  - $\mathbb{Q}^+ = \left\{x \in R \left| x = \frac{p}{q}, \text{for some positive integers } p, q\right.\right\}$

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



Russell's Paradox

- Let S be the set of all sets which are not members of themselves.
- A paradox results from trying to answer the question
- "Is S a member of itself?"
- Related Paradox:
  - Henry is a barber who shaves all people who do not shave themselves.
  - A paradox results from trying to answer the question
  - "Does Henry shave himself?"

Some things to remember

- Sets can be elements of sets.
  - $\{\{1,2,3\}, a, \{b, c\}\}$
  - $\{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$
- The empty set is different from a set containing the empty set.
  - $\emptyset \neq \{\emptyset\}$

Set Equality

- 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

- 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.
- Special Subsets
  - Because $a \in \emptyset$ is always false, $\emptyset \subseteq S$, for every set S.
  - 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
  - show that if x belongs to A, then x also belongs to B.
- Showing that A is not a Subset of 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 ∈ B.)
- Examples:
  - The set of all computer science majors at your school is a subset of all students at your school.
  - The set of integers with squares less than 100 is not a subset of the set of nonnegative integers.

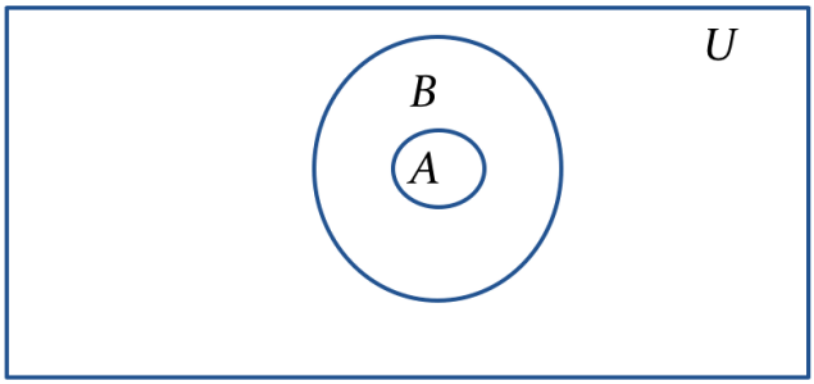
Another look at Equality of Sets

- 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

- $\forall x\big((x \in A \rightarrow x \in B) \wedge (x \in B \rightarrow x \in A)\big)$
- This is equivalent to
  - $A \subseteq B$ and $B \subseteq A$

Proper Subsets

- If $A \subseteq B$, but $A \neq B$, then we say A is a proper subset of B, denoted by $A \subset B$.
- If $A \subset B$, then $\forall x(x \in A \rightarrow x \in B) \wedge \exists(x \in B \wedge x \notin A)$ is true.
- Venn Diagram



Set Cardinality

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

Power Sets

- The set of all subsets of a set A, denoted $\mathcal{P}(A)$, is called the power set of A.
- Example
  - If $A = \{a, b\}$ then $\mathcal{P}(A) = \big\{\emptyset, \{a\}, \{b\}, \{a, b\}\big\}$
- 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 $(a_1, a_2, \dots, a_n)$ is the ordered collection that
  - has a_1 as its first element
  - and a_2 as its second element
  - and so on until an as its last element.
- Two n-tuples are equal if and only if their corresponding elements are equal.
- 2-tuples are called ordered pairs.
- The ordered pairs (a, b) and (c, d) are equal if and only if $a = c$ and $b = d$.
- Note: $(a, b) = \{a, \{a, b\}\}$

Cartesian Product

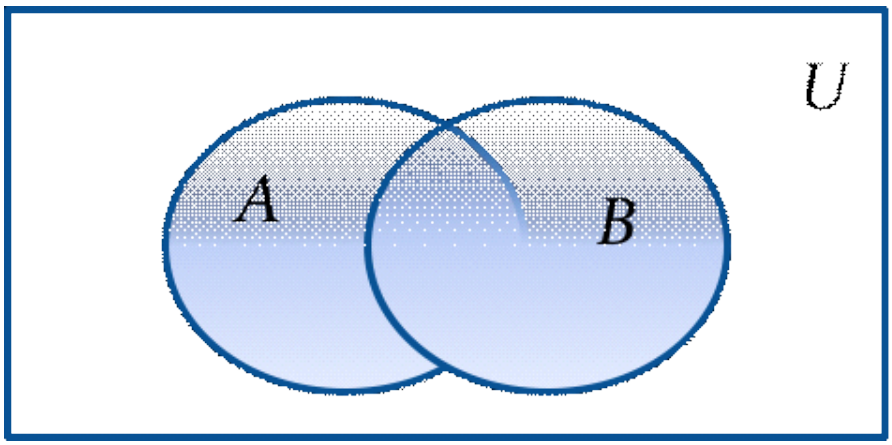
- Cartesian Product of two sets
  - The Cartesian Product of two sets A and B, denoted by $A \times B$ is
  - the set of ordered pairs (a, b) where $a \in A$ and $b \in B$.
  - $A \times B = \{(a, b) | a \in A \wedge b \in B\}$
- Example:
  - $A = \{a, b\}$
  - $B = \{1,2,3\}$
  - $A \times B = \{(a, 1), (a, 2), (a, 3), (b, 1), (b, 2), (b, 3)\}$
- Cartesian Product of more sets
  - The cartesian products of the sets $A_1, A_2, \dots, A_n$
  - denoted by $A_1 \times A_2 \times \cdots \times A_n$
  - is the set of ordered n-tuples $(a_1, a_2, \dots, a_n)$
  - where a_i belongs to A_i for $i = 1,2, \dots, n$
  - $A_1 \times A_2 \times \cdots \times A_n = \{(a_1, a_2, \dots, a_n) | a_i \in A_i \text{ for } i = 1,2, \dots, n\}$
- Example
  - What is A × B × C where A = {0,1}, B = {1,2} and C = {0,1,2}
  - $A \times B \times C$ = {(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), (1,2,2)}

2.2 Set Operations

Friday, February 9, 2018 8:50 AM

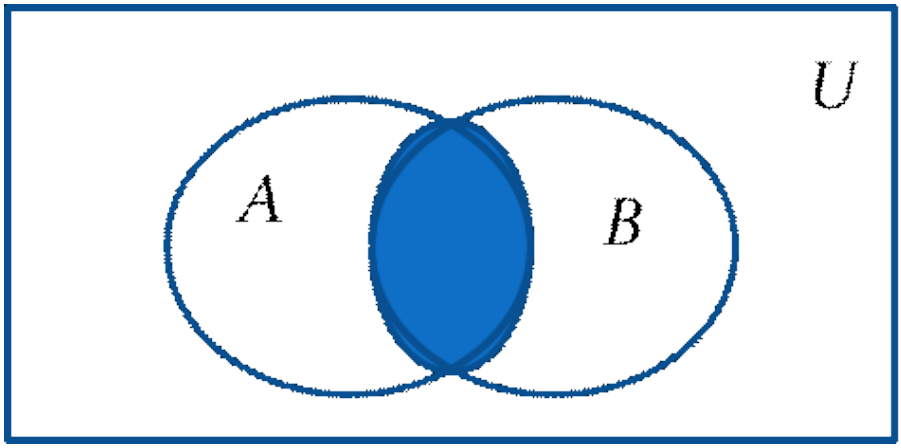
Union

- Definition
  - Let A and B be sets.
  - The union of the sets A and B, denoted by $A \cup B$, is the set:
  - $\{x | x \in A \vee x \in B\}$
- Example: What is $\{1,2,3\} \cup \{3,4,5\}$?
  - $\{1,2,3,4,5\}$
- Venn Diagram



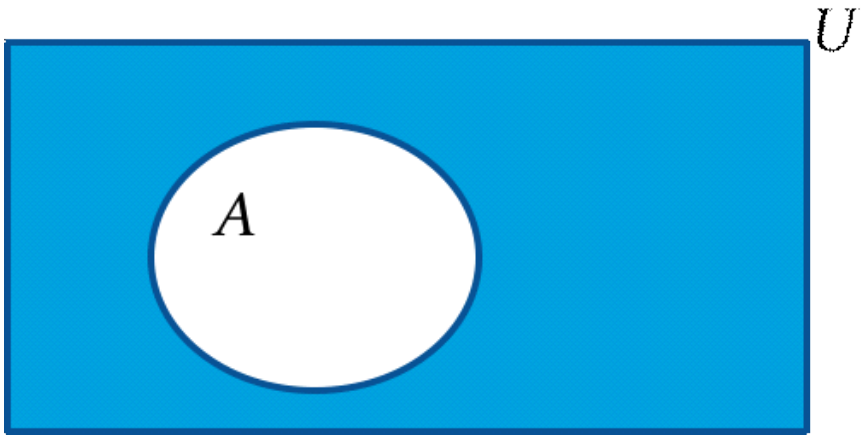
Intersection

- 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 $\{1,2,3\} \cap \{3,4,5\}$?
  - $\{3\}$
- Example: What is $\{1,2,3\} \cap \{4,5,6\}$?
  - $\emptyset$
- Venn Diagram



Complement

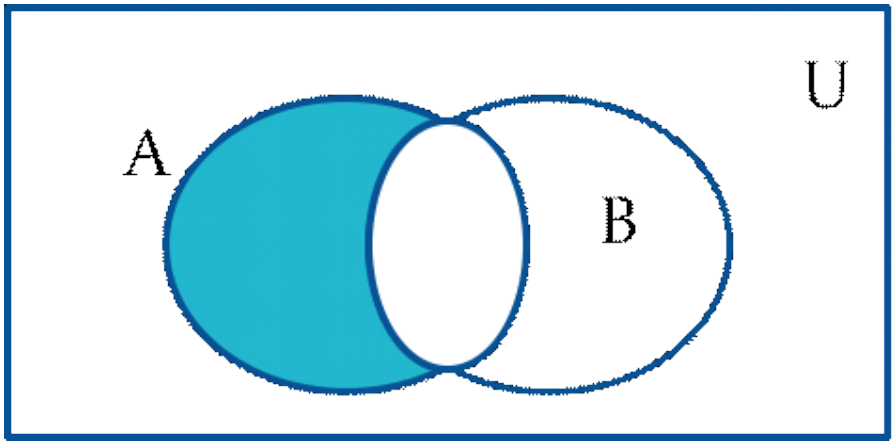
- Definition
  - If A is a set, then the complement of the A (with respect to U), denoted by $\bar{A}$ is the set $U - A$
  - $\bar{A} = \{x \in U | x \notin A\}$
  - (The complement of A is sometimes denoted by A^c.)
- Example
  - If U is the positive integers less than 100,
  - what is the complement of $\{x \mid x > 70\}$
  - $\{x | x \leq 70\}$
- Venn Diagram



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

- Venn Diagram



Set Identities

- Identity laws
  - $A \cup \emptyset = A$
  - $A \cap U = A$
- Domination laws
  - $A \cup U = U$
  - $A \cap \emptyset = \emptyset$
- Idempotent laws
  - $A \cup A = A$
  - $A \cap A = A$
- Complementation law
  - $\overline{(\overline{A})} = \overline{A}$
- Communtative laws
  - $A \cup B = B \cup A$
  - $A \cap B = B \cap A$
- Associative laws
  - $A \cup (B \cup C) = (A \cup B) \cup C$
  - $A \cap (B \cap C) = (A \cap B) \cap C$
- Distributive laws
  - $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
  - $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
- De Morgan's laws
  - $\overline{A \cup B} = \bar{A} \cap \bar{B}$
  - $\overline{A \cap B} = \bar{A} \cup \bar{B}$
- Absorption laws
  - $A \cup (A \cap B) = A$
  - $A \cap (A \cup B) = A$

- Complement laws
  - $A \cup \bar{A} = U$
  - $A \cap \bar{A} = \emptyset$

Proof of Second De Morgan Law

- Prove that $\overline{A \cap B} = \bar{A} \cup \bar{B}$
- We can prove this identity by showing that:
  - $\overline{A \cap B} \subseteq \bar{A} \cup \bar{B}$
  - $\bar{A} \cup \bar{B} \subseteq \overline{A \cap B}$
- Set-Builder Notation

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

Generalized Unions and Intersections

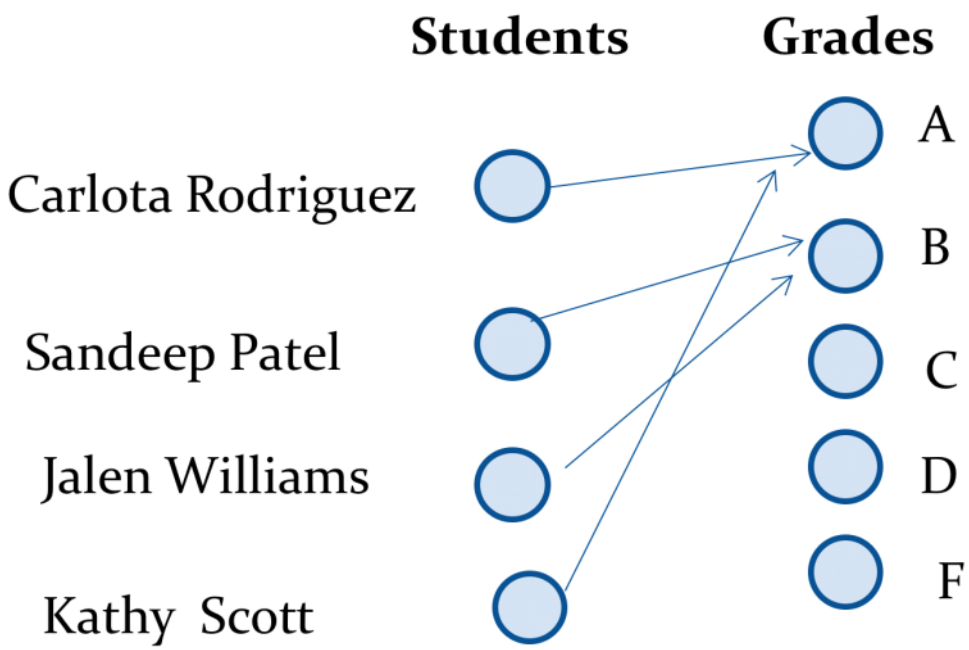
- Let $A_1, A_2, \ldots, A_n$ be an indexed collection of sets.
  - $\bigcup_{i=1}^{n} A_i = A_1 \cup A_2 \cup \cdots \cup A_n$
  - $\bigcap_{i=1}^{n} A_i = A_1 \cap A_2 \cap \cdots \cap A_n$
- These are well defined, since union and intersection are associative.

2.3 Functions

Friday, February 9, 2018 9:16 AM

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.
- Example



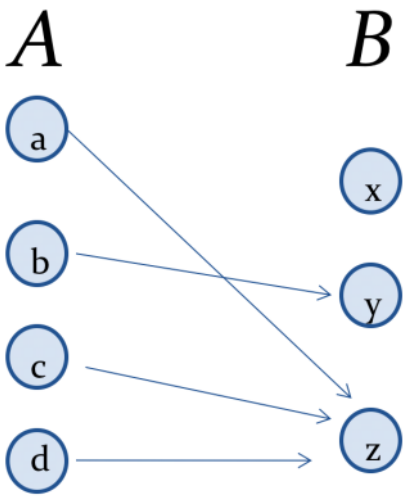
- Relation
  - A function $f: A \to B$ can also be defined as a subset of $A \times B$ (a relation).
  - This subset is restricted to be a relation where
    no two elements of the relation have the same first element.
  - Specifically, a function f from A to B contains
    one, and only one ordered pair (a, b) for every element $a \in A$.
    - $\forall x \left[x \in A \to \exists y [y \in B \wedge (x, y) \in f]\right]$
    - $\forall x, y_1, y_2 \left[[(x, y_1) \in f \wedge (x, y_2) \in f] \to y_1 = y_2\right]$
- Terminology

- Given a function f: A → 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.

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 $n!$

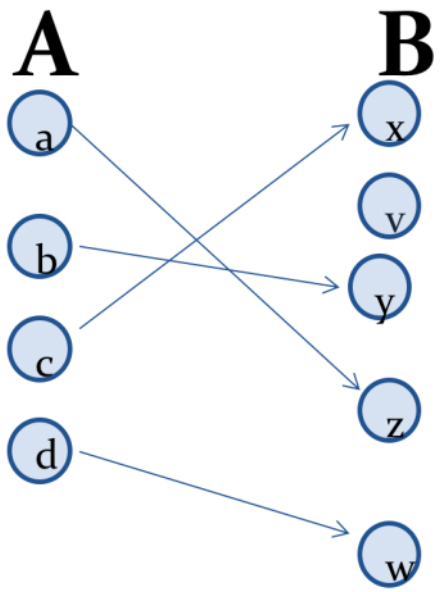
Example



- $f(a) = z$
- The image of d is z
- The domain of f is A
- The codomain of f is B
- The preimage of y is b
- $f(a) = \{y, z\}$
- The preimage of z is $\{a, c, d\}$
- $f\{a, b, c\} = \{y, z\}$
- $f\{c, d\} = \{z\}$

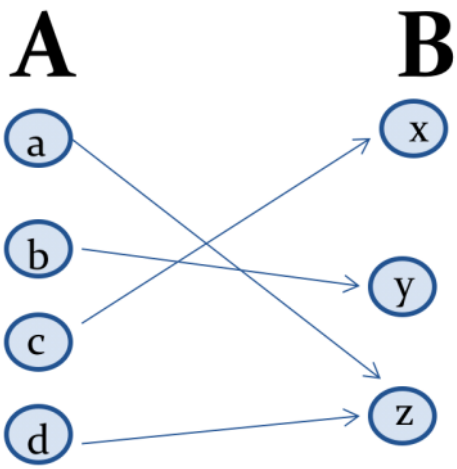
Injections

- 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.



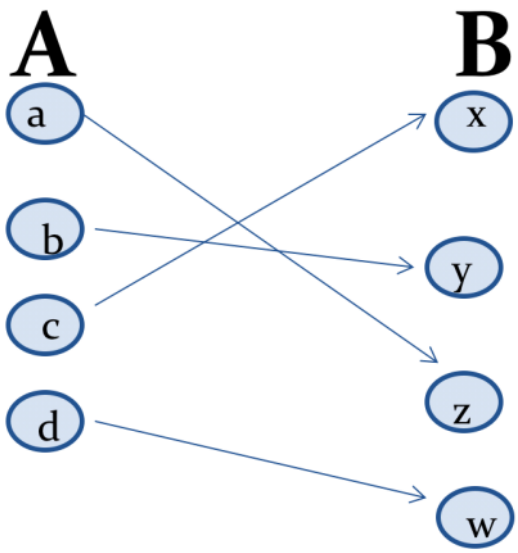
Surjections

- 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.



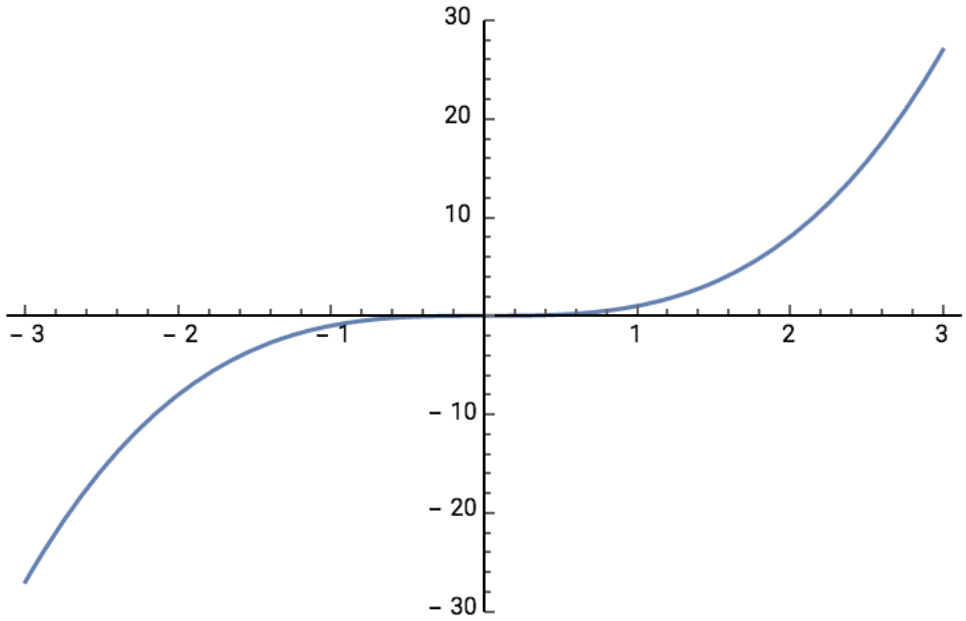
Bijections

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



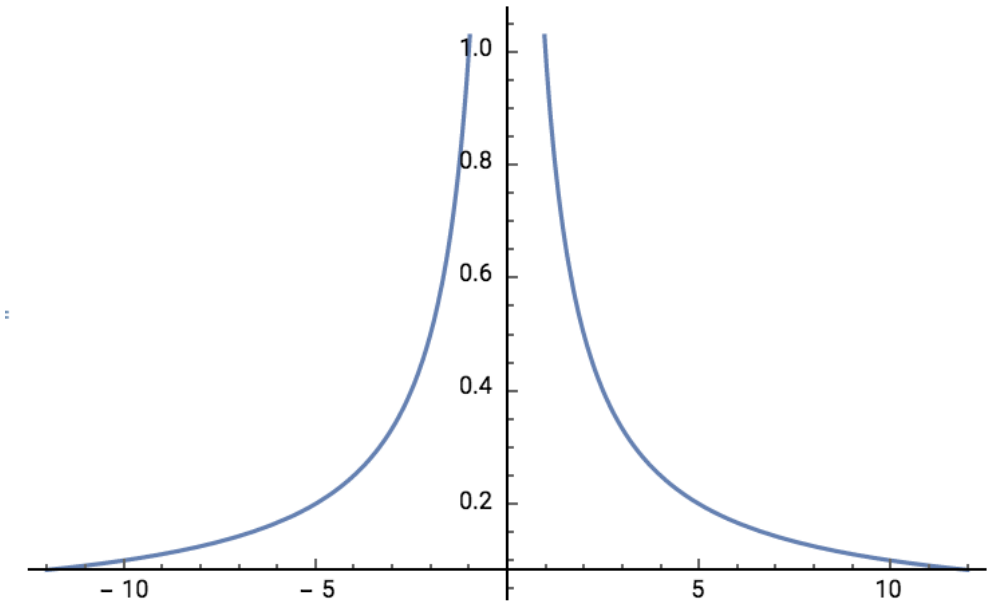
Showing that f is one-to-one or onto

- To show that f is injective
  - For $x, y, \in A$ if $x \neq y$ then $f(x) \neq f(y)$
- To show that f is not injective
  - Find $x, y \in A$ s.t. $x \neq y$ and $f(x) = f(y)$
- Example 1
  - Let f be the function from $\{a, b, c, d\}$ to $\{1,2,3\}$ defined by
    - $f(a) = 3$
    - $f(b) = 2$
    - $f(c) = 1$
    - $f(d) = 3$
  - Is f an onto function?
    - Yes, f is onto.
    - Since all 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
  - Is the function $f(x) = x^2$ from the set of integers to the set of integers onto?
  - No, f is not onto because there is no integer x with $x^2 = -1$, for example.
- Example 3
  - Let f be the function from the $\mathbb{N}$ to the even natural numbers defined by
  - $f(n) = 2n$. Is f an onto function? One to one?
  - f is an onto function, and f is one to one
- Example 4
  - Is the function $f(x) = x^3$ from $\mathbb{R}$ to $\mathbb{R}$ onto? One to one?



- f is an onto function, and f is one to one

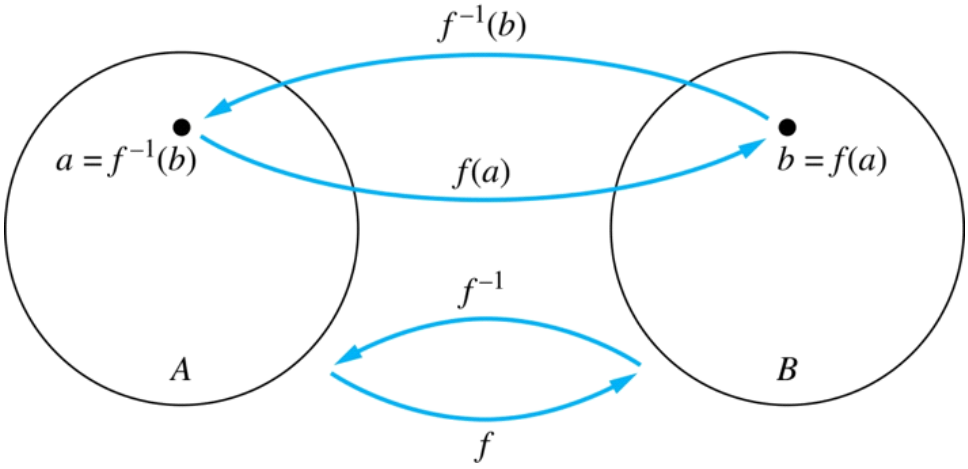
- Example 5
  - Is the function $f(x) = \frac{1}{|x|}$ fomr $\mathbb{R} \setminus \{0\}$ to $\mathbb{R}$ onto? One to one?



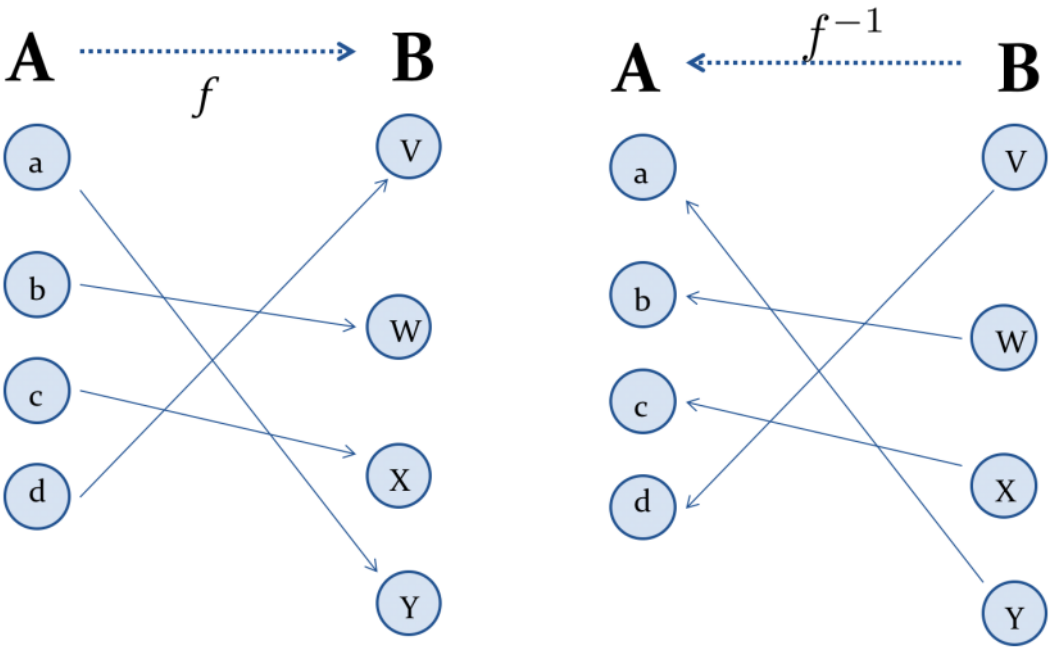
- f is not injective, and f is not surjective.

Inverse Functions

- 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
- $f^{-1}(y) = x$ iff $f(x) = y$
- No inverse exists unless f is a bijection. Why?



- Example



Questions

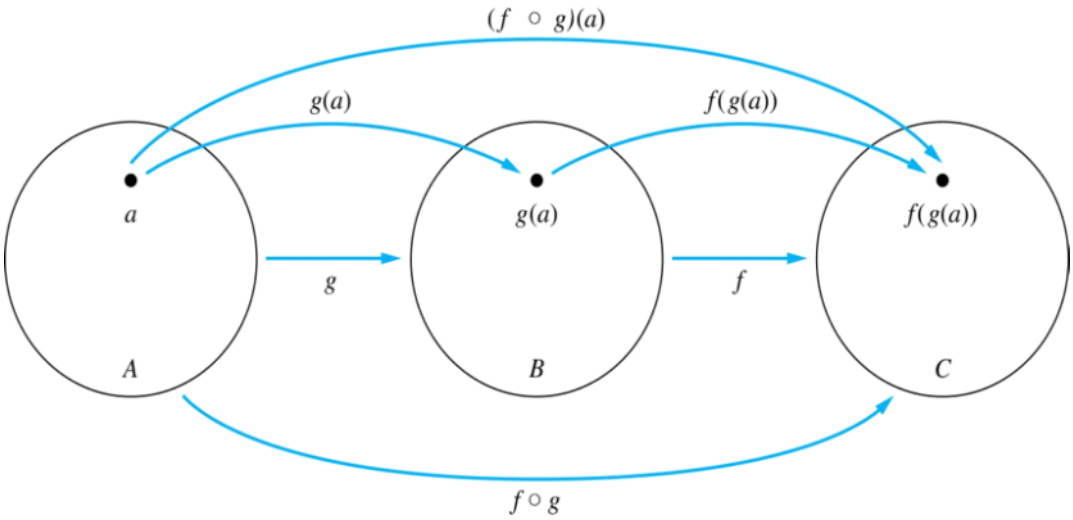
- Example 1
  - Let f be the function from $\{a, b, c\}$ to $\{1,2,3\}$ such that
    - $f(a) = 2$
    - $f(b) = 3$
    - $f(c) = 1$
  - Is f invertible and if so what is its inverse?
  - The function f is invertible because it is a one-to-one correspondence.
  - The inverse function f^{-1} reverses the correspondence given by f, so
    - $f^{-1}(1) = c$
    - $f^{-1}(2) = b$
    - $f^{-1}(3) = a$
- Example 2
  - Let $f: \mathbb{Z} \to \mathbb{Z}$ be such that $f(x) = x + 1$.

- Is f invertible, and if so, what is its inverse?
- The function f is invertible because it is a one-to-one correspondence.
- The inverse function f^{-1} reverses the correspondence so $f^{-1}(y) = y - 1$.

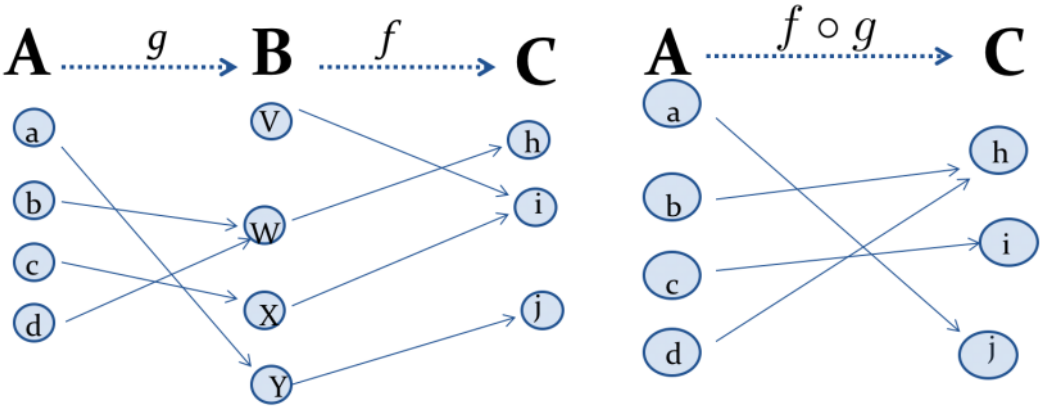
- Example 3
  - Let $f: \mathbb{R} \to \mathbb{R}$ be such that $f(x) = x^2$.
  - Is f invertible, and if so, what is its inverse?
  - The function f is not invertible because it is not one-to-one.

Composition

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



- Example



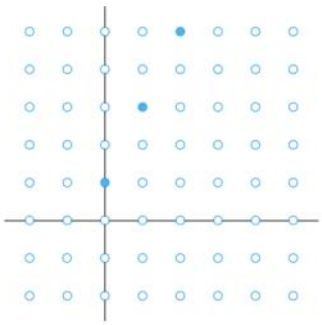
Composition Questions

- Example 1
  - If $f(x) = x^2$ and $g(x) = 2x + 1$
  - Then $f(g(x)) = (2x + 1)^2$
  - And $g(f(x)) = 2x^2 + 1$

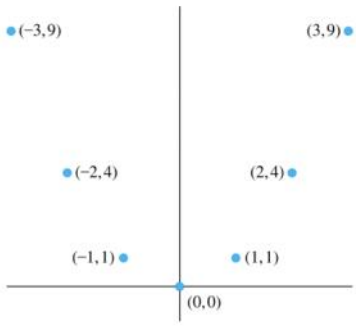
- Example 2
  - Let f and g be functions from the set of integers to the set of integers defined by
    - $f(x) = 2x + 3$
    - $g(x) = 3x + 2$
  - What is the composition of f and g, and also the composition of g and f ?
    - $f \circ g(x) = f(g(x)) = f(3x + 2) = 2(3x + 2) + 3 = 6x + 7$
    - $g \circ f(x) = g(f(x)) = g(2x + 3) = 3(2x + 3) + 2 = 6x + 11$

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 $\{(a, b)|a \in A \text{ and } f(a) = b\}$.
- Example



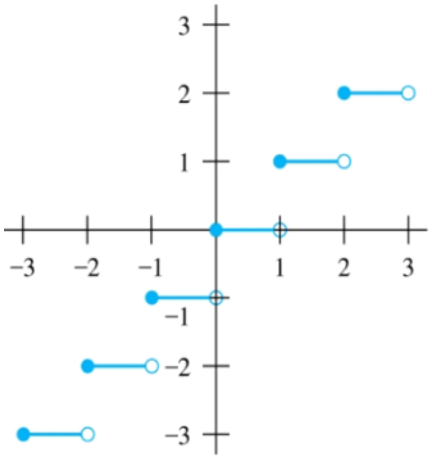
Graph of $f(n) = 2n + 1$ from Z to Z



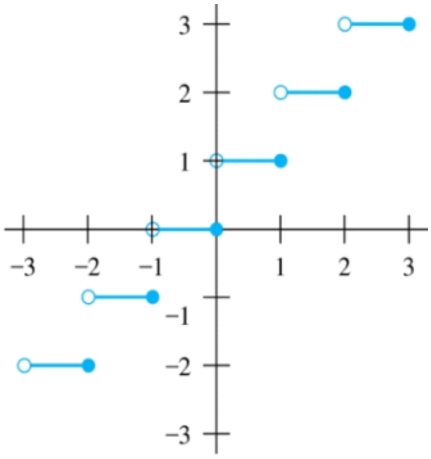
Graph of $f(x) = x^2$ from Z to Z

Some Important Functions

- The floor function $f(x) = \lfloor x \rfloor$ is the largest integer less than or equal to x.
- The ceiling function $f(x) = \lceil x \rceil$ is the smallest integer greater than or equal to x
- Example
  - $\lceil 3.5 \rceil = 4, \lfloor 3.5 \rfloor = 3$
  - $\lceil -1.5 \rceil = -1, \lfloor -1.5 \rfloor = -2$



(a) $y = [x]$



(b) $y = [x]$

Factorial Function

- $f: \mathbb{N} \to \mathbb{Z}^+$, denoted by $f(n) = n!$ is
- the product of the first n positive integers when n is a nonnegative integer.
  - $f(n) = 1 \cdot 2 \cdots (n - 1) \cdot n$,
  - $f(0) = 0! = 1$
- Examples:
  - $f(1) = 1! = 1$
  - $f(2) = 2! = 1 \cdot 2 = 2$
  - $f(6) = 6! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$
  - $f(20) = 2{,}432{,}902{,}008{,}176{,}640{,}000$

Partial Function

- A partial function f from a set A to a set B is

  an assignment to each element a in a subset of A, of a unique element b in B.
- The sets A and B are called the domain and codomain of f, respectively.
- We day that f is undefined for elements in A that are not in the domain of definition of f.
- When the domain of definition of f equals A, we say that f is a total function.
- Example
  - $f: \mathbb{N} \to \mathbb{Z}$ where $f(n) = \sqrt{n}$ is a partial function from $\mathbb{Z}$ to $\mathbb{R}$
  - where the domain of definition is the set of nonnegative integers.
  - Note that f is undefined for negative integers.

2.4 Sequences and Summations

Monday, February 12, 2018 9:35 AM

Introduction

- Sequences are ordered lists of elements.
  - 1,2,3,5,8
  - 1,3,9,27,81, …
- 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.

Sequences

- Definition
  - A sequence is a function from a subset of the integers 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.
- Example
  - Consider the sequence $\{a_n\}$ where
  - $a_n = \frac{1}{n}, \{a_n\} = \{a_1, a_2, a_3, \dots\}$
  - $1, \frac{1}{2}, \frac{1}{3}, \dots$

Strings

- 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.

Geometric Progression

- Definition
  - A geometric progression is a sequence of the form: $a, ar, ar^2, \dots, ar^n, \dots$
  - where the initial term a and the common ratio r are real numbers
- Example 1
  - Let $a = 1$ and $r = -1$. Then:
  - $\{b_n\} = \{b_0, b_1, b_2, b_3, b_4, \dots\} = \{1, -1, 1, -1, 1, \dots\}$

- Example 2
  - Let $a = 2$ and $r = 5$. Then:
  - $\{c_n\} = \{c_0, c_1, c_2, c_3, c_4, \dots\} = \{2,10,50,250,1250, \dots\}$
- Example 3
  - Let $a = 6$ and $r = 1/3$. Then:
  - $\{d_n\} = \{d_0, d_1, d_2, d_3, d_4, \dots\} = \left\{6,2,\frac{2}{3},\frac{2}{9},\frac{2}{27}, \dots\right\}$

Arithmetic Progression

- Definition
  - A arithmetic progression is a sequence of the form
  - $a, a + d, a + 2d, \dots, a + nd, \dots$
  - where the initial term a and the common difference d are real numbers
- Example 1
  - Let a $= -1$ and d $= 4$:
  - $\{s_n\} = \{s_0, s_1, s_2, s_3, s_4 \dots\} = \{-1,3,7,11,15 \dots\}$
- Example 2
  - Let a $= 7$ and d $= -3$:
  - $\{t_n\} = \{t_0, t_1, t_2, t_3, t_4 \dots\} = \{7,4,1,-2,-5 \dots\}$
- Example 3
  - Let a $= 1$ and d $= 2$:
  - $\{u_n\} = \{u_0, u_1, u_2, u_3, u_4 \dots\} = \{1,3,5,7,9 \dots\}$

Recurrence Relations

- A recurrence relation for the sequence $\{a_n\}$ is an equation that
  - expresses a_n in terms of a_0, a_1, …, 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.
- Example
  - 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?
  - $a_1 = a_0 + 3 = 2 + 3 = 5$
  - $a_2 = 5 + 3 = 8$
  - $a_3 = 8 + 3 = 11$

Fibonacci Sequence

- Define the Fibonacci sequence, f0 ,f1 ,f2,..., by:
  - Initial Conditions: $f_0 = 0, f_1 = 1$
  - Recurrence Relation: $f_n = f_{n-1} + f_{n-2}$
- Find f_2, f_3, f_4, f_5, f_6
  - $f_2 = f_1 + f_0 = 1 + 0 = 1$
  - $f_3 = f_2 + f_1 = 1 + 1 = 2$
  - $f_4 = f_3 + f_2 = 2 + 1 = 3$
  - $f_5 = f_4 + f_3 = 3 + 2 = 5$
  - $f_6 = f_5 + f_4 = 5 + 3 = 8$

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).
- Method 1: Working upward, forward substitution
  - Let $\{a_n\}$ be a sequence that satisfies the recurrence relation
  - $a_n = a_{n-1} + 3$ for $n = 2,3,4,$ ... and suppose that $a_1 = 2$
  - $a_2 = 2 + 3$
  - $a_3 = (2 + 3) + 3 = 2 + 3 \cdot 2$
  - $a_4 = (2 + 2 \cdot 3) + 3 = 2 + 3 \cdot 3$
  - ⋮
  - $a_n = a_{n-1} + 3 = \big(2 + 3 \cdot (n-2)\big) + 3 = 2 + 3(n-1)$
- Method 2: Working downward, backward substitution
  - Let $\{a_n\}$ be a sequence that satisfies the recurrence relation
  - $a_n = a_{n-1} + 3$ for $n = 2,3,4,$ ... and suppose that $a_1 = 2$
  - $a_n = a_{n-1} + 3$
  - $= (a_{n-2} + 3) + 3 = a_{n-2} + 3 \cdot 2$
  - $= (a_{n-3} + 3) + 3 \cdot 2 = a_{n-3} + 3 \cdot 3$
  - $= \cdots$
  - $= a_2 + 3(n-2) = (a_1 + 3) + 3(n-2) = 2 + 3(n-1)$

Summations

- Sum of the terms $a_m, a_{m+1}, \ldots, a_n$ from the sequence $\{a_n\}$

- Notation for $a_m + a_{m+1} + \cdots + a_n$
  - $\sum_{j=m}^{n} a_j$
- The variable j is called the index of summation. It runs through all the integers starting with its lower limit m and ending with its upper limit n.
- More generally for a set S
  - $\sum_{j \in S} a_j$
- Examples:
  - $r^0 + r^1 + r^2 + r^3 + \cdots + r^n = \sum_{j=0}^{n} r^j$
  - $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots = \sum_{i=1}^{\infty} \frac{1}{i}$
  - If $S = \{2,5,7,10\}$ then $\sum_{j \in S} a_j = a_2 + a_5 + a_7 + a_{10}$

Geometric Series

- Sums of terms of geometric progressions
- $\sum_{j=0}^{n} ar^j = \begin{cases} \frac{ar^{n+1} - a}{r-1} & r \neq 1 \\ (n+1)a & r = 1 \end{cases}$

Product Notation

- Sum of the terms $a_m, a_{m+1}, \ldots, a_n$ from the sequence $\{a_n\}$
- Notation for $a_m \times a_{m+1} \times \cdots \times a_n$
  - $\prod_{j=m}^{n} a_j$

2.5 Cardinality of Sets

Wednesday, February 14, 2018 9:31 AM

Cardinality

- The cardinality of a set A is equal to the cardinality of a set B, denoted $|A| = |B|$,
- if and only if there is a one-to-one correspondence (i.e., a bijection) from A to B
- If there is a one-to-one function (i.e., an injection) from A to B, then
- the cardinality of A is less than or equal to the cardinality of B and we write $|A| \leq |B|$
- When $|A| \leq |B|$ and A and B have different cardinality,
- we say that the cardinality of A is less than the cardinality of B and write $|A| < |B|$

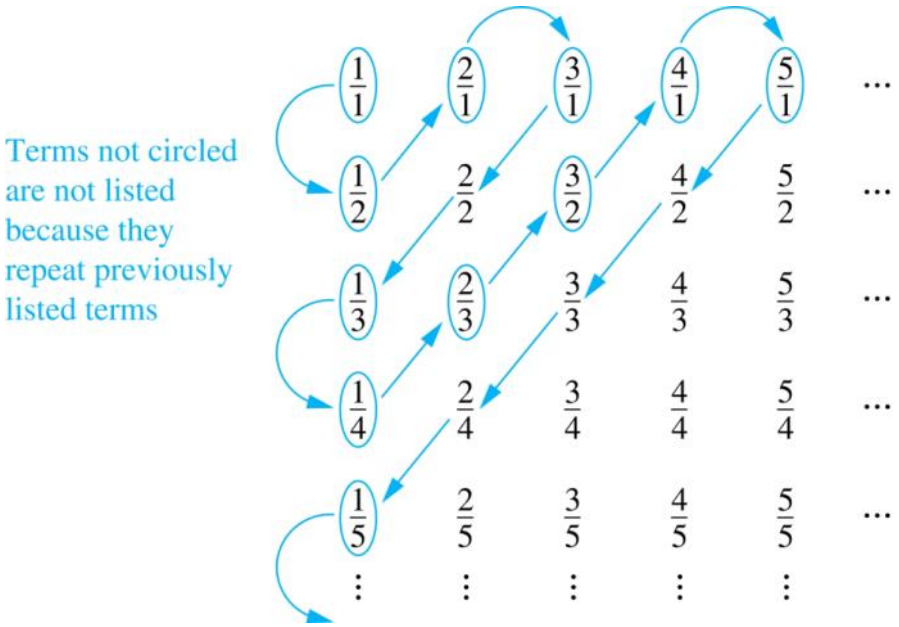
Countable and Uncountable

- A set that is either finite or has the same cardinality as $\mathbb{Z}^+$ is called countable.
- A set that is not countable is uncountable.
- The set of real numbers $\mathbb{R}$ is an uncountable set.
- When an infinite set is countable (countably infinite) its cardinality is $\aleph_0$
- (where $\aleph$ is aleph, the 1st letter of the Hebrew alphabet).
- We write $|S| = \aleph_0$ and say that S has cardinality "aleph null."

Showing that a Set is Countable

- An infinite set is countable iff it is possible to list the elements of the set in a sequence.
- A 1-1 correspondence f from the set of positive integers to a set S can be expressed
- in terms of a sequence $a_1, a_2, \ldots, a_n$ where $a_1 = f(1), a_2 = f(2), \ldots, a_n = f(n)$
- Example 1: The set of positive even integers E is countable set.
  - Let $f(x) = 2x$
  - $\begin{matrix} 1 & 2 & 3 & 4 & \ldots \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow \\ 2 & 4 & 6 & 8 & \ldots \end{matrix}$
  - Then f is a bijection from $\mathbb{N}$ to E since f is both one-to-one and onto.
  - To show that it is one-to-one, suppose that $f(n) = f(m)$.
  - Then $2n = 2m$, and so $n = m$.
  - To see that it is onto, suppose that t is an even positive integer.
  - Then $t = 2k$ for some positive integer k and $f(k) = t$.
- Example 2: The set of integers $\mathbb{Z}$ is countable.
  - Can list in a sequence: $0, 1, -1, 2, -2, 3, -3, \ldots$
  - Or can define a bijection from $\mathbb{N}$ to $\mathbb{Z}$:
    - $f(2n) = -n$
    - $f(2n+1) = n+1$

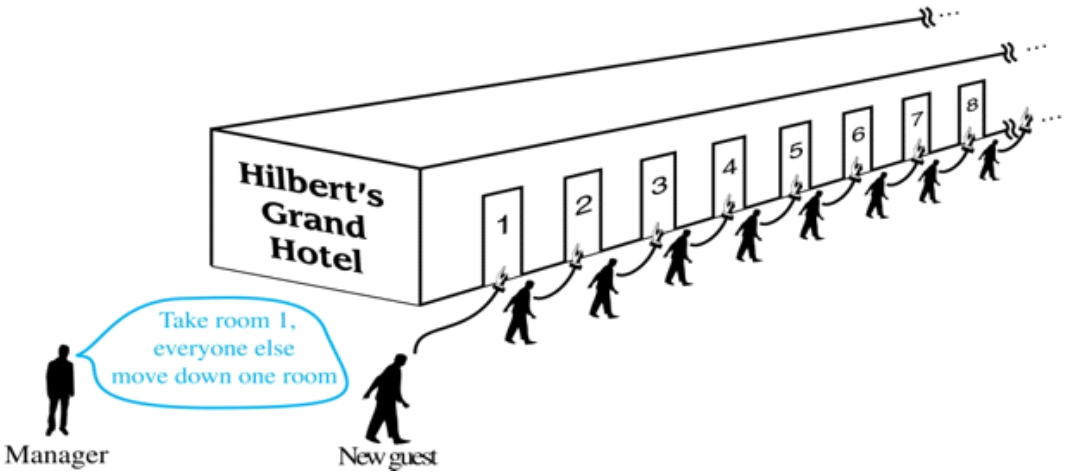
- Example 3: The positive rational numbers are countable.
  - A rational number can be expressed as $\frac{p}{q}$ where $p, q \in \mathbb{Z}$ and $q \neq 0$.
  - Note:
    - p and q such that $q \neq 0$.
  - The positive rational numbers are countable since they can be arranged in a sequence



- Example 4: Union of countable sets is countable
  - $A = \{a_1, a_2, \dots, a_n, \dots\}$
  - $B = \{b_1, b_2, \dots, b_n, \dots\}$
  - $A \cup B = \{a_1, b_1, a_2, b_2, \dots, a_n, b_n, \dots\}$
- Example 5: The set of all rationals is countable
  - $\mathbb{Q} = \{0\} \cup \mathbb{Q}^+ \cup \mathbb{Q}^-$
  - $f\left(-\frac{p}{q}\right) = \frac{p}{q}$ is a bijiection from $\mathbb{Q}^-$ to $\mathbb{Q}^+$
- Example 6: The set of finite string S over a finite alphabet A is counitable infinite
  - $A = \{a, b\}$
  - List all strings with length
    - 0: λ
    - 1: a, b
    - 2: aa, ab, ba, bb
    - 3: $aaa, aab, aba, abb, baa, bab, bba, bbb$
    - …
- Example 7: Show that the set of all Java program is countable
  - Just list all the strings

Hilbert's Grand Hotel

- The Grand Hotel has countably infinite number of rooms, each occupied by a guest.
- We can always accommodate a new guest at this hotel.
- How is this possible?
- Because the rooms of Grand Hotel are countable
- We can list them as Room 1, Room 2, Room 3, and so on.
- When a new guest arrives, we move the guest in Room n to Room $n + 1$
- This frees up Room 1, which we assign to the new guest, and all the current guests still have rooms.
- The hotel can also accommodate a countable number of new guests, all the guests on a countable number of buses where each bus contains a countable number of guests



The Real Numbers are Uncountable

- The method is called the Cantor diagnalization argument
- Suppose $\mathbb{R}$ is countable. Then the real numbers between 0 and 1 are also countable
- The real numbers between 0 and 1 can be listed in order r1 , r2 , r3 ,... .
- Let the decimal representation of this listing be
  - $r_1 = 0.d_{11}d_{12}d_{13}$ ...
  - $r_2 = 0.d_{21}d_{22}d_{23}$ ...
  - $r_3 = 0.d_{31}d_{32}d_{33}$ ...
- Form a new real number with the decimal expansion $r = 0.s_1s_2s_3$ ... where
  - $s_n = \begin{cases} 4 & d_{nn} = 4 \\ 3 & d_{nn} \neq 3 \end{cases}$
- r is not equal to any of the $r_1, r_2, r_3, \ldots$
- Because it differs from r_i in its i-th position after the decimal point.
- Therefore there is a real number between 0 and 1 that is not on the list
- since every real number has a unique decimal expansion.
- Hence, all the real numbers between 0 and 1 cannot be listed
- so the set of real numbers between 0 and 1 is uncountable.
- Since a set with an uncountable subset is uncountable

- the set of real numbers is uncountable.

Computability

- We say that a function is computable if there is a computer program in some programming language that finds the values of this function.
- If a function is not computable we say it is uncomputable.
- There are uncomputable functions.
- We have shown that the set of Java programs is countable.
- We can show that the set of functions $f: \mathbb{N} \to \mathbb{N}$ is uncountable
- Therefore there must be uncomputable functions

2.6 Matrices

Monday, February 19, 2018 8:48 AM

Matrices

- Matrices are useful discrete structures that can be used in many ways.
- For example, they are used to:
  - describe certain types of functions known as linear transformations.
  - Express which vertices of a graph are connected by edges (see Chapter 10).
- Here we cover the aspect of matrix arithmetic that will be needed later.
- Definition
  - A matrix is a rectangular array of numbers.
  - A matrix with m rows and n columns is called an $m \times n$ matrix.
  - The plural of matrix is matrices.
  - A matrix with the same number of rows as columns is called square.
  - Two matrices are equal if they have the same number of rows and the same number of columns and the corresponding entries in every position are equal.
- Example: 3×2 matrix
  - $\begin{bmatrix} 1 & 1 \\ 0 & 2 \\ 1 & 3 \end{bmatrix}$
- Notation
  - Let m and n be positive integers and let
    - $A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$
  - The ith row of A is the $1 \times n$ matrix
    - $[a_{i1}, a_{i2}, \dots, a_{in}]$.
  - The jth column of A is the $m \times 1$ matrix:
    - $\begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{bmatrix}$
  - The (i, j)th element or entry of A is the element a_{ij}
  - We can use $A = [a_{ij}]$ to denote the matrix with its (i, j)th element equal to a_{ij}

Matrix Arithmetic: Addition

- Let $A = [a_{ij}]$ and $B = [b_{ij}]$ be $m \times n$ matrices.
- The sum of A and B, denoted by $A + B$, is the $m \times n$ matrix that has $a_{ij} + b_{ij}$ as its (i, j)th element.

- In other words, A + B = $[a_{ij}$ + bij].
- Example
  - $\begin{bmatrix}1 & 0 & -1\\2 & 2 & -3\\3 & 4 & 0\end{bmatrix}+\begin{bmatrix}3 & 4 & -1\\1 & -3 & 0\\-1 & 1 & 2\end{bmatrix}=\begin{bmatrix}4 & 4 & -2\\3 & -1 & -3\\2 & 5 & 2\end{bmatrix}$
- Note that matrices of different sizes cannot be added.

Matrix Multiplication

- Let A be an $m \times k$ matrix and B be a $k \times n$ matrix.
- The product of A and B, denoted by AB, is the $m \times n$ matrix that has its (i, j)th element equal to the sum of the products of the corresponding elements from the ith row of A and the jth column of B.
- In other words, if AB = $[c_{ij}]$ then $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{kj}b_{kj}$.
- Example
  - $\begin{bmatrix}1 & 0 & 4\\2 & 1 & 1\\3 & 1 & 0\\0 & 2 & 2\end{bmatrix}\begin{bmatrix}2 & 4\\1 & 1\\3 & 0\end{bmatrix}=\begin{bmatrix}14 & 4\\8 & 9\\7 & 13\\8 & 2\end{bmatrix}$

Matrix Multiplication is not Commutative

- Let $A=\begin{bmatrix}1 & 1\\2 & 1\end{bmatrix}, B=\begin{bmatrix}2 & 1\\1 & 1\end{bmatrix}$, then
- $AB=\begin{bmatrix}3 & 2\\5 & 3\end{bmatrix}, BA=\begin{bmatrix}4 & 3\\3 & 2\end{bmatrix}$
- Thus $AB \neq BA$

Identity Matrix and Powers of Matrices

- The identity matrix of order n is the $m \times n$ matrix $I_n = \left[\delta_{ij}\right]$, where
  - δ_{ij} = 1 if i = j
  - δ_{ij} = 0 if $i \neq j$
- $I_n=\begin{bmatrix}1 & & \\ & \ddots & \\ & & 1\end{bmatrix}$
- $AI_n = I_mA = A$ when A is an $m \times n$ matrix
- Powers of square matrices can be defined. When A is an $n \times n$ matrix, we have:
  - $A^0 = I_n$
  - $A^r = \underbrace{AA\cdots A}_{r \text{ times}}$

Transposes of Matrices

- Let A = $[a_{ij}]$ be an $m \times n$ matrix.
- The transpose of A, denoted by A^t ,is
- the $n \times n$ matrix obtained by interchanging the rows and columns of A.

- If $A^t = [b_{ij}]$, then $b_{ij} = a_{ji}$ for $i = 1,2,\dots,n$ and $j = 1,2,\dots,m$
- The transpose of the matrix $\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\end{bmatrix}$ is the matrix $\begin{bmatrix}1 & 4\\2 & 5\\3 & 6\end{bmatrix}$

Symmetric Matrices

- A square matrix A is called symmetric if $A = A^t$.
- Thus $A = [a_{ij}]$ is symmetric if $a_{ij} = a_{ji}$ for i and j with $1 \le i \le n$ and $1 \le j \le n$.
- The matrix $\begin{bmatrix}1 & 1 & 0\\1 & 0 & 0\\0 & 1 & 0\end{bmatrix}$ is square
- Symmetric matrices do not change when their rows and columns are interchanged

Zero-One Matrices

- A matrix all of whose entries are either 0 or 1 is called a zero-one matrix.
- Algorithms operating on discrete structures represented by zero-one matrices are based on Boolean arithmetic defined by the following Boolean operations:
  - $b_1 \wedge b_2 = \begin{cases}1 & \text{if } b_1 = b_2 = 1\\0 & \text{otherwise}\end{cases}$
  - $b_1 \vee b_2 = \begin{cases}1 & \text{if } b_1 = 1 \text{ or } b_2 = 1\\0 & \text{otherwise}\end{cases}$

Joint and Meet of Zero-One Matrices

- Definition: Let $A = [a_{ij}]$ and $B = [b_{ij}]$ be an $m \times n$ zero-one matrices.
- The join of A and B is the zero-one matrix with (i,j)th entry $a_{ij} \vee b_{ij}$.
- The join of A and B is denoted by $A \vee B$.
- The meet of of A and B is the zero-one matrix with (i,j)th entry $a_{ij} \wedge b_{ij}$.
- The meet of A and B is denoted by $A \wedge B$.
- Example
  - Find the join and meet of the zero-one matrices
    - $A = \begin{bmatrix}1 & 0 & 1\\0 & 1 & 0\end{bmatrix}$
    - $B = \begin{bmatrix}0 & 1 & 0\\1 & 1 & 0\end{bmatrix}$
  - The joint of A and B is
    - $A \vee B = \begin{bmatrix}1 & 1 & 1\\1 & 1 & 0\end{bmatrix}$
  - The meet of A and B is
    - $A \wedge B = \begin{bmatrix}0 & 0 & 0\\0 & 1 & 0\end{bmatrix}$

Boolean Product of Zero-One Matrices

- Definition:
  - Let $A = [a_{ij}]$ be an $m \times k$ zero-one matrix and $B = [b_{ij}]$ be a $k \times n$ zero-one

matrix.

- The Boolean product of A and B, denoted by $A \odot B$, is the $m \times n$ zero-one matrix with (i,j)th entry $c_{ij} = \left(a_{i1} \wedge b_{1j}\right) \vee \left(a_{i2} \wedge b_{2j}\right) \vee \cdots \vee \left(a_{ik} \wedge b_{kj}\right)$.

- Example: Find the Boolean product of A and B, where
  - $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$
  - $A \odot B = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

Boolean Powers of Zero-One Matrices

- Let A be a square zero-one matrix and let r be a positive integer.
- The rth Boolean power of A is the Boolean product of r factors of A, denoted by $A^{[r]}$.
- Hence, $A^{[r]} = \underbrace{A \odot A \odot \cdots \odot A}_{r \text{ times}}$
- We define $A^{[0]} = I_n$
- The Boolean product is well defined because the Boolean product of matrices is associative
- Example
  - Let $A = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$
  - Find $A^{[n]}$ for all positive integers n
  - $A^{[2]} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$
  - $A^{[3]} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
  - $A^{[4]} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
  - $A^{[5]} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

3.1 Algorithms

Monday, February 19, 2018 9:31 AM

Algorithms

- Definition
  - An algorithm is a finite set of precise instructions for performing a computation or for solving a problem.
- Example: Describe an algorithm for finding the maximum value in a finite sequence of integers.
  - Perform the following steps:
  - Set the temporary maximum equal to the first integer in the sequence.
  - Compare the next integer in the sequence to the temporary maximum.
    - If it is larger than the temporary maximum,
    - set the temporary maximum equal to this integer.
  - Repeat the previous step if there are more integers. If not, stop.
  - When the algorithm terminates, the temporary maximum is the largest integer in the sequence.

Specifying Algorithms

- Algorithms can be specified in different ways.
- Their steps can be described in English or in pseudocode.
- Pseudocode is an intermediate step between an English language description of the steps and a coding of these steps using a programming language.
- The form of pseudocode we use is specified in Appendix 3.
- It uses some of the structures found in popular languages such as C++ and Java.
- Programmers can use the description of an algorithm in pseudocode to construct a program in a particular language.
- Pseudocode helps us analyze the time required to solve a problem using an algorithm, independent of the actual programming language used to implement algorithm.

Properties of Algorithms

- Input
  - An algorithm has input values from a specified set.
- Output
  - From the input values, the algorithm produces the output values from a specified set.
  - The output values are the solution.
- Correctness
  - An algorithm should produce the correct output values for each set of input values.

- Finiteness
  - An algorithm should produce the output after a finite number of steps for any input.
- Effectiveness
  - It must be possible to perform each step of the algorithm correctly and in a finite amount of time.
- Generality
  - The algorithm should work for all problems of the desired form.

Finding the Maximum Element in a Finite Sequence

- The algorithm in pseudocode:

```
procedure max(a_1, a_2, ...., a_n: integers)
   max := a_1
   for i := 2 to n
      if max < a_i then max := a_i
   return max{max is the largest element}
```

- Does this algorithm have all the properties listed on the previous slide?

Some Example Algorithm Problems

- Three classes of problems will be studied in this section.
- Searching Problems
  - finding the position of a particular element in a list.
- Sorting problems
  - putting the elements of a list into increasing order.
- Optimization Problems
  - determining the optimal value of a particular quantity over all possible inputs.

Searching Problems

- The general searching problem is to locate an element x in the list of distinct elements $a_1, a_2, \ldots, a_n$, or determine that it is not in the list.
- The solution to a searching problem is the location of the term in the list that equals x (that is, i is the solution if $x = a_i$) or 0 if x is not in the list.
- For example, a library might want to check to see if a patron is on a list of those with overdue books before allowing him/her to checkout another book.
- We will study two different searching algorithms; linear search and binary search.

Linear Search Algorithm

- The linear search algorithm locates an item in a list by examining elements in the sequence one at a time, starting at the beginning.
- First compare x with a1. If they are equal, return the position 1.
- If not, try a_2. If $x = a_2$, return the position 2.

- Keep going, and if no match is found when the entire list is scanned, return 0.

```
procedure linear search(x:integer,
          a_1, a_2, ...,a_n: distinct integers)
i := 1
while (i ≤ n and x ≠ a_i)
    i := i + 1
if i ≤ n then location := i
else location := 0
return location{location is the subscript of the term that
   equals x, or is 0 if x is not found}
```

Binary Search

- Assume the input is a list of items in increasing order.
- The algorithm begins by comparing the element to be found with the middle element.
  - If the middle element is lower, the search proceeds with the upper half of the list.
  - If it is not lower, the search proceeds with the lower half of the list
  - Repeat this process until we have a list of size 1.
  - If the element we are looking for is equal to the element in the list, the position is returned.
  - Otherwise, 0 is returned to indicate that the element was not found.
- In Section 3.3, we show that the binary search algorithm is much more efficient than linear search.
- Here is a description of the binary search algorithm in pseudocode.

```
procedure binary search(x: integer, a_1,a_2,..., a_n: increasing integers)
i := 1 {i is the left endpoint of interval}
j := n {j is right endpoint of interval}
while i < j
    m := ⌊(i + j)/2⌋
    if x > a_m then i := m + 1
    else j := m
if x = a_i then location := i
else location := 0
return location{location is the subscript i of the term a_i equal to x,
                or 0 if x is not found}
```

Sorting

- To sort the elements of a list is to put them in increasing order (numerical order, alphabetic, and so on).
- Sorting is an important problem because:
  - A nontrivial percentage of all computing resources are devoted to sorting different kinds of lists, especially applications involving large databases of information that need to be presented in a particular order (e.g., by customer, part number etc.).

- An amazing number of fundamentally different algorithms have been invented for sorting. Their relative advantages and disadvantages have been studied extensively.
- Sorting algorithms are useful to illustrate the basic notions of computer science.

- A variety of sorting algorithms are studied in this book; binary, insertion, bubble, selection, merge, quick, and tournament.
- In Section 3.3, we'll study the amount of time required to sort a list using the sorting algorithms covered in this section.

Bubble Sort

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

procedure *bubblesort*($a_1,\ldots,a_n$: real numbers with $n \geq 2$)
  for $i := 1$ to $n-1$
    for $j := 1$ to $n-i$
      if $a_j > a_{j+1}$ **then** interchange a_j and a_{j+1}
{$a_1,\ldots, a_n$ is now in increasing order}

Insertion Sort

- Insertion sort begins with the 2nd element.
- It compares the 2nd element with the 1st and puts it before the first if it is not larger.
- Next the 3rd element is put into the correct position among the first 3 elements.
- In each subsequent pass, the $(n+1)$-th element is put into its correct position among the first $n+1$ elements.
- Linear search is used to find the correct position.

```
procedure insertion sort
 (a_1,...,a_n:
    real numbers with n ≥ 2)
   for j := 2 to n
      i := 1
      while a_j > a_i
         i := i + 1
      m := a_j
      for k := 0 to j − i − 1
         a_{j-k} := a_{j-k-1}
      a_i := m
{Now a_1,...,a_n is in increasing order}
```

Greedy Algorithms

- Optimization problems minimize or maximize some parameter over all possible inputs.
- Among the many optimization problems we will study are:
  - Finding a route between two cities with the smallest total mileage.
  - Determining how to encode messages using the fewest possible bits.
  - Finding the fiber links between network nodes using the least amount of fiber.
- Optimization problems can often be solved using a greedy algorithm, which makes the "best" choice at each step.
- Making the "best choice" at each step does not necessarily produce an optimal solution to the overall problem, but in many instances, it does.
- After specifying what the "best choice" at each step is, we try to prove that this approach always produces an optimal solution, or find a counterexample to show that it does not.
- The greedy approach to solving problems is an example of an algorithmic paradigm, which is a general approach for designing an algorithm.
- We return to algorithmic paradigms in Section 3.3.

Greedy Algorithms: Making Change

- Example
  - Design a greedy algorithm for making change (in U.S. money) of n cents with the following coins
    - quarters (25 cents)
    - dimes (10 cents)
    - nickels (5 cents)
    - pennies (1 cent)
  - using the least total number of coins.

- Idea
  - At each step choose the coin with the largest possible value that does not exceed the amount of change left.
  - If n = 67 cents, first choose a quarter leaving $67-25 = 42$ cents. Then choose another quarter leaving $42 - 25 = 17$ cents
  - Then choose 1 dime, leaving $17 - 10 = 7$ cents.
  - Choose 1 nickel, leaving $7 - 5 = 2$ cents.
  - Choose a penny, leaving one cent. Choose another penny leaving 0 cents.
- Solution
  - G reedy change-making algorithm for n cents.
  - The algorithm works with any coin denominations $c_1, c_2, \ldots, c_r$.

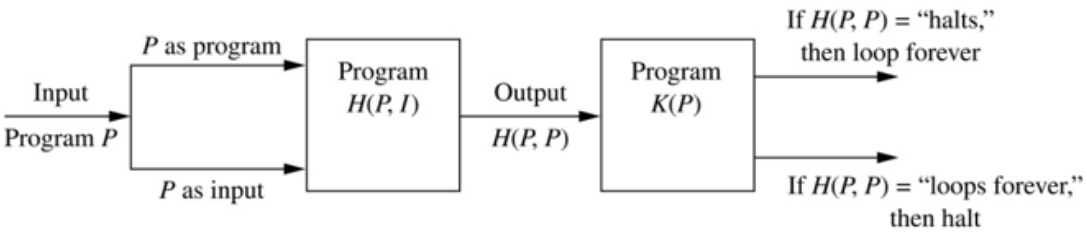
```
procedure change(c1, c2, ..., cr: values of coins, where c1 > c2 > ... > cr;
    n: a positive integer)
for i := 1 to r
    di := 0 [di counts the coins of denomination ci]
    while n ≥ ci
        di := di + 1 [add a coin of denomination ci]
        n = n - ci
[di counts the coins ci]
```

  - For the example of U.S. currency, we may have quarters, dimes, nickels and pennies, with $c_1 = 25, c_2 = 10, c_3 = 5, c_4 = 1$.
- Proving Optimality
  - Lemma 1
    - If n is a positive integer, then n cents in change using quarters, dimes, nickels, and pennies, using the fewest coins possible has at most 2 dimes, 1 nickel, 4 pennies, and cannot have 2 dimes and a nickel.
    - The total amount of change in dimes, nickels, and pennies must not exceed 24 cents.
  - Proof
    - If we had 3 dimes, we could replace them with a quarter and a nickel.
    - If we had 2 nickels, we could replace them with 1 dime.
    - If we had 5 pennies, we could replace them with a nickel.
    - If we had 2 dimes and 1 nickel, we could replace them with a quarter.
    - The allowable combinations, have a maximum value of 24 cents; 2 dimes and 4 pennies.
  - Theorem
    - The greedy change-making algorithm for U.S. coins produces change using the fewest coins possible.
  - Proof
    - Assume there is a positive integer n such that change can be made for n cents using quarters, dimes, nickels, and pennies, with a fewer total number of coins than given by the algorithm.

- Then, $\dot{q} < q$ where $\dot{q}$ is the number of quarters used in this optimal way and q is the number of quarters in the greedy algorithm's solution.
- But this is not possible by Lemma 1, since the value of the coins other than quarters cannot be greater than 24 cents.
- Similarly, by Lemma 1, the two algorithms must have the same number of dimes, nickels, and quarters.

Halting Problem

- Can we develop a procedure that takes as input a computer program along with its input and determines whether the program will eventually halt with that input.
- Solution: Proof by contradiction.
- Assume that there is such a procedure and call it H(P,I).
- The procedure H(P,I) takes as input a program P and the input I to P.
  - H outputs "halt" if it is the case that P will stop when run with input I.
  - Otherwise, H outputs "loops forever."
- Since a program is a string of characters, we can call H(P,P).
- Construct a procedure K(P), which works as follows.
  - If H(P,P) outputs "loops forever" then K(P) halts.
  - If H(P,P) outputs "halt" then K(P) goes into an infinite loop printing "ha" on each iteration.
- Now we call K with K as input, i.e. K(K).
  - If the output of H(K,K) is "loops forever" then K(K) halts. A Contradiction.
  - If the output of H(K,K) is "halts" then K(K) loops forever. A Contradiction.
- Therefore, there cannot be a procedure that can decide whether or not an arbitrary program halts.
- The halting problem is unsolvable.



3.2 The Growth of Functions

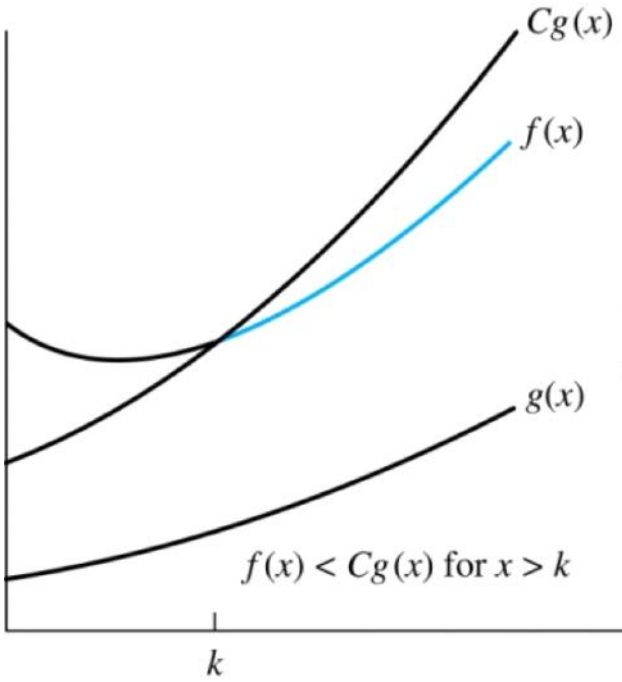
Friday, February 23, 2018 8:55 AM

The Growth of Functions

- In both computer science and in mathematics, there are many times when we care about how fast a function grows.
- In computer science, we want to understand how quickly an algorithm can solve a problem as the size of the input grows.
  - We can compare the efficiency of two different algorithms for solving the same problem.
  - We can also determine whether it is practical to use a particular algorithm as the input grows.
  - We'll study these questions in Section 3.3.
- Two of the areas of mathematics where questions about the growth of functions are studied are:
  - number theory (covered in Chapter 4)
  - combinatorics (covered in Chapters 6 and 8)

Big-O Notation

- Let f and g be functions from the set of integers or the set of real numbers to the set of real numbers.
- We say that f(x) is O(g(x)) if there are constants C and k such that
  - $|f(x)| \leq C|g(x)|$ whenever $x > k$.
- This is read as "$f(x)$ is big-O of g(x)" or "g asymptotically dominates f."
- The constants C and k are called witnesses to the relationship $f(x)$ is $O(g(x))$.
- Only one pair of witnesses is needed.



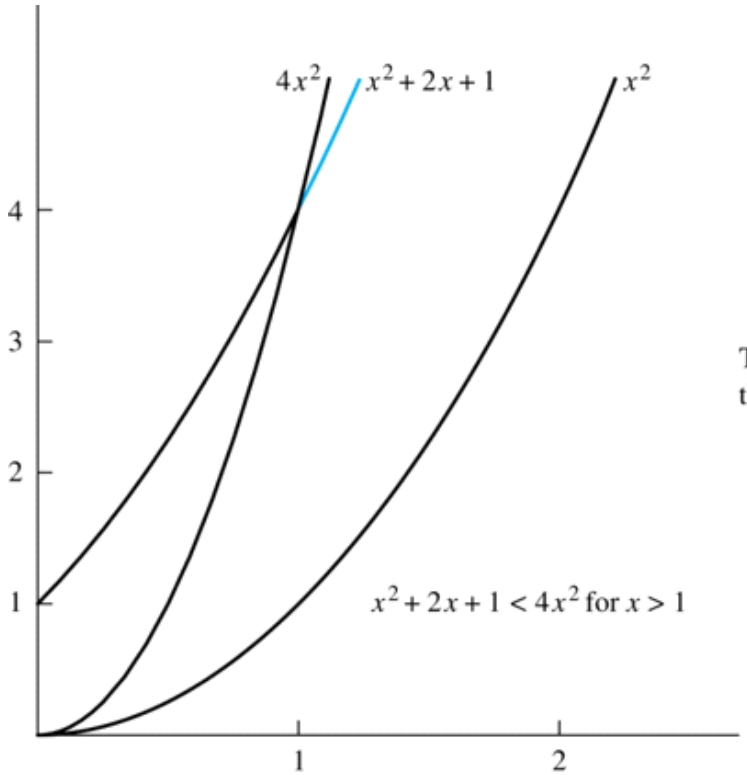
The part of the graph of $f(x)$ that satisfies $f(x) < Cg(x)$ is shown in color.

Some Important Points about Big-O Notation

- If one pair of witnesses is found, then there are infinitely many pairs.
- We can always make the k or the C larger and still maintain the inequality $|f(x)| \leq C|g(x)|$.
- If $f(x)$ is $O(g(x))$ and $g(x)$ is $O(h(x))$ then $f(x)$ is $O(h(x))$
- You may see "$f(x) = O(g(x))$" instead of "$f(x)$ is $O(g(x))$."
- But this is an abuse of the equals sign since the meaning is that
  there is an inequality relating the values of f and g, for sufficiently large values of x.
- It is ok to write $f(x) \in O(g(x))$, because $O(g(x))$ represents the set of functions that are $O(g(x))$.
- Usually, we will drop the absolute value sign since
  we will always deal with functions that take on positive values.

Using the Definition of Big-O Notation

- Example: Show that $f(x) = x^2 + 2x + 1$ is $O(x^2)$.
  - Since when $x > 1$, $x < x^2$ and $1 < x^2$
  - $0 \leq x^2 + 2x + 1 \leq x^2 + 2x^2 + x^2 - 4x^2$
  - Can take C = 4 and k = 1 as witnesses to show that $f(x)$ is $O(x^2)$
  - Alternatively, when $x > 2$, we have $2x \leq x^2$ and $1 < x^2$.
  - Hence, $0 \leq x^2 + 2x + 1 \leq x^2 + x^2 + x^2 = 3x^2$ when $x > 2$.
  - Can take C = 3 and k = 2 as witnesses instead.



The part of the graph of $f(x) = x^2 + 2x + 1$ that satisfies $f(x) < 4x^2$ is shown in blue.

- Example: Show that $7x^2$ is $O(x^3)$
  - When $x > 7$, $7x\text{^}2 < x^3$.
  - Take C =1 and $k = 7$ as witnesses to establish that $7x^2$ is $O(x^3)$.
  - (Would C = 7 and k = 1 work?) Yes

- Example: Show that n^2 is not $O(n)$
  - Suppose there are constants C and k for which $n^2 \leq Cn$, whenever $n > k$.
  - Then $n \leq C$ must hold for all $n > k$. A contradiction!

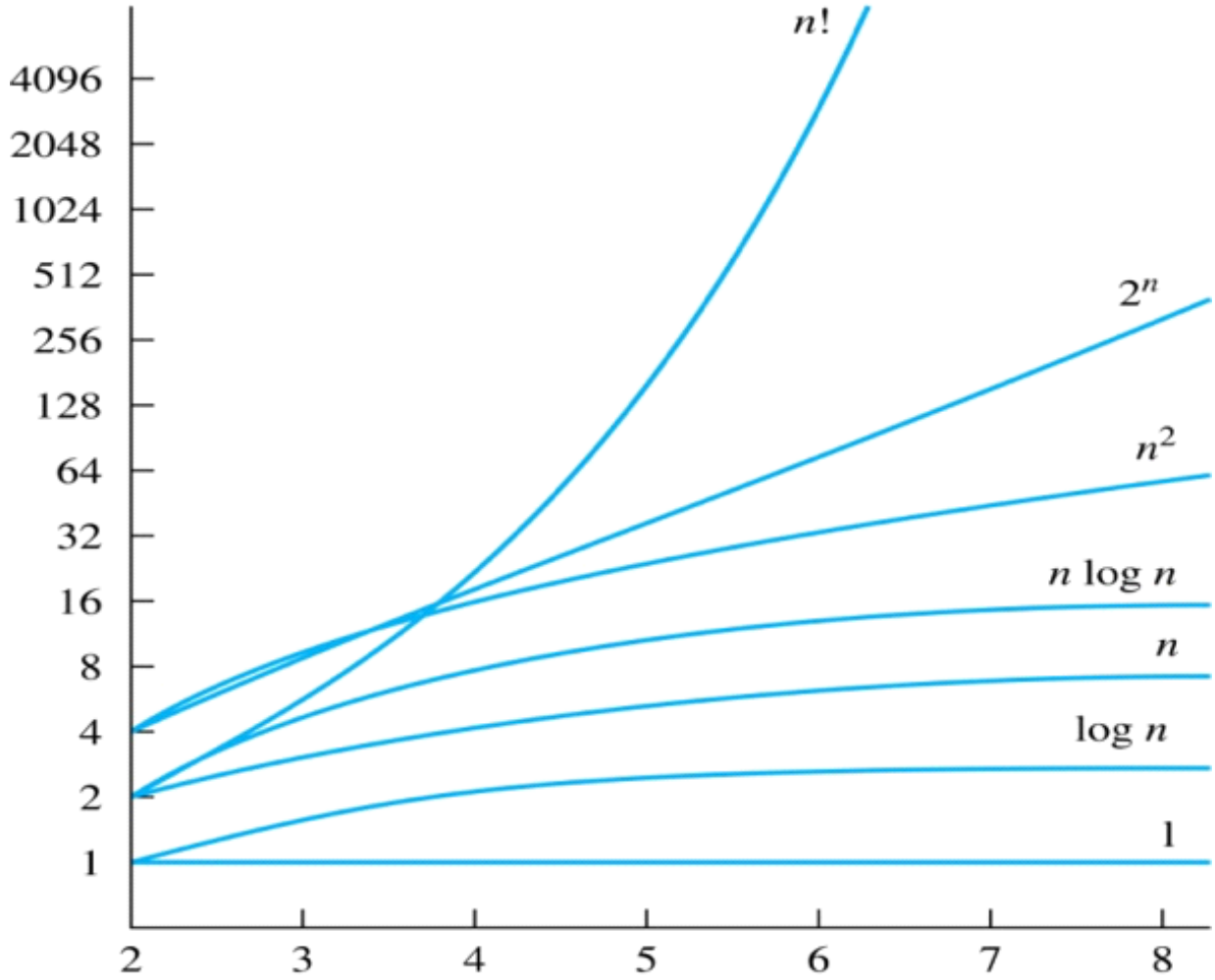
Big-O Estimates for Polynomials

- Let $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ where $a_0, a_1, \ldots a_n$ are real numbers with $a_n \neq 0$
- Then $f(x)$ is $O(x_n)$.
- $|f(x)| = |a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0|$
- $\leq |an|xn + |an-1|\, xn-1 + \cdots + |a1|x1 + |a0|$
- $= x_n \left(|a_n| + \frac{|a_{n-1}|}{x} + \cdots + \frac{|a1|}{x_{n-1}} + \frac{|a_0|}{x_n} \right)$
- $\leq x_n \left(|a_n| + |a_{n-1}| + \cdots + |a_1| + |a_0| \right)$
- Take $C = |a_n| + |a_{n-1}| + \cdots + |a_0|$ and $k = 1$.
- Then $f(x)$ is $O(x^n)$.
- The leading term $a_n x^n$ of a polynomial dominates its growth.

Big-O Estimates for Some Important Functions

- Example: Use big-O notation to estimate the sum of the first n positive integers.
  - $1 + 2 + \cdots + n \leq n + n + \cdots + n = n^2$
  - $1 + 2 + \cdots + n$ is $O(n^2)$ taking $C = 1$ and $k = 1$
- Example: Use big-O notation to estimate the factorial function $f(n) = n! = 1 \times 2 \times \cdots \times n$
  - $n! = 1 \times 2 \times \cdots \times n \leq n \times n \times \cdots \times n^n$
  - $n!$ is $O(n^n)$ taking $C = 1$ and $k = 1$
- Example: Use big-O notation to estimate $\log n!$
  - Given that $n! < n^n$
  - then $\log n! \leq n \cdot \log n$.
  - Hence, $\log n!$ is $O(n \log n)$ taking $C = 1$ and $k = 1$.

Display of Growth of Functions



Combinations of Functions

- If $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$ then
  - $(f_1 + f_2)(x)$ is $O(\max\{|g_1(x)|, |g_2(x)|\})$.
- If $f_1(x)$ and $f_2(x)$ are both $O(g(x))$ then
  - $(f_1 + f_2)(x)$ is $O(g(x))$.
- If $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$ then
  - $(f_1 f_2)(x)$ is $O(g_1(x)g_2(x))$.

Big-Omega Notation

- Let f and g be functions from the set of integers/real numbers to the set of real numbers.
- We say that $f(x)$ is $\Omega(g(x))$ if there are constants C and k such that $|f(x)| \geq C|g(x)|$ when $x > k$
- We say that "$f(x)$ is big-Omega of $g(x)$."
- Big-O gives an upper bound on the growth of a function, while Big-Omega gives a lower bound.
- Big-Omega tells us that a function grows at least as fast as another.
- $f(x)$ is $\Omega(g(x))$ if and only if $g(x)$ is $O(f(x))$.
- This follows from the definitions.
- Example: Show $f(x) = 8x^3 + 5x^2 + 7$ is $\Omega(g(x))$ where $g(x) = x^3$

- $f(x) = 8x^3 + 5x^2 + 7 \geq 8x^3$ for all positive real numbers x
- Is it also the case that $g(x) = x^3$ is $O(8x^3 + 5x^2 + 7)$? Yes

Big-Theta Notation

- Let f and g be functions from the set of integers/real numbers to the set of real numbers.
- The function $f(x)$ is $\Theta(g(x))$ if $f(x)$ is $O(g(x))$ and $f(x)$ is $\Omega(g(x))$
- We say that
  - "f is big-Theta of $g(x)$"
  - "$f(x)$ is of order $g(x)$"
  - "$f(x)$ and $g(x)$ are of the same order."
- $f(x)$ is $\Theta(g(x))$ if and only if
  there exists constants C_1, C_2 and k such that $C_1 g(x) < f(x) < C_2 g(x)$ if $x > k$.
- This follows from the definitions of big-O and big-Omega.
- When $f(x)$ is $\Theta(g(x))$ it must also be the case that $g(x)$ is $\Theta(f(x))$
- Note that $f(x)$ is $\Theta(g(x))$ if and only if it is the case that $f(x)$ is $O(g(x))$ and $g(x)$ is $O(f(x))$.
- Sometimes writers are careless and write as if big-O notation has the same meaning as big-Theta.

Examples of Big-Theta Notation

- Example 1: Show that the sum of the first n positive integers is $\Theta(n^2)$
  - Let $f(n) = 1 + 2 + \cdots + n$.
  - We have already shown that $f(n)$ is $O(n^2)$.
  - To show that $f(n)$ is $\Omega(n^2)$, we need a positive constant C such that
    $f(n) > Cn^2$ for sufficiently large n.
  - Summing only the terms greater than $\frac{n}{2}$ we obtain the inequality

$$\begin{aligned} 1 + 2 + \cdots + n &\geq \left\lceil \frac{n}{2} \right\rceil + \left(\left\lceil \frac{n}{2} \right\rceil + 1 \right) + \cdots + n \\ &\geq \left\lceil \frac{n}{2} \right\rceil + \left\lceil \frac{n}{2} \right\rceil + \cdots + \left\lceil \frac{n}{2} \right\rceil \\ &= \left(n - \left\lceil \frac{n}{2} \right\rceil + 1 \right) \left\lceil \frac{n}{2} \right\rceil \\ &\geq \left(\frac{n}{2} \right) \left(\frac{n}{2} \right) = \frac{n^2}{4} \end{aligned}$$

  - Taking $C = \frac{1}{4}$, $f(n) > Cn^2$ for all positive integers n.
  - Hence, $f(n)$ is $\Omega(n^2)$, and we can conclude that $f(n)$ is $\Theta(n^2)$.
- Example 2: Show that $f(x) = 3x^2 + 8x \log x$ is $\Theta(x^2)$
  - $f(x) = 3x^2 + 8x \log x \leq 11x^2$ for x > 1,
  - since $0 \leq 8x \log x \leq 8x^2$.
  - Hence, $3x^2 + 8x \log x$ is $O(x^2)$.
  - x^2 is clearly $O(3x^2 + 8x \log x)$
  - Hence, $3x^2 + 8x \log x$ is $\Theta(x^2)$

3.3 Complexity of Algorithms

Monday, February 26, 2018 9:03 AM

The Complexity of Algorithms

- Given an algorithm, how efficient is this algorithm for solving a problem given input of a particular size?
- To answer this question, we ask:
    - How much time does this algorithm use to solve a problem?
    - How much computer memory does this algorithm use to solve a problem?
- When we analyze the time the algorithm uses to solve the problem given input of a particular size, we are studying the time complexity of the algorithm.
- When we analyze the computer memory the algorithm uses to solve the problem given input of a particular size, we are studying the space complexity of the algorithm.
- In this course, we focus on time complexity.
- The space complexity of algorithms is studied in later courses.
- We will measure time complexity in terms of the number of operations an algorithm uses and we will use big-O and big-Theta notation to estimate the time complexity.
- We can use this analysis to see whether it is practical to use this algorithm to solve problems with input of a particular size.
- We can also compare the efficiency of different algorithms for solving the same problem.
- We ignore implementation details because it is extremely complicated to consider them.

Time Complexity

- To analyze the time complexity of algorithms, we determine the number of operations, such as comparisons and arithmetic operations (addition, multiplication, etc.).
- We can estimate the time a computer may actually use to solve a problem using the amount of time required to do basic operations.
- We ignore minor details, such as the “house keeping” aspects of the algorithm.
- We will focus on the worst-case time complexity of an algorithm.
- This provides an upper bound on the number of operations an algorithm uses to solve a problem with input of a particular size.
- It is usually much more difficult to determine the average case time complexity of an algorithm.
- This is the average number of operations an algorithm uses to solve a problem over all inputs of a particular size.

Complexity Analysis of Algorithms

- Example: Describe the time complexity of the algorithm for finding the maximum element in a finite sequence.

```
procedure max(a1, a2, ...., an: integers)
   max := a1
   for i := 2 to n
      if max < ai then max := ai
   return max{max is the largest element}
```

- Count the number of comparisons.
- The max $< a_i$ comparison is made $n - 1$ times.
- Each time i is incremented, a test is made to see if $i \leq n$.
- One last comparison determines that $i > n$.
- Exactly $2(n-1) + 1 = 2n - 1$ comparisons are made.
- Hence, the time complexity of the algorithm is $\Theta(n)$.

- Example: Determine the time complexity of the linear search algorithm.

```
procedure linear search(x:integer,
          a1, a2, ...,an: distinct integers)
i := 1
while (i ≤ n and x ≠ ai)
    i := i + 1
if i ≤ n then location := i
else location := 0
return location{location is the subscript of the term that equals x, or is 0 if
     x is not found}
```

- Count the number of comparisons.
- At each step two comparisons are made; $i \leq n$ and $x \neq a_i$.
- To end the loop, one comparison $i \leq n$ is made.
- After the loop, one more $i \leq n$ comparison is made.
- If $x = a_i$, $2i + 1$ comparisons are used.
- If x is not on the list, $2n + 1$ comparisons are made and then an additional comparison is used to exit the loop.
- So, in the worst case $2n + 2$ comparisons are made.
- Hence, the complexity is $\Theta(n)$.

- Example: Describe the average case performance of the linear search algorithm.
- Assume the element is in the list and that the possible positions are equally likely.
- By the argument on the previous slide, if $x = a_i$, the number of comparisons is $2i + 1$
- $\dfrac{3 + 5 + 7 + \cdots + (2n+1)}{n} = \dfrac{2(1 + 2 + 3 + \cdots + n) + n}{n} = \dfrac{2\left[\dfrac{n(n+1)}{2}\right]}{n} + 1 = n + 2$
- Hence, the average-case complexity of linear search is $\Theta(n)$.

- Example: Describe the time complexity of binary search in terms of the number of comparisons used.

```
procedure binary search(x: integer, a_1,a_2,..., a_n: increasing integers)
 i := 1 {i is the left endpoint of interval}
 j := n {j is right endpoint of interval}
 while i < j
      m := ⌊(i + j)/2⌋
      if x > a_m then i := m + 1
      else j := m
 if x = a_i then location := i
 else location := 0
 return location{location is the subscript i of the term a_i equal to x, or 0 if x is not found}
```

  - Assume (for simplicity) n $= 2^k$ elements. Note that $k = \log n$.
  - Two comparisons are made at each stage; $i < j$, and $x > a_m$.
  - At the first iteration the size of the list is $2k$ and after the first iteration it is $2k - 1$.
  - Then $2k - 2$ and so on until the size of the list is $2^1 = 2$.
  - At the last step, a comparison tells us that the size of the list is the size is $2^0 = 1$ and the element is compared with the single remaining element.
  - Hence, at most $2k + 2 = 2\log n + 2$ comparisons are made.
  - Therefore, the time complexity is $\Theta(\log n)$, better than linear search.

- Example: What is the worst-case complexity of bubble sort in terms of the number of comparisons made?

```
procedure bubblesort(a_1,...,a_n: real numbers
                     with n ≥ 2)
   for i := 1 to n − 1
      for j := 1 to n − i
         if a_j > a_{j+1} then interchange a_j and a_{j+1}
{a_1,..., a_n is now in increasing order}
```

  - A sequence of $n - 1$ passes is made through the list.
  - On each pass $n - i$ comparisons are made.
  - $(n-1) + (n-2) + \cdots + 2 + 1 = \frac{n(n-1)}{2} = \frac{1}{2}n^2 - \frac{1}{2}n$
  - The worst-case complexity of bubble sort is $\Theta(n^2)$

- Example: What is the worst-case complexity of insertion sort in terms of the number of comparisons made?

```
procedure insertion sort(a_1,...,a_n:
            real numbers with n ≥ 2)
    for j := 2 to n
        i := 1
        while a_j > a_i
            i := i + 1
        m := a_j
        for k := 0 to j − i − 1
            a_{j-k} := a_{j-k-1}
        a_i := m
```

- The total number of comparisons are
- $2 + 3 + \cdots + n = \frac{n(n-1)}{2} - 1$
- Therefore the complexity is $\Theta(n^2)$

- Example: How many additions of integers and multiplications of integers are used by the matrix multiplication algorithm to multiply two $n \times n$ matrices.

```
procedure matrix multiplication(A, B: matrices)
    for i := 1 to m
        for j := 1 to n
            c_ij := 0
            for q := 1 to k
                c_ij := c_ij + a_iq b_qj
return C{C = [c_ij] is the product of A and B}
```

- There are n^2 entries in the product.
- Finding each entry requires n multiplications and $n - 1$ additions.
- Hence, n^3 multiplications and $n^2(n-1)$ additions are used.
- Hence, the complexity of matrix multiplication is $O(n^3)$.

Understanding the Complexity of Algorithms

TABLE 1 Commonly Used Terminology for the Complexity of Algorithms.

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

Complexity of Problems

- Tractable Problem
  - There exists a polynomial time algorithm to solve this problem.
  - These problems are said to belong to the Class P.
- Intractable Problem
  - There does not exist a polynomial time algorithm to solve this problem
- Unsolvable Problem
  - No algorithm exists to solve this problem, e.g., halting problem.
- Class NP
  - Solution can be checked in polynomial time.
  - But no polynomial time algorithm has been found for finding a solution to problems in this class.
- NP Complete Class
  - If you find a polynomial time algorithm for one member of the class,
  - it can be used to solve all the problems in the class.

P Versus NP Problem

- The P versus NP problem asks whether the class $P = NP$?
- Are there problems whose solutions can be checked in polynomial time, but can not be solved in polynomial time?
- Note that just because no one has found a polynomial time algorithm is different from showing that the problem can not be solved by a polynomial time algorithm.
- If a polynomial time algorithm for any of the problems in the NP complete class were found, then that algorithm could be used to obtain a polynomial time algorithm for every problem in the NP complete class.

- Satisfiability (in Section 1.3) is an NP complete problem.
- It is generally believed that P≠NP since no one has been able to find a polynomial time algorithm for any of the problems in the NP complete class.
- The problem of P versus NP remains one of the most famous unsolved problems in mathematics (including theoretical computer science).
- The Clay Mathematics Institute has offered a prize of $1,000,000 for a solution.

4.1 Divisibility and Modular Arithmetic

Wednesday, February 28, 2018 8:59 AM

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 $3 \mid 12$.
  - $3 \mid 7$ is false
  - $3 \mid 12$ is true

Properties of Divisibility

- Theorem 1: Let a, b, and c be integers, where $a \neq 0$.
  - If $a \mid b$ and $a \mid c$, then $a \mid (b + c)$;
  - If $a \mid b$, then $a \mid bc$ for all integers c;
  - 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, $b + c = as + at = a(s + t)$.
  - Hence, $a \mid (b + c)$
- 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.

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 $0 \leq r < d$, such that
  - $a = dq + r$

- d is called the divisor.
- a is called the dividend.
- q is called the quotient.
- r is called the remainder.

- Example: What are the quotient and reminder when 101 is divided by 11?
  - $101 = 11 \cdot 9 + 2$
  - Thus the quotient is 9, and the remainder is 2
  - $101 \text{ div } 11 = 9$
  - $101 \bmod 11 = 2$
- Example: What are the quotient and reminder when -11 is divided by 3?
  - $-11 = 3 \cdot (-4) + 1$

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 $a \not\equiv b \pmod{m}$
- Determine whether 17 is congruent to 5 modulo 6
  - $17 \equiv 5 \pmod{6}$ because 6 divides $17 - 5 = 12$.
- Determine whether 24 and 14 are congruent modulo 6.
  - $24 \not\equiv 14 \pmod{6}$ since $24 - 14 = 10$ is not divisible by 6.

The Relationship between (mod m) and mod m Notations

- The use of "mod" in $a \equiv b \pmod{m}$ and $a \bmod m = b$ are different.
  - $a \equiv b \pmod{m}$ is a relation on the set of integers.
  - In $a \bmod m = b$, 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
  - $a \bmod m = b \bmod m$.

More on Congruence

- 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 $a \equiv b \pmod{m}$.

Congruence of Sums and Products

- Theorem 5
  - Let m be a positive integer.
  - If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then
    - $a + c \equiv b + d \pmod{m}$
    - $ac \equiv bd \pmod{m}$

- Proof:
  - Because $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$
  - by Theorem 4 there are integers s and t with $b = a + sm$ and $d = c + tm$.
  - Therefore,
    - $b + d = (a + sm) + (c + tm) = (a + c) + m(s + t)$ and
    - $b\,d = (a + sm)(c + tm) = ac + m(at + cs + stm)$.
  - Hence, $a + c \equiv b + d \pmod{m}$ 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
  - $18 = 7 + 11 \equiv 2 + 1 = 3 \pmod{5}$
  - $77 = 7 \cdot 11 \equiv 2 \cdot 1 = 2 \pmod{5}$

Algebraic Manipulation of Congruence

- 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.
  - 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 $7 \not\equiv 4 \pmod{6}$.

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

Arithmetic Modulo m

- Definitions
  - Let $\mathbb{Z}_m$ be the set of nonnegative integers less than m: {0,1, …., m−1}
  - The operation $+_m$ is defined as $a +_m b = (a + b) \bmod m$.
  - This is addition modulo m.
  - The operation $\cdot_m$ is defined as $a \cdot_m b = (a \cdot b) \bmod m$.
  - This is multiplication modulo m.
- Example: Find $7 +_{11} 9$ and $7 \cdot_{11} 9$.
  - $7 +_{11} 9 = (7 + 9) \bmod 11 = 16 \bmod 11 = 5$
  - $7 \cdot_{11} 9 = (7 \cdot 9) \bmod 11 = 63 \bmod 11 = 8$
- The operations $+_m$ and $\cdot_m$ satisfy many of the same properties as ordinary addition and multiplication.
  - Closure
    - If a and b belong to $\mathbb{Z}_m$, then $a +_m b$ and $a \cdot_m b$ belong to $\mathbb{Z}_m$.
  - Associativity
    - If a, b, and c belong to $\mathbb{Z}_m$, then
    - $(a +_m b) +_m c = a +_m (b +_m c)$
    - $(a \cdot_m b) \cdot_m c = a \cdot_m (b \cdot_m c)$.
  - Commutativity
    - If a and b belong to $\mathbb{Z}_m$, then
    - $a +_m b = b +_m a$
    - $a \cdot_m b = b \cdot_m a$.
  - Identity elements
    - The elements 0 and 1 are identity elements for addition and multiplication modulo m, respectively.
    - If a belongs to $\mathbb{Z}_m$, then $a +_m 0 = a$ and $a \cdot_m 1 = a$.
  - Additive inverses
    - If $a \neq 0$ belongs to $\mathbb{Z}_m$
    - then $m - a$ is the additive inverse of a modulo m
    - and 0 is its own additive inverse.

        - $a +_m (m - a) = 0$
        - $0 +_m 0 = 0$
    - Distributivity
        - If a, b, and c belong to $\mathbb{Z}_m$, then
        - $a \cdot_m (b +_m c) = (a \cdot_m b) +_m (a \cdot_m c)$
        - $(a +_m b) \cdot_m c = (a \cdot_m c) +_m (b \cdot_m c)$.
- Multiplicative inverses have not been included since they do not always exist.
- For example, there is no multiplicative inverse of 2 modulo 6.
- Using the terminology of abstract algebra
    - $\mathbb{Z}_m$ with $+_m$ is a commutative group
    - $\mathbb{Z}_m$ with $+_m$ and $\cdot_m$ is a commutative ring

4.2 Integer Representations and Algorithms

Friday, March 2, 2018 9:13 AM

Representations of Integers

- In the modern world, we use decimal, or base 10, notation to represent integers.
- For example when we write 965, we mean $9 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0$.
- We can represent numbers using any base b, where b is a positive integer greater than 1.
- The bases b = 2 (binary), b = 8 (octal), and b= 16 (hexadecimal) are important for computing and communications
- The ancient Mayans used base 20 and the ancient Babylonians used base 60.

Base b Representations

- We can use positive integer b greater than 1 as a base, because of this theorem:
- Theorem 1
  - Let b be a positive integer greater than 1.
  - Then if n is a positive integer, it can be expressed uniquely in the form:
  - $n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0$
  - where k is a nonnegative integer, $a_0, a_1, \ldots, a_k$ are nonnegative integers less than b
  - and $ak \neq 0$. The a_j $(j = 0, \ldots, k)$ are called the base-b digits of the representation.
- The representation of n given in Theorem 1 is called the base b expansion of n
- and is denoted by $(a_k a_{k-1} \ldots a_1 a_0)_b$.
- We usually omit the subscript 10 for base 10 expansions.

Binary Expansions

- Most computers represent integers and do arithmetic with binary expansions of integers.
- In these expansions, the only digits used are 0 and 1.
- What is the decimal expansion of the integer that has $(1\ 0101\ 1111)_2$ as its binary expansion?
  - $(1\ 0101\ 1111)_2 = 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 = 351$
- What is the decimal expansion of the integer that has $(11011)2$ as its binary expansion?
  - $(11011)_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 = 27$

Octal Expansions

- The octal expansion (base 8) uses the digits {0,1,2,3,4,5,6,7}.
- What is the decimal expansion of the number with octal expansion $(7016)_8$?
  - $7 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8^1 + 6 = 3598$
- What is the decimal expansion of the number with octal expansion $(111)_8$?
  - $1 \cdot 8^2 + 1 \cdot 8^1 + 1 = 64 + 8 + 1 = 73$

Hexadecimal Expansions

- The hexadecimal expansion needs 16 digits, but our decimal system provides only 10.
- So letters are used for the additional symbols.
- The hexadecimal system uses the digits {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}.
- The letters A through F represent the decimal numbers 10 through 15.
- What is the decimal expansion of the number with hexadecimal expansion $(2AE0B)_{16}$?
  - $2 \cdot 16^4 + 10 \cdot 16^3 + 14 \cdot 16^2 + 0 \cdot 16^1 + 11 = 175627$

  What is the decimal expansion of the number with hexadecimal expansion $(E5)_{16}$?
  - $14 \cdot 16^1 + 5 = 224 + 5 = 229$

Base Conversion

- To construct the base b expansion of an integer n:
- Divide n by b to obtain a quotient and remainder.
  - $n = bq_0 + a_0, 0 \leq a_0 < b$
- The remainder, a_0 , is the rightmost digit in the base b expansion of n. Next, divide q_0 by b.
  - $q_0 = bq_1 + a_1, 0 \leq a_1 < b$
- The remainder, a_1, is the second digit from the right in the base b expansion of n.
- Continue by successively dividing the quotients by b
- obtaining the additional base b digits as the remainder.
- The process terminates when the quotient is 0.
- Algorithm

procedure *base b expansion*(n, b: positive integers with $b > 1$)
$q := n$
$k := 0$
while $(q \neq 0)$
    $a_k := q$ **mod** b
    $q := q$ **div** b
    $k := k + 1$
return$(a_{k-1}, \ldots, a_1, a_0)$ {$(a_{k-1} \ldots a_1 a_0)_b$ is base b expansion of n}

  - q represents the quotient obtained by successive divisions by b, starting with q = n.
  - The digits in the base b expansion are the remainders of the division given by q mod b.
  - The algorithm terminates when q = 0 is reached.
- Find the octal expansion of $(12345)_{10}$
  - Successively dividing by 8 gives:
  - $12345 = 8 \cdot 1543 + 1$
  - $1543 = 8 \cdot 192 + 7$
  - $192 = 8 \cdot 24 + 0$
  - $24 = 8 \cdot 3 + 0$

- $3 = 8 \cdot 0 + 3$
- The remainders are the digits from right to left yielding $(30071)_8$.

Comparison of Hexadecimal, Octal, and Binary Representations

- Each octal digit corresponds to a block of 3 binary digits.
- Each hexadecimal digit corresponds to a block of 4 binary digits.
- So, conversion between binary, octal, and hexadecimal is easy.

TABLE 1 Hexadecimal, Octal, and Binary Representation of the Integers 0 through 15.

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Octal	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
Binary	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111

Conversion Between Binary, Octal, and Hexadecimal Expansions

- Find the octal and hexadecimal expansions of $(11\ 1110\ 1011\ 1100)_2$.
- To convert to octal, we group the digits into blocks of three
- $(011\ 111\ 010\ 111\ 100)_2$, adding initial 0s as needed.
- The blocks from left to right correspond to the digits 3,7,2,7, and 4.
- Hence, the solution is $(37274)_8$.
- To convert to hexadecimal, we group the digits into blocks of four
- $(0011\ 1110\ 1011\ 1100)_2$, adding initial 0s as needed.
- The blocks from left to right correspond to the digits 3,E,B, and C.
- Hence, the solution is $(3EBC)_{16}$.

Binary Addition of Integers

- Algorithms for performing operations with integers using their binary expansions are important as computer chips work with binary numbers. Each digit is called a bit.

procedure *add*(a, b: positive integers)
{the binary expansions of a and b are $(a_{n-1},a_{n-2},\ldots,a_0)_2$ and $(b_{n-1},b_{n-2},\ldots,b_0)_2$, respectively}
$c := 0$
for $j := 0$ to $n - 1$
  $d := \lfloor (a_j + b_j + c)/2 \rfloor$
  $s_j := a_j + b_j + c - 2d$
  $c := d$
$s_n := c$
return $(s_0,s_1,\ldots,s_n)$ {the binary expansion of the sum is $(s_n,s_{n-1},\ldots,s_0)_2$}

- The number of additions of bits used by the algorithm to add two n-bit integers is $O(n)$.

4.3 Primes and Greatest Common Divisors

Monday, March 5, 2018 8:53 AM

Primes

- Definition
  - A positive integer p greater than 1 is called prime if the only positive factors 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.

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 nondecreasing size.
- Examples
  - $100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 \cdot 5^2$
  - $641 = 641$
  - $999 = 3 \cdot 3 \cdot 3 \cdot 37 = 3^3 \cdot 37$
  - $1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}$

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.
- Delete all the integers, other than 2, divisible by 2.
- Delete all the integers, other than 3, divisible by 3.
- Next, delete all the integers, other than 5, divisible by 5.
- Next, delete all the integers, other than 7, divisible by 7.
- Since all the remaining integers are not divisible by any of the previous integers, other than 1
- The primes are: {2,3,5,7,11,15,1719,23,29,31,37,41,43,47,53, 59,61,67,71,73,79,83,89, 97}

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.

Infinitude of Primes

- Theorem
  - There are infinitely many primes. (Euclid)
- Proof
  - Assume finitely many primes: $p_1, p_2, \ldots, p_n$
  - Let $q = p_1 p_2 \cdots p_n + 1$
  - Either q is prime or by the fundamental theorem of arithmetic it is a product of primes.
  - But none of the primes p_j divides q since if $p_j \mid q$, then
  - p_j divides $q - p_1 p_2 \cdots p_n = 1$.
  - Hence, there is a prime not on the list $p_1, p_2, \ldots, p_n$.

- It is either q, or if q is composite, it is a prime factor of q.
- This contradicts the assumption that $p_1, p_2, \ldots, p_n$ are all the primes.
- Consequently, there are infinitely many primes.

Mersene Primes

- Prime numbers of the form $2^p - 1$, where p is prime, are called Mersene primes.
- $2^2 - 1 = 3, 2^3 - 1 = 7, 2^5 - 1 = 37$, and $2^7 - 1 = 127$ are Mersene primes.
- $2^{11} - 1 = 2047$ is not a Mersene 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 Mersene primes.
- As of mid-2011, 47 Mersene primes were known, the largest is $2^{43,112,609} - 1$, which has nearly 13 million decimal digits.

Distribution of Primes

- Mathematicians have been interested in the distribution of prime numbers among the positive integers.
- In the nineteenth century, the prime number theorem was proved which gives an asymptotic estimate for the number of primes not exceeding x.
- Prime Number Theorem
  - The ratio of the number of primes not exceeding x
  - and $\frac{x}{\ln x}$ approaches 1 as x grows without bound.
  - The theorem tells us that the number of primes not exceeding x, can be approximated by $\frac{x}{\ln x}$.
  - The odds that a randomly selected positive integer less than n is prime are approximately $\left(\frac{n}{\ln n}\right)/n = \frac{1}{\ln n}$.

Primes and Arithmetic Progressions

- Euclid's proof that there are infinitely many primes can be easily adapted to show that there are infinitely many primes in the following $4k + 3$ $(k = 1,2,3 \ldots)$
- In the 19th century G. Lejuenne Dirchlet showed that
  - every arithmetic progression $ka + b$, $(k = 1,2,3 \ldots)$
  - where a and b have no common factor greater than 1 contains infinitely many primes.
- Are there long arithmetic progressions made up entirely of primes?
  - 5,11, 17, 23, 29 is an arithmetic progression of five primes.
  - 199, 409, 619, 829, 1039,1249,1459,1669,1879,2089 is an arithmetic progression of ten primes.
- In the 1930s, Paul Erdős conjectured that for every positive integer n greater than 1
- there is an arithmetic progression of length n made up entirely of primes.
- This was proven in 2006, by Ben Green and Terrence Tau.

Generating Primes

- The problem of generating large primes is of both theoretical and practical interest.

- We will see that finding large primes with hundreds of digits is important in cryptography.
- So far, no useful closed formula that always produces primes has been found.
- There is no simple function $f(n)$ such that $f(n)$ is prime for all positive integers n.
- But $f(n) = n^2 - n + 41$ is prime for all integers 1,2,..., 40.
- Because of this, we might conjecture that $f(n)$ is prime for all positive integers n.
- But $f(41) = 41^2$ is not prime.
- More generally, there is no polynomial with integer coefficients such that $f(n)$ is prime for all positive integers n.
- Fortunately, we can generate large integers which are almost certainly primes. See Chapter 7.

Conjectures about Primes

- Even though primes have been studied extensively for centuries, many conjectures about them are unresolved, including:
- Goldbach's Conjecture
  - Every even integer n, $n > 2$, is the sum of two primes.
  - It has been verified by computer for all positive even integers up to $1.6 \cdot 1018$.
  - The conjecture is believed to be true by most mathematicians.
- There are infinitely many primes of the form $n^2 + 1$, where n is a positive integer.
  - But it has been shown that there are infinitely many primes of the form $n^2 + 1$
  - where n is a positive integer or the product of at most two primes.
- The Twin Prime Conjecture
  - The twin prime conjecture is that there are infinitely many pairs of twin primes.
  - Twin primes are pairs of primes that differ by 2.
  - Examples are 3 and 5, 5 and 7, 11 and 13, etc.
  - The current world's record for twin primes (as of mid 2011) consists of numbers
  - $65{,}516{,}468{,}355 \cdot 23^{33{,}333} \pm 1$, which have 100,355 decimal digits.

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 calledthe greatest common divisor
  - The greatest common divisor of a and b is denoted by $\gcd(a, b)$.
- What is the greatest common divisor of 24 and 36?
  - $\gcd(24, 36) = 12$
- What is the greatest common divisor of 17 and 22?
  - $\gcd(17,22) = 1$
- 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 $1 \leq i < j \leq n$.
- Determine whether the integers 10, 17 and 21 are pairwise relatively prime.
  - Because $\gcd(10,17) = 1$, $\gcd(10,21) = 1$, and $\gcd(17,21) = 1$
  - 10, 17, and 21 are pairwise relatively prime.
- Determine whether the integers 10, 19, and 24 are pairwise relatively prime.
  - Because $\gcd(10,24) = 2$, 10, 19, and 24 are not pairwise relatively prime.

Finding the Greatest Common Divisor Using Prime Factorizations

- Suppose the prime factorizations of a and b are:
  - $a = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n}, b = p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n}$
  - 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
  - $120 = 23 \cdot 3 \cdot 5$
  - $500 = 22 \cdot 53$
  - $\gcd(120,500) = 2^{\min(3,2)} \cdot 3^{\min(1,0)} \cdot 5^{\min(1,3)} = 2^2 \cdot 3^0 \cdot 5^1 = 20$
- 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.

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
  - $\text{lcm}(2^3 3^5 7^2, 2^4 3^3) = 2^{\max(3,4)} \cdot 3^{\max(5,3)} \cdot 7^{\max(2,0)} = 2^4 \cdot 3^5 \cdot 7^2$
- 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 $ab = \gcd(a, b) \cdot \text{lcm}(a, b)$

Euclidean Algorithm

- 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.
- Find $\gcd(91, 287)$:
  - $287 = 91 \cdot 3 + 14$
  - $91 = 14 \cdot 6 + 7$
  - $14 = 7 \cdot 2 + 0$
  - $\gcd(287, 91) = \gcd(91, 14) = \gcd(14, 7) = 7$
- The Euclidean algorithm expressed in pseudocode is

```
procedure gcd(a, b: positive integers)
x := a
y := 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 $a > b$.

Correctness of Euclidean Algorithm

- Lemma 1: Let $a = bq + r$, where a, b, q, and r are integers. Then $\gcd(a, b) = \gcd(b, r)$.
  - Suppose that d divides both a and b.
  - Then d also divides $a - bq = r$
  - Hence, any common divisor of a and b must also be any common divisor of b and r.
  - Suppose that d divides both b and r.
  - Then d also divides $bq + r = a$.
  - Hence, any common divisor of a and b must also be a common divisor of b and r.
  - Therefore, $\gcd(a, b) = \gcd(b, r)$.
- Proof
  - Suppose that a and b are positive integers with $a \geq b$.
  - Let $r_0 = a$ and $r_1 = b$.
  - Successive applications of the division algorithm yields:
    - $r_0 = r_1 q_1 + r_2, \quad 0 \leq r_2 < r_1$
    - $r_1 = r_2 q_2 + r_3, \quad 0 \leq r_3 < r_2$
    - ⋮

        - $r_{n-2} = r_{n-1}q_{n-1} + r_2, 0 \leq r_n < r_{n-1}$
        - $r_{n-1} = r_n q_n$
    - Eventually, a remainder of zero occurs in the sequence of terms: $a = r_0 > r_1 > r_2 > \cdots \geq 0$.
    - The sequence can't contain more than a terms.
    - By Lemma 1 , $\gcd(a, b) = \gcd(r_0, r_1) = \cdots = \gcd(r_{n-1}, r_n) = \gcd(r_n , 0) = r_n$.
    - Hence the greatest common divisor is the last nonzero remainder in the sequence of divisions

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.
- Example
    - $\gcd(6,14) = (-2)\cdot 6 + 1\cdot 14$

Finding gcds as Linear Combinations

- Express $\gcd(252,198) = 18$ as a linear combination of 252 and 198.
- First use the Euclidean algorithm to show $\gcd(252,198) = 18$
    - $252 = 1\cdot 198 + 54$
    - $198 = 3\cdot 54 + 36$
    - $54 = 1\cdot 36 + 18$
    - $36 = 2\cdot 18$
- Now working backwards
    - $18 = 54 - 1\cdot 36$
    - $36 = 198 - 3\cdot 54$
- 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 1)) 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.

Consequences of Bézout's Theorem

- Lemma 2: If a, b, and c are positive integers such that $\gcd(a, b) = 1$ and $a \mid bc$, then $a \mid c$.
  - Assume $\gcd(a, b) = 1$ and $a \mid bc$
  - Since $\gcd(a, b) = 1$, by Bézout's Theorem there are integers s and t such that
    - $sa + tb = 1$
  - Multiplying both sides of the equation by c, yields $sac + tbc = c$.
  - From Theorem 1 of Section 4.1:
    - $a \mid tbc$ (part 2)
    - and a divides $sac + tbc$ since $a \mid sac$ and $a \mid tbc$ (part 1)
  - We conclude $a \mid c$, since $sac + tbc = c$.
- Lemma 3: If p is prime and $p \mid a_1 a_2 \cdots a_n$, then $p \mid a_i$ for some i.
- Lemma 3 is crucial in the proof of the uniqueness of prime factorizations.
  - If $p \mid a_1 a_2 \ldots a_n$ and p does not divide a_1 then $\gcd(a_1, p) = 1$, so $p \mid a_2 \ldots a_n$
  - If $p \mid a_2 \ldots a_n$ and p does not divide a_2 then $\gcd(a_2, p) = 1$, so $p \mid a_3 \ldots a_n$
  - ⋮
  - Either this process stops because some $p \mid a_i$ for some $i < n$ or $p \mid a_n$

Uniqueness of Prime Factorization

- A prime factorization of a positive integer where the primes are in nondecreasing order is unique.
  - This part of the fundamental theorem of arithmetic.
  - Every positive integer has a prime factorization into primes, will be proved in Section 5.2.
- Proof: (by contradiction)
  - Suppose that the positive integer n can be written as a product of primes in two distinct ways
    - $n = p_1 p_2 \ldots p_s$ and $n = q_1 q_2 \ldots q_t$
  - Remove all common primes from the factorizations to get
    - $n = p_{i_1} p_{i_2} \ldots p_{i_u}$ and $n = q_{j_1} q_{j_2} \ldots q_{j_v}$
  - By Lemma 3, it follows that p_{i_1} divides q_{j_k}, for some k
  - contradicting the assumption that p_{i_1} and q_{j_k} are distinct primes.
  - Hence, there can be at most one factorization of n into primes in nondecreasing order.

Dividing Congruences by an Integer

- Dividing both sides of a valid congruence by an integer does not always produce a valid congruence
- 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 $\gcd(c, m) = 1$
- It follows that $m \mid (a - b)$. Hence, $a \equiv b \pmod{m}$.

4.4 Solving Congruences

Wednesday, March 7, 2018 9:24 AM

Linear Congruences

- Definition
  - A congruence of the form $ax \equiv b \pmod{m}$,
  - 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 $ax \equiv b \pmod{m}$ are
  - all integers x that satisfy the congruence.
- Definition
  - An integer ā such that $\bar{a}a \equiv 1 \pmod{m}$ is said to be an inverse of a modulo m.
- Example
  - 5 is an inverse of 3 modulo 7 since $5 \cdot 3 = 15 \equiv 1 \pmod{7}$
- One method of solving linear congruences makes use of an inverse ā, if it exists.
- Although we can not divide both sides of the congruence by a
- we can multiply by ā to solve for x.

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.
  - So, there is a unique positive integer ā less than m that is an inverse of a modulo m
  - And every other inverse of a modulo m is congruent to ā 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.

Finding Inverses

- The Euclidean algorithm and Bézout coefficients gives us

- a systematic approaches to finding inverses.
- Find an inverse of 3 modulo 7.
    - 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.
- Find an inverse of 101 modulo 42620.
    - First use the Euclidian algorithm to show that $\gcd(101,42620) = 1$.
        - $42620 = 45 \cdot 101 + 75$
        - $101 = 1 \cdot 75 + 26$
        - $75 = 2 \cdot 26 + 23$
        - $26 = 1 \cdot 23 + 3$
        - $23 = 7 \cdot 3 + 2$
        - $3 = 1 \cdot 2 + 1$
        - $2 = 2 \cdot 1$
    - 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
    - Therefore 1601 is an inverse of 101 modulo 42620

Using Inverses to Solve Congruences

- We can solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides by ā.
- What are the solutions of the congruence $3x \equiv 4 \pmod{7}$.
    - 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 $-1, -8, -15, \ldots$

The Chinese Remainder Theorem

- In the first century, the Chinese mathematician Sun-Tsu asked:
  - There are certain things whose number is unknown.
  - When divided by 3, the remainder is 2
  - When divided by 5, the remainder is 3
  - When divided by 7, the remainder is 2.
  - What will be the number of things?
- This puzzle can be translated into the solution of the system of congruences:
  - $x \equiv 2 \pmod 3$
  - $x \equiv 3 \pmod 5$
  - $x \equiv 2 \pmod 7$
- Theorem 2: (The Chinese Remainder Theorem)
  - Let $m_1, m_2, \ldots, m_n$ be pairwise relatively prime positive integers greater than one
  - Let $a_1, a_2, \ldots, a_n$ be 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.
- Algorithm
  - To construct a solution first let $M_k = \frac{m}{m_k}$ for $k = 1,2, \ldots, n$ and $m = m_1 m_2, \ldots, m_n$
  - Since $\gcd(m_k, M_k) = 1$, 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
    - $x = a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3$
  - 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.
  - $x \equiv a_1 \pmod{m_1}$
  - $x \equiv a_2 \pmod{m_2}$
  - $\vdots$
  - $x \equiv a_n \pmod{m_n}$

- 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
    - $m = 3 \cdot 5 \cdot 7 = 105$
    - $M_1 = \frac{m}{3} = 35$
    - $M_2 = \frac{m}{5} = 21$
    - $M_3 = \frac{m}{7} = 15$
  - We see that
    - $y_1 = 2$ is an inverse of $M_1 = 35$ modulo 3 since $35 \cdot 2 \equiv 2 \cdot 2 \equiv 1 \pmod 3$
    - $y_2 = 1$ is an inverse of $M_2 = 21$ modulo 5 since $21 \equiv 1 \pmod 5$
    - $y_3 = 1$ is an inverse of $M_3 = 15$ modulo 7 since $15 \equiv 1 \pmod 7$
  - Hence,
    - $x = a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3$
    - $= 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 = 233 \equiv 23 \pmod{105}$
  - We have shown that 23 is the smallest positive integer that is a simultaneous solution.

- Back Substitution
  - We can also solve systems of linear congruences with pairwise relatively prime moduli
  - by rewriting a congruences as an equality using Theorem 4 in Section 4.1
  - substituting the value for the variable into another congruence,
  - and continuing the process until we have worked through all the congruences.
  - This method is known as back substitution.

- Example
  - Use the method of back substitution to find all integers x such that
    - $x \equiv 1 \pmod 5$
    - $x \equiv 2 \pmod 6$
    - $x \equiv 3 \pmod 7$.

- By Theorem 4, the first congruence can be rewritten as $x = 5t + 1$, where t is an integer.
- Substituting into the second congruence yields $5t + 1 \equiv 2 \pmod 6$.
- Solving this tells us that $t \equiv 5 \pmod 6$.
- Using Theorem 4 again gives $t = 6u + 5$ where u is an integer.
- Substituting this back into $x = 5t + 1$, gives $x = 5(6u + 5) + 1 = 30u + 26$.
- Inserting this into the third equation gives $30u + 26 \equiv 3 \pmod 7$.
- Solving this congruence tells us that $u \equiv 6 \pmod 7$.
- By Theorem 4, $u = 7v + 6$, where v is an integer.
- Substituting this expression for u into $x = 30u + 26$
- tells us that $x = 30(7v + 6) + 26 = 210v + 206$.
- Translating this back into a congruence we find the solution $x \equiv 206 \pmod{210}$

Fermat's Little Theorem

- 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 $a^p \equiv a \pmod p$
- This is useful in computing the remainders modulo p of large powers of integers.
- Find $7^{222} \bmod 11$.
  - By Fermat's little theorem, we know that $7^{10} \equiv 1 \pmod{11}$
  - and so $\left(7^{10}\right)^k \equiv 1 \pmod{11}$, for every positive integer k
  - Therefore, $7^{222} = 7^{22\times 10+2} = \left(7^{10}\right)^{22} \times 7^2 \equiv 1^{22} \times 49 \equiv 5 \pmod{11}$.
  - Hence, $7^{222} \bmod 11 = 5$.

Pseudoprimes

- By Fermat's little theorem n > 2 is prime, where
  - $2^{n-1} \equiv 1 \pmod n$.
- But if this congruence holds, n may not be prime.
- Composite integers n such that $2^{n-1} \equiv 1 \pmod n$ are called pseudoprimes to the base 2.
- Example: The integer 341 is a pseudoprime to the base 2.
  - $341 = 11 \cdot 31$
  - $2^{340} \equiv 1 \pmod{341}$ (see in Exercise 37)
  - We can replace 2 by any integer $b \geq 2$.
- Definition
  - Let b be a positive integer.
  - If n is a composite integer, and $b^{n-1} \equiv 1 \pmod n$
  - then n is called a pseudoprime to the base b
- Given a positive integer n, such that $2^{n-1} \equiv 1 \pmod n$:

- If n does not satisfy the congruence, it is composite.
- If n does satisfy the congruence, it is either prime or a pseudoprime to the base 2.

- Doing similar tests with additional bases b, provides more evidence as to whether n is prime.
- Among the positive integers not exceeding a positive real number x, compared to primes
- there are relatively few pseudoprimes to the base b.
- For example, among the positive integers less than 10^{10} there are 455,052,512 primes
- but only 14,884 pseudoprimes to the base 2

Primitive Roots

- Definition
  - A primitive root modulo a prime p is an integer r in $\mathbb{Z}_p$ such that
  - every nonzero element of $\mathbb{Z}_p$ is a power of r
- Example
  - Since every element of $\mathbb{Z}_{11}$ is a power of 2, 2 is a primitive root of 11.
  - Powers of 2 modulo 11
    - $2^1 = 2$
    - $2^2 = 4$
    - $2^3 = 8$
    - $2^4 = 5$
    - $2^5 = 10$
    - $2^6 = 9$
    - $2^7 = 7$
    - $2^8 = 3$
    - $2^9 = 6$
    - $2^{10} = 1$
- Example
  - Since not all elements of $\mathbb{Z}_{11}$ are powers of 3, 3 is not a primitive root of 11.
  - Powers of 3 modulo 11
    - $3^1 = 3$
    - $3^2 = 9$
    - $3^3 = 5$
    - $3^4 = 4$
    - $3^5 = 1$
    - and the pattern repeats for higher powers.
- Important Fact
  - There is a primitive root modulo p for every prime number p

Discrete Logarithms

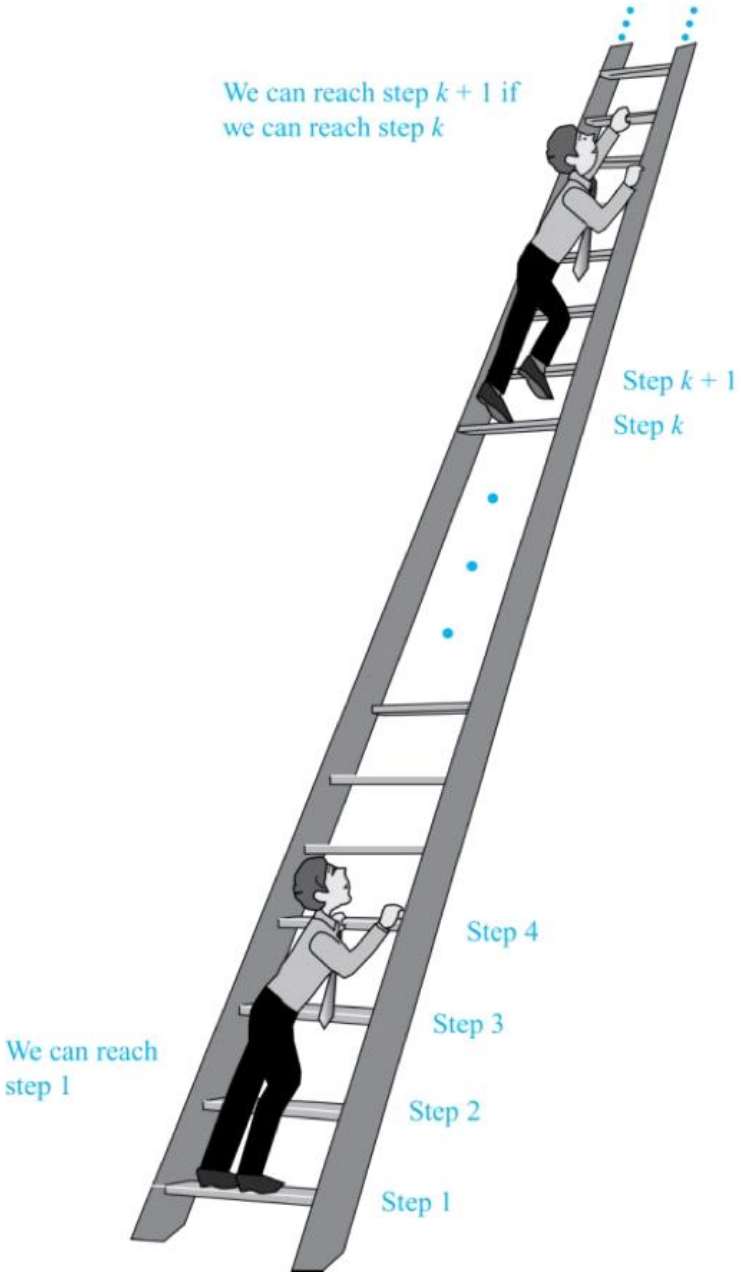
- Suppose p is prime and r is a primitive root modulo p.
- If a is an integer between 1 and $p-1$, that is an element of $\mathbb{Z}_p$,
- there is a unique exponent e such that $r^e = a$ in $\mathbb{Z}_p$, that is, $r^e \bmod p = a$.
- Definition
    - Suppose p is prime
    - r is a primitive root modulo p
    - and a is an integer between 1 and $p-1$, inclusive.
    - If $r^3 \bmod p = a$ and $1 \leq e \leq p-1$
    - We say that e is the discrete logarithm of a modulo p to the base r and we write $\log_r a = e$
- Example 1
    - We write $\log_2 3 = 8$
    - Since the discrete logarithm of 3 modulo 11 to the base 2 is 8 as $28 = 3$ modulo 11.
- Example 2
    - We write $\log_2 5 = 4$
    - since the discrete logarithm of 5 modulo 11 to the base 2 is 4 as $24 = 5$ modulo 11.
- There is no known polynomial time algorithm for computing the discrete logarithm
- The problem plays a role in cryptography as will be discussed in Section 4.6.

5.1 Mathematical Induction

Monday, March 12, 2018 8:52 AM

Climbing an Infinite Ladder

- Suppose we have an infinite ladder:
  - We can reach the first rung of the ladder.
  - If we can reach a particular rung of the ladder
  - then we can reach the next rung.
- From (1), we can reach the first rung.
- Then by applying (2), we can reach the second rung.
- Applying (2) again, the third rung. And so on.
- We can apply (2) any number of times to reach any particular rung, no matter how high up.



Principle of Mathematical Induction

- To prove that $P(n)$ is true for all positive integers n, we complete these steps:
  - Basis Step: Show that $P(1)$ is true.
  - Inductive Step: Show that $P(k) \rightarrow P(k+1)$ is true for all positive integers k
- To complete the inductive step
  - assuming the inductive hypothesis that $P(k)$ holds for an arbitrary integer k
  - show that must $P(k+1)$ be true.
- Climbing an Infinite Ladder Example:
  - Basis Step
    - By (1), we can reach rung 1.
  - Inductive Step

        - Assume the inductive hypothesis that we can reach rung k
        - Then by (2), we can reach rung $k + 1$.
    - Hence, $P(k) \rightarrow P(k + 1)$ is true for all positive integers k
    - We can reach every rung on the ladder.

Important Points About Using Mathematical Induction

- Mathematical induction can be expressed as the rule of inference
    - $\big(P(1) \wedge \forall k\big(P(k) \rightarrow P(k + 1)\big)\big) \rightarrow \forall n\, P(n)$
    - where the domain is the set of positive integers.
- In mathematical induction, we don't assume that $P(k)$ is true for all positive integers!
- We show that if we assume that $P(k)$ is true, then $P(k + 1)$ must also be true.
- Proofs by mathematical induction do not always start at the integer 1.
- In such a case, the basis step begins at a starting point b where b is an integer.
- We will see examples of this soon.

Proving a Summation Formula by Mathematical Induction

- Example
    - Show that: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$
- Solution
    - Basis Step
        - $P(1)$ is true since $\frac{1(1+1)}{2} = 1$
    - Inductive Step
        - Assume true for $P(k)$
        - The inductive hypothesis is $\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$
        - Under this assumption
        - $1 + 2 + \cdots + k + (k + 1) = \frac{k(k+1)}{2} + (k + 1) = \frac{(k+1)(k+2)}{2}$

Validity of Mathematical Induction

- Mathematical induction is valid because of the well ordering property, which states that
- every nonempty subset of the set of positive integers has a least element
- Suppose that P(1) holds and $P(k) \rightarrow P(k + 1)$ is true for all positive integers k.
- Assume there is at least one positive integer n for which $P(n)$ is false.
- Then the set S of positive integers for which $P(n)$ is false is nonempty.
- By the well-ordering property, S has a least element, say m.
- We know that m cannot be 1 since $P(1)$ holds.

- Since m is positive and greater than 1, $m-1$ must be a positive integer.
- Since $m-1 < m$, it is not in S, so $P(m-1)$ must be true.
- But then, since the conditional $P(k) \rightarrow P(k+1)$ for every positive integer k holds,
- $P(m)$ must also be true. This contradicts $P(m)$ being false.
- Hence, $P(n)$ must be true for every positive integer n.

Conjecturing and Proving Correct a Summation Formula

- Example
  - Conjecture and prove correct a formula for the sum of the first n positive odd integers
  - Then prove your conjecture
- Solution
  - We have
    - $1 = 1$
    - $1+3=4$
    - $1+3+5=9$
    - $1+3+5+7=16$
    - $1+3+5+7+9=25$.
  - We can conjecture that the sum of the first n positive odd integers is n^2,
    - $1+3+5+\cdots+(2n-1)+(2n+1)=n^2$
  - We prove the conjecture is proved correct with mathematical induction.
  - Basis Step
    - P(1) is true since $1^2=1$.
  - Inductive Step: $P(k) \rightarrow P(k+1)$ for every positive integer k.
    - Inductive Hypothesis: $1+3+5+\cdots+(2k-1)=k^2$
    - So, assuming $P(k)$, it follows that:
    - $1+3+5+\cdots+(2k-1)+(2k+1)$
    - $=[1+3+5+\cdots+(2k-1)]+(2k+1)$
    - $=k^2+(2k+1)$
    - $=(k+1)^2$
  - Hence, we have shown that $P(k+1)$ follows from $P(k)$.
  - Therefore the sum of the first n positive odd integers is n^2

Proving Inequalities

- Example
  - Use mathematical induction to prove that $n < 2^n$ for all positive integers n.
- Solution
  - Let $P(n)$ be the proposition that $n < 2^n$.
  - Basis Step

        - $P(1)$ is true since $1 < 2^1 = 2$.
    - Inductive Step
        - Assume $P(k)$ holds, i.e., $k < 2^k$, for an arbitrary positive integer k.
        - Must show that P(k + 1) holds.
        - Since by the inductive hypothesis, $k < 2^k$, it follows that:
        - $k + 1 < 2^k + 1 \leq 2^k + 2^k = 2 \cdot 2^k = 2^{k+1}$
        - Therefore $n < 2^n$ holds for all positive integers n.
- Example
    - Use mathematical induction to prove that $2^n < n!$, for every integer $n \geq 4$
- Solution
    - Let $P(n)$ be the proposition that $2^n < n!$
    - Basis Step
        - $P(4)$ is true since $24 = 16 < 4! = 24$
    - Inductive Step
        - Assume P(k) holds, i.e., $2^k < k!$ for an arbitrary integer $k \geq 4$.
        - To show that $P(k + 1)$ holds
        - $2^{k+1} = 2 \cdot 2^k < 2 \cdot 2^k < (k + 1)k! = (k + 1)!$
        - Therefore, $2^n < n!$ holds, for every integer $n \geq 4$

Proving Divisibility Results

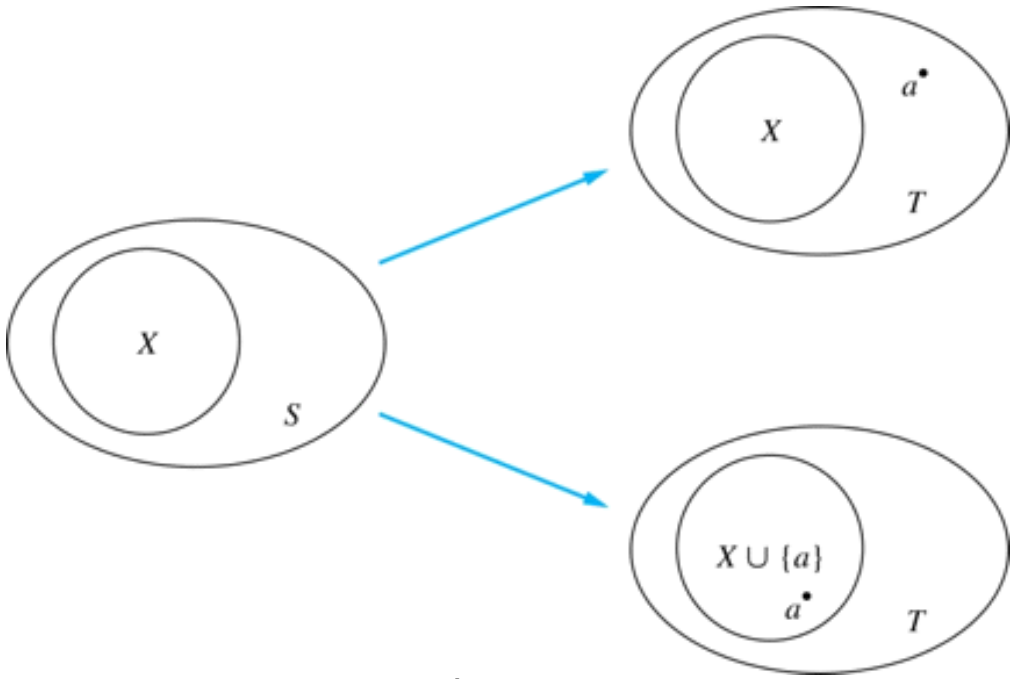
- Example
    - Use mathematical induction to prove that $n^3 - n$ is divisible by 3
    - for every positive integer n.
- Solution
    - Let $P(n)$ be the proposition that $n^3 - n$ is divisible by 3.
    - Basis Step
        - $P(1)$ is true since $13 - 1 = 0$, which is divisible by 3
    - Inductive Step
        - Assume $P(k)$ holds, i.e., $k^3 - k$ is divisible by 3, for an arbitrary positive integer k
        - To show that P(k + 1) follows
        - $(k + 1)^3 - (k + 1) = \left(k^3 + 3k^2 + 3k + 1\right) - (k + 1) = \left(k^3 - k\right) + 3\left(k^2 + k\right)$
        - By the inductive hypothesis, the first term $\left(k^3 - k\right)$ is divisible by 3
        - and the second term is divisible by 3 since it is an integer multiplied by 3.
        - So by part (i) of Theorem 1 in Section 4.1, $(k + 1)^3 - (k + 1)$ is divisible by 3
    - Therefore, $n^3 - n$ is divisible by 3, for every integer positive integer n.

Number of Subsets of a Finite Set

- Example

- Use mathematical induction to show that if S is a finite set with n elements
- where n is a nonnegative integer, then S has 2^n subsets.
- (Chapter 6 uses combinatorial methods to prove this result.)

- Solution
  - Let $P(n)$ be the proposition that a set with n elements has 2^n subsets.
  - Basis Step
    - $P(0)$ is true, because the empty set has only itself as a subset and $2^0 = 1$.
  - Inductive Step
    - Assume $P(k)$ is true for an arbitrary nonnegative integer k.
    - Inductive Hypothesis
    - For an arbitrary nonnegative integer k, every set with k elements has 2^k subsets
    - Let T be a set with $k + 1$ elements
    - Then $T = S \cup \{a\}$, where $a \in T$ and $S = T - \{a\}$. Hence $|S| = k$.
    - For each subset X of S, there are exactly two subsets of T, i.e., X and $X \cup \{a\}$.



    - By the inductive hypothesis S has 2^k subsets.
    - Since there are two subsets of T for each subset of S,
    - the number of subsets of T is $2 \cdot 2^k = 2^{k+1}$

An Incorrect "Proof" by Mathematical Induction

- $P(n)$:=every set of n lines in the plane, no two of which are parallel, meet in a point
- Here is a "proof" that $P(n)$ is true for all positive integers $n \geq 2$.
- Basis Step
  - The statement P(2) is true
  - because any two lines in the plane that are not parallel meet in a common point.
- Inductive hypothesis

- $P(k)$ is true for the positive integer $k \geq 2$
- every set of k lines in the plane, no two of which are parallel, meet in a common point.

- Inductive Step
  - We must show that if $P(k)$ holds, then $P(k+1)$ holds
  - If every set of k lines in the plane, no two of which are parallel, k ≥ 2, meet in a point
  - Then every set of $k+1$ lines in the plane, no two of which are parallel, meet in a point.
  - Consider a set of $k+1$ distinct lines in the plane, no two parallel.
  - By the inductive hypothesis, the first k of these lines must meet in a common point p_1.
  - By the inductive hypothesis, the last k of these lines meet in a common point p_2.
  - If p_1 and p_2 are different points, all lines containing both of them must be the same line since two points determine a line.
  - This contradicts the assumption that the lines are distinct.
  - Hence, $p_1 = p_2$ lies on all $k+1$ distinct lines, and therefore $P(k+1)$ holds.
  - Assuming that $k \geq 2$, distinct lines meet in a common point
  - Then every $k+1$ lines meet in a common point.
- There must be an error in this proof since the conclusion is absurd. But where is the error?
  - $P(k) \rightarrow P(k+1)$ only holds for $k \geq 3$
  - It is not the case that $P(2)$ implies $P(3)$
  - The first two lines must meet in a common point p_1 and the second two must meet in a common point p_2
  - They do not have to be the same point since only the second line is common to both sets of lines.

5.2 Strong Induction and Well-Ordering

Wednesday, March 14, 2018 9:03 AM

Strong Induction

- To prove that $P(n)$ is true for all positive integers n
- where $P(n)$ is a propositional function, complete two steps:
- Basis Step
  - Verify that the proposition $P(1)$ is true.
- Inductive Step
  - Show the conditional statement
    - $[P(1) \wedge P(2) \wedge \cdots \wedge P(k)] \rightarrow P(k+1)$
  - holds for all positive integers k.
- Strong Induction is sometimes called
  - the second principle of mathematical induction
  - complete induction

Proof using Strong Induction

- Prove that every natural number $n > 7$ can be written as $3p + 5q$
- where p and q are natural numbers
- Prove this result using strong induction
- Basis Step
  - $8 = 3 \times 1 + 5 \times 1$
- Inductive Step
  - Inductive hypothesis: The statement is true for any n for $k \geq n \geq 8$
  - In particular it is true for $k + 1 - 3$ (assuming $k + 1 - 3 \geq 8$)
  - So $k + 1 - 3 = 3p + 5q$ and so $k + 1 = 3(p + 1) + 5q$
- What happens if $k + 1 = 9$ or $k + 1 = 10$?
  - Add those cases into the basis step
  - $9 = 3 \times 3 + 5 \times 0$
  - $10 = 3 \times 0 + 5 \times 2$

Which Form of Induction Should Be Used?

- We can always use strong induction instead of mathematical induction.
- But there is no reason to use it if it is simpler to use mathematical induction
- In fact, the principles of mathematical induction, strong induction, and the well-ordering property are all equivalent.
- Sometimes it is clear how to proceed using one of the three methods, but not the other two.

Proof of the Fundamental Theorem of Arithmetic

- Show that if n is an integer greater than 1
- Then n can be written as the product of primes.
- Let $P(n)$ be the proposition that n can be written as a product of primes.
- Basis Step
  - P(2) is true since 2 itself is prime.
- Inductive Step
  - The inductive hypothesis is $P(j)$ is true for $j \in \mathbb{Z}$ with $2 \leq j \leq k$
  - To show that $P(k+1)$ must be true under this assumption
  - Two cases need to be considered:
  - If $k+1$ is prime, then $P(k+1)$ is true.
  - Otherwise, $k+1$ is composite
  - And it can be written as the product of two positive integers
  - a and b with $2 \leq a \leq b \leq k+1$
  - By inductive hypothesis a and b can be written as product of primes
  - Therefore k + 1 can also be written as the product of those primes.
- Hence, every integer greater than 1 can be written as product of primes

Well-Ordering Property

- Well-ordering property
  - Every nonempty set of nonnegative integers has a least element.
- The well-ordering property is one of the axioms of the positive integers
- The well-ordering property can be used directly in proofs.
- The well-ordering property can be generalized.
- Definition: A set is well ordered if every subset has a least element.
  - N is well ordered under $\leq$.
  - The set of finite strings over an alphabet using lexicographic ordering is well ordered.

Proof of The Division Algorithm

- Use the well-ordering property to prove the division algorithm
  - If a is an integer and d is a positive integer, then
  - there are unique integers q and r with $0 \leq r < d$, such that
  - $a = dq + 4$
- Let S be the set of nonnegative integers of the form $a = dq, q \in \mathbb{Z}$.
- The set is nonempty since $-dq$ can be made as large as needed
- By the well-ordering property, S has a least element $r = a - dq_0$
- The integer r is nonnegative.

- It also must be the case that $r < d$
- If it were not, then there would be a smaller nonnegative element in S
    - $a - d(q_0 + 1) = a - dq_0 - d = r - d > 0$
- Therefore, there are integers q and r with $0 \leq r < d$

5.3 Recursive Definitions and Structural Induction

Wednesday, March 14, 2018 9:31 AM

Recursively Defined Functions

- Definition
  - A recursive or inductive definition of a function consists of two steps.
  - Basis Step
    - Specify the value of the function at zero.
  - Recursive Step
    - Give a rule for finding the at an integer from its values at smaller integers
  - A function $f(n)$ is the same as a sequence $a_0, a_1, \ldots$ where $f(i) = a_i$
  - This was done using recurrence relations in Section 2.4
- Example 1
  - Suppose f is defined by
    - $f(0) = 3$
    - $f(n+1) = 2f(n) + 3$
  - Find $f(1), f(2), f(3), f(4)$
    - $f(1) = 2f(0) + 3 = 2 \cdot 3 + 3 = 9$
    - $f(2) = 2f(1) + 3 = 2 \cdot 9 + 3 = 21$
    - $f(3) = 2f(2) + 3 = 2 \cdot 21 + 3 = 45$
    - $f(4) = 2f(3) + 3 = 2 \cdot 45 + 3 = 93$
- Example 2
  - Give a recursive definition of the factorial function n!
  - $f(0) = 1$
  - $f(n+1) = (n+1) \cdot f(n)$
- Example
  - Give a recursive definition of $\sum_{k=0}^{n} a_k$
  - The first part of the definition is
    - $\sum_{k=0}^{0} a_k = a_0$
  - The second part is
    - $\sum_{k=0}^{n+1} a_k = \left(\sum_{k=0}^{n} a_k\right) + a_{n+1}$

Fibonacci Numbers

- The Fibonacci numbers are defined as follows:
  - $f_0 = 0$
  - $f_1 = 1$
  - $f_n = f_{n-1} + f_{n-2}$
- Find f_2, f_3, f_4, f_5
  - $f_2 = f_1 + f_0 = 1 + 0 = 1$
  - $f_3 = f_2 + f_1 = 1 + 1 = 2$
  - $f_4 = f_3 + f_2 = 2 + 1 = 3$
  - $f_5 = f_4 + f_3 = 3 + 2 = 5$
- Show that whenever $n \geq 3, f_n > \alpha^{n-2}$, where $\alpha = \frac{1+\sqrt{5}}{2}$
  - Let $P(n)$ be the statement $f_n > \alpha^{n-2}$.
  - Use strong induction to show that $P(n)$ is true whenever $n \geq 3$.
  - Basis step
    - $P(3)$ holds since $\alpha < 2 = f_3$
    - $P(4)$ holds since $\alpha^2 = \frac{3+\sqrt{5}}{2} < 3 = f_4$
  - Inductive step
    - Assume that $P(j)$ holds
    - i.e., $f_j > \alpha^{j-2}$ for all integers j with $3 \leq j \leq k$, where $k \geq 4$.
    - Show that $P(k+1)$ holds, i.e., $f_{k+1} > \alpha^{k-1}$.
    - Since $\alpha^2 = \alpha + 1$ (because α is a solution of $x^2 - x - 1 = 0$),
    - $\alpha^{k-1} = \alpha^2 \cdot \alpha^{k-3} = (\alpha + 1) \cdot \alpha^{k-3} = \alpha \cdot \alpha^{k-3} + 1 \cdot \alpha^{k-3} = \alpha^{k-2} + \alpha^{k-3}$
    - By the inductive hypothesis, because $k \geq 4$ we have
      - $f_{k-1} > \alpha^{k-3}$
      - $f_k > \alpha^{k-3}$
    - Therefore, it follows that
      - $f_{k+1} = f_k + f_{k-1} > \alpha^{k-2} + \alpha^{k-3} = \alpha^{k-1}$
    - Hence, $P(k+1)$ is true.

Lamé's Theorem

- Lamé's Theorem
  - Let a and b be positive integers with $a \geq b$.
  - Then the number of divisions used by the Euclidian algorithm to find gcd(a,b)
  - is less than or equal to five times the number of decimal digits in b.
- Proof
  - When we use the Euclidian algorithm to find gcd(a,b) with $a \geq b$,
  - n divisions are used to obtain (with $a = r_0, b = r_1$)

- $r_0 = r_1 q_1 + r_2, \quad 0 < r_2 < r_1$
- $r_1 = r_2 q_2 + r_3, \quad 0 < r_3 < r_2$
- ⋮
- $r_{n-2} = r_{n-1} q_{n-1} + r_n, \quad 0 < r_n < r_{n-1}$
- $r_{n-1} = r_n q_n$

- Since each quotient $q_1, q_2, \ldots, q_{n-1}$ is at least 1 and $q_n \geq 2$
  - $r_n \geq 1 = f_2$
  - $r_{n-1} \geq 2r_n \geq 2f_2 = f_3$
  - $r_{n-2} \geq r_{r-1} + r_n \geq f_3 + f_2 = f_4$
  - ⋮
  - $r_2 > r_3 + r_4 \geq f_{n-1} + f_{n-2} = f_n$
  - $b = r_1 \geq r_2 + r_3 \geq f_n + f_{n-1} = f_{n+1}$
- If n divisions are used to find $\gcd(a, b)$ with $a \geq b$, then b $\geq f_{n+1}$
- By Example 4, $f_{n+1} > \alpha^{n-1}$, for $n > 2$, where $\alpha = \frac{1+\sqrt{5}}{2}$.
- Therefore, $b > \alpha^{n-1}$.
- Because $\log_{10} \alpha \approx 0.208 > \frac{1}{5}$, $\log_{10} b > (n-1) \log_{10} \alpha > \frac{n-1}{5}$
- Hence, $n - 1 < 5 \cdot \log_{10} b$
- Suppose that b has k decimal digits. Then $b < 10k$ and $\log_{10} b < k$.
- It follows that $n - 1 < 5k$ and since k is an integer, $n \leq 5k$.
- Therefore, O(log b) divisions are used to find $\gcd(a, b)$ whenever $a > b$.
- The number of divisions needed is less than or equal to $5 \cdot (\log_{10} b + 1)$
- Since the number of decimal digits in b is less than or equal to $\log_{10} b + 1$

Recursively Defined Sets and Structures

- Recursive definitions of sets have two parts:
  - The basis step specifies an initial collection of elements.
  - The recursive step gives the rules for forming new elements in the set from those already known to be in the set.
- Sometimes the recursive definition has an exclusion rule, which specifies that
  - the set contains nothing other than those elements specified in the basis step
  - and generated by applications of the rules in the recursive step.
- We always assume that the exclusion rule holds, even if it is not explicitly mentioned.
- Example: Subset of Integers S
  - Basis step: 3 ∈ S.
  - Recursive step: If x ∈ S and y ∈ S, then $x + y$ is in S.
  - Initially 3 is in S, then $3 + 3 = 6$, then $3 + 6 = 9$, etc.
- Example: The natural numbers ℕ.

- Basis step: $0 \in \mathbb{N}$.
- Recursive step: If n is in $\mathbb{N}$, then $n + 1$ is in $\mathbb{N}$.
- Initially 0 is in $\mathbb{N}$, then $0 + 1 = 1$, then $1 + 1 = 2$, etc.

Strings

- Definition: The set Σ^* of strings over the alphabet Σ:
  - Basis step: $\lambda \in \Sigma^*$ (λ is the empty string)
  - Recursive step: If w is in Σ^* and x is in Σ, $wx \in \Sigma^*$.
- Example
  - If $\Sigma = \{0,1\}$
  - The strings in in Σ^* are the set of all bit strings, λ, 0, 1, 00,01,10, 11, etc.
- Example
  - If $\Sigma = \{a,b\}$, show that aab is in Σ^*.
  - Since $\lambda \in \Sigma^*$ and $a \in \Sigma$, $a \in \Sigma^*$.
  - Since $a \in \Sigma^*$ and $a \in \Sigma$, $aa \in \Sigma^*$.
  - Since $aa \in \Sigma^*$ and $b \in \Sigma$, $aab \in \Sigma^*$.

String Concatenation

- Two strings can be combined via the operation of concatenation.
- Let Σ be a set of symbols and Σ^* be the set of strings formed from the symbols in Σ.
- We can define the concatenation of two strings, denoted by $\cdot$, recursively as follows:
  - Basis step
    - If $w \in \Sigma^*$, then $w \cdot \lambda = w$
  - Recursive step
    - If $w_1 \in \Sigma^*$ and $w_2 \in \Sigma^*$ and $x \in \Sigma^*$ then $w_1 \cdot (w_2 x) = (w_1 \cdot w_2)x$
- Often $w_1 \cdot w_2$ is written as $w_1 w_2$.
- If $w_1 =$ abra and $w_2 =$ cadabra, the concatenation $w_1 w_2 =$ abracadabra.

Length of a String

- Give a recursive definition of $l(w)$, the length of the string w.
- The length of a string can be recursively defined by:
  - $l(\lambda) = 0$
  - $l(wx) = l(w) + 1$ if $w \in \Sigma^*$ and $x \in \Sigma$

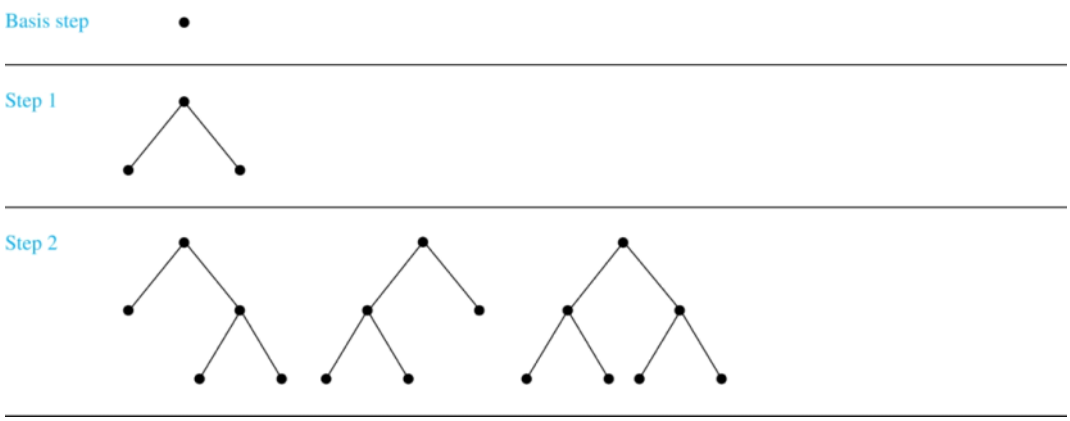
Rooted Trees

- The set of rooted trees, where a rooted tree consists of
  - a set of vertices containing a distinguished vertex called the root
  - edges connecting these vertices
- can be defined recursively by these steps

- Basis step
  - A single vertex r is a rooted tree.
- Recursive step
  - Suppose that $T_1, \dots, T_n$ are disjoint rooted trees with roots $r_1, \dots, r_n$ respectively.
  - Then the graph formed by
    - start with a root r, which is not in any of the rooted trees $T_1, \dots, T_n$
    - add an edge from r to each of the vertices $r_1, \dots, r_n$
  - is also a rooted tree.

Full Binary Trees

- The set of full binary trees can be defined recursively by these steps.
  - Basis step
    - There is a full binary tree consisting of only a single vertex r.
  - Recursive step
    - If T_1 and T_2 are disjoint full binary trees
    - There is a full binary tree, denoted by $T_1 \cdot T_2$, consisting of
      - a root r
      - edges connecting the root to each of the roots of T_1 and T_2



- The height $h(T)$ of a full binary tree T is defined recursively as follows:
  - Basis step
    - The height of a full binary tree T consisting of only a root r is $h(T) = 0$
  - Recursive step
    - If T_1 and T_2 are full binary trees
    - Then the full binary tree $T = T_1 \cdot T_2$ has height
    - $h(T) = 1 + \max\left(h(T_1), h(T_2)\right)$
- The number of vertices $n(T)$ of a full binary tree T is defined recursively as follows
  - Basis step
    - $n(T) = 1$ for full binary tree T consisting of only a root r
  - Recursive step
    - If T_1 and T_2 are full binary trees

- Then the full binary tree $T = T_1 \cdot T_2$ has the number of vertices
- $n(T) = 1 + n(T_1) + n(T_2)$

Structural Induction

- To prove a property of the elements of a recursively defined set, we use structural induction
- Basis step
  - Show that The result holds for all elements specified in the basis step
- Recursive step
  - Suppose the statement is true for each of the elements used to construct new elements in the recursive step of the definition
  - Show that the result holds for these new elements.
- The validity of structural induction can be shown to follow from the principle of mathematical induction.

Structural Induction and Binary Trees

- If T is a full binary tree, then $n(T) \leq 2^{h(T)+1} - 1$
- Proof: Use structural induction
- Basis step
  - The result holds for a full binary tree consisting only of a root
  - $n(T) = 1$ and $h(T) = 0$
  - Hence, $n(T) = 1 \leq 2^{0+1} - 1 = 1$
- Recursive step
  - Assume $n(T_1) \leq 2^{h(T_1)+1} - 1$ and $n(T_2) \leq 2^{h(T_2)+1} - 1$ for full binary trees T_1, T_2
  - $n(T) = 1 + n(T_1) + n(T_2)$
  - $\leq 1 + (2^{h(T_1)+1} - 1) + (2^{h(T_2)+1} - 1)$
  - $\leq 2 \cdot \max(2^{h(T_1)+1}, 2^{h(T_2)+1}) - 1$
  - $= 2 \cdot 2^{\max(h(T_1),h(T_2))+1} - 1$
  - $= 2 \cdot 2^{h(t)} - 1$
  - $= 2^{h(t)+1} - 1$

5.4 Recursive Algorithms

Monday, March 19, 2018 9:05 AM

Recursive Algorithms

- An algorithm is called recursive if it solves a problem by reducing it to an instance of the same problem with smaller input.
- For the algorithm to terminate, the instance of the problem must eventually be reduced to some initial case for which the solution is known.

Recursive Factorial Algorithm

- Give a recursive algorithm for computing $n!$, where n is a nonnegative integer.
- Use the recursive definition of the factorial function.

procedure *factorial*(n: nonnegative integer)
if $n = 0$ **then return** 1
else return $n \cdot factorial\ (n - 1)$
{output is $n!$}

Recursive Exponentiation Algorithm

- Give a recursive algorithm for computing a^n, where
  - a is a nonzero real number
  - n is a nonnegative integer
- Use the recursive definition of a^n

procedure *power*(a: nonzero real number, n: nonnegative integer)
if $n = 0$ **then return** 1
else return $a \cdot power\ (a, n - 1)$
{output is a^n}

Recursive GCD Algorithm

- Give a recursive algorithm for computing the greatest common divisor of two nonnegative integers a and b with $a < b$
- Use
  - the reduction $\gcd(a, b) = \gcd(b \bmod a, a)$
  - the condition $\gcd(0, b) = b$ when $b > 0$.

```
procedure gcd(a,b: nonnegative integers
                    with a < b)
if a = 0 then return b
else return gcd (b mod a, a)
{output is gcd(a, b)}
```

Recursive Binary Search Algorithm

- Construct a recursive version of a binary search algorithm.
- Assume
  - $a_1, a_2, \ldots, a_n$ is an increasing sequence of integers.
  - Initially i is 1 and j is n
  - We are searching for x

```
procedure binary search(i, j, x : integers, 1≤ i ≤ j ≤n)
m := ⌊(i + j)/2⌋
if x = a_m then
        return m
else if (x < a_m and i < m) then
        return binary search(i,m−1,x)
else if (x > a_m and j >m) then
        return binary search(m+1,j,x)
else return 0
{output is location of x in a_1, a_2,...,a_n if it appears, otherwise 0}
```

Proving Recursive Algorithms Correct

- Both mathematical and str0ng induction are useful techniques to show that recursive algorithms always produce the correct output.
- Prove that the algorithm for computing the powers of real numbers is correct.

```
procedure power(a: nonzero real number, n: nonnegative integer)
if n = 0 then return 1
else return a · power (a, n − 1)
{output is a^n}
```

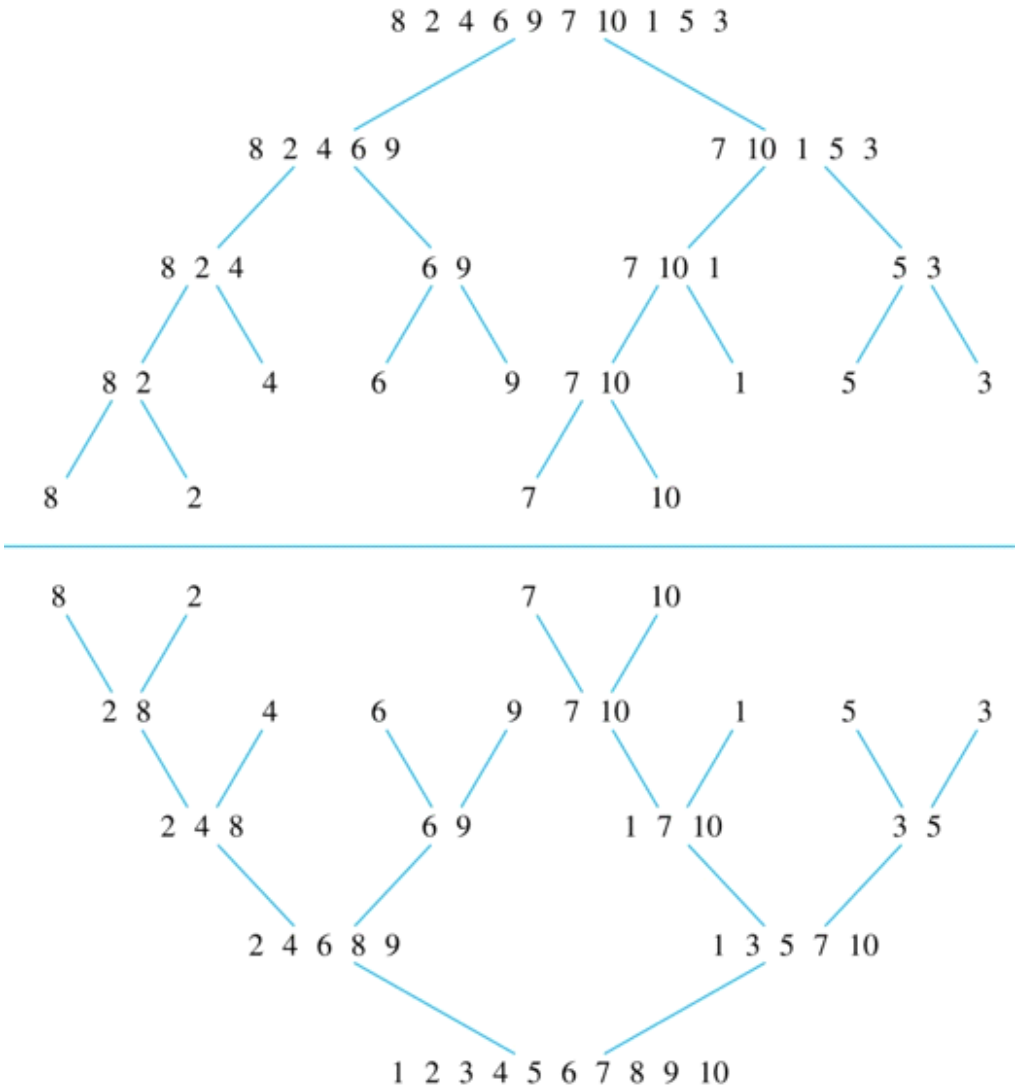
- Use mathematical induction on the exponent n.
- Basis step
  - $a^0 = 1$ for every nonzero real number a, and power$(a, 0) = 1$
- Inductive step
  - The inductive hypothesis is that power$(a, k) = a^k$, for all $a \neq 0$.
  - Assuming the inductive hypothesis, the algorithm correctly computes a^{k+1}
  - power$(a, k + 1) = a \cdot$ power $(a, k) = a \cdot a^k = a^{k+1}$

Merge Sort

- Merge Sort works by iteratively splitting a list (with an even number of elements)

into two sublists of equal length until each sublist has one element.

- Each sublist is represented by a balanced binary tree.
- At each step a pair of sublists is successively merged into a list with the elements in increasing order. The process ends when all the sublists have been merged.
- The succession of merged lists is represented by a binary tree.
- Example
    - Use merge sort to put the list 8,2,4,6,9,7,10, 1, 5, 3 into increasing order.



Recursive Merge Sort

- Construct a recursive merge sort algorithm
- Begin with the list of n elements L

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

- Subroutine merge, which merges two sorted lists

```
procedure merge(L_1, L_2 :sorted lists)
L := empty list
while L_1 and L_2 are both nonempty
   remove smaller of first elements of L_1 and L_2 from its list;
        put at the right end of L
   if this removal makes one list empty
      then remove all elements from the other list and append them to L
return L {L is the merged list with the elements in increasing order}
```

- Complexity of Merge: Two sorted lists with m elements and n elements can be merged into a sorted list using no more than $m + n - 1$ comparisons

Complexity of Merge Sort

- The number of comparisons needed to merge a list with n elements is $O(n \log n)$.
- For simplicity, assume that n is a power of 2, say 2^m.
- At the end of the splitting process, we have a binary tree with m levels, and 2^m lists with one element at level m.
- The merging process begins at level m with the pairs of 2^m lists with one element combined into 2^{m-1} lists of two elements.
- Each merger takes two one comparison.
- The procedure continues, at each level $(k = m, m-1, \dots, 3,2,1)$ 2^k lists with 2^{m-k} elements are merged into 2^{k-1} lists, with 2^{m-k+1} elements at level $k-1$.
- We know (by the complexity of the merge subroutine) that each merger takes at most $2^{m-k} + 2^{m-k} - 1 = 2^{m-k+1} - 1$ comparisons.
- Summing over the number of comparisons at each level, shows that
- $\sum_{k=1}^{m} 2^{k-1}\left(2^{m-k+1} - 1\right) = \sum_{k=1}^{m} 2^m - \sum_{k=1}^{m} 2^{k-1} = m \cdot 2^m - (2^m - 1) = n \log n - n + 1$
- because $m = \log n$ and $n = 2^m$
- The fastest comparison-based sorting algorithms have $O(n \log n)$ time complexity
- So, merge sort achieves the best possible big-O estimate of time complexity

6.1 The Basics of Counting

Wednesday, March 21, 2018 8:52 AM

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.
- Example 1
  - How many bit strings of length seven are there?
  - Since each of the seven bits is either a 0 or a 1, the answer is $2^7 = 128$.
- Example 2
  - How many different license plates can be made if
  - each plate contains a sequence of 3 uppercase English letters followed by 3 digits?
  - There are $26 \cdot 26 \cdot 26 \cdot 10 \cdot 10 \cdot 10 = 17{,}576{,}000$ different possible license plates.
- Example 3: Counting Functions
  - How many functions are there from a set with m elements to a set with n elements?
  - 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.
- Example 4: Counting One-to-One Functions
  - How many 1-to-1 functions are there from a set with m elements to one with n elements?
  - 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 $n(n-1)(n-2)\cdots(n-m+1)$ functions.
- Example 5: Counting Subsets of a Finite Set
  - Show that the number of different subsets of a finite set S is $2^{|S|}$
  - When the elements of S are listed in an arbitrary order
  - There is a one-to-one correspondence between subsets of S and bit strings of length |S|.
  - When the ith element is in the subset, the bit string has a 1 in the ith position and a 0 otherwise.
  - By the product rule, there are $2^{|S|}$ such bit strings, and therefore $2^{|S|}$ subsets.
- Example 6: Product Rule in Terms of Sets
  - Let $A_1, A_2, \ldots, A_m$ be finite sets
  - 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: $|A_1 \times A_2 \times \cdots \times A_m| = |A_1| \cdot |A_2| \cdots |A_m|$

- Example 7: DNA and Genomes
  - A gene is a segment of a DNA molecule that encodes a particular protein and the entirety of genetic information of an organism is called its genome.
  - DNA molecules consist of two strands of blocks known as nucleotides.
  - Each nucleotide is composed of bases: adenine(A), cytosine(C), guanine(G), thymine(T).
  - The DNA of bacteria has between 10^5 and 10^7 links (one of the four bases).
  - Mammals have between 10^8 and 10^{10} links.
  - So, by the product rule there are at least 4^{10^5} different sequences of bases in the DNA of bacteria and 4^{10^8} different sequences of bases in the DNA of mammals.
  - The human genome includes approximately 23,000 genes, each with 1,000 or more links.

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,
  - 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.
- 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.
  - There are $37 + 83 = 120$ possible ways to pick a representative.
- 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

Combining the Sum and Product Rule

- Example 1
  - 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.
  - Use the product rule: $26 + 26 \cdot 10 = 286$
- Example 2: Counting Passwords
  - Each user on a computer system has a password

- which is 6 to 8 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?
- Let P be the total number of passwords
- Let P_6, P_7, and P_8 be the passwords of length 6, 7, and 8.
- By the sum rule $P = P_6 + P_7 + P_8$.
- To find each of P_6, P_7, and P_8, 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, $P = P_6 + P_7 + P_8 = 2{,}684{,}483{,}063{,}360$.

Basic Counting Principles: Subtraction Rule

- The 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:
  - $|A \cup B| = |A| + |B| - |A \cap B|$
- Example: Counting Bit Strings
  - How many bit strings of length eight either start with a 1 bit or end with the two bits 00?
  - 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 $128 + 64 - 32 = 160$.

Basic Counting Principles: Division Rule

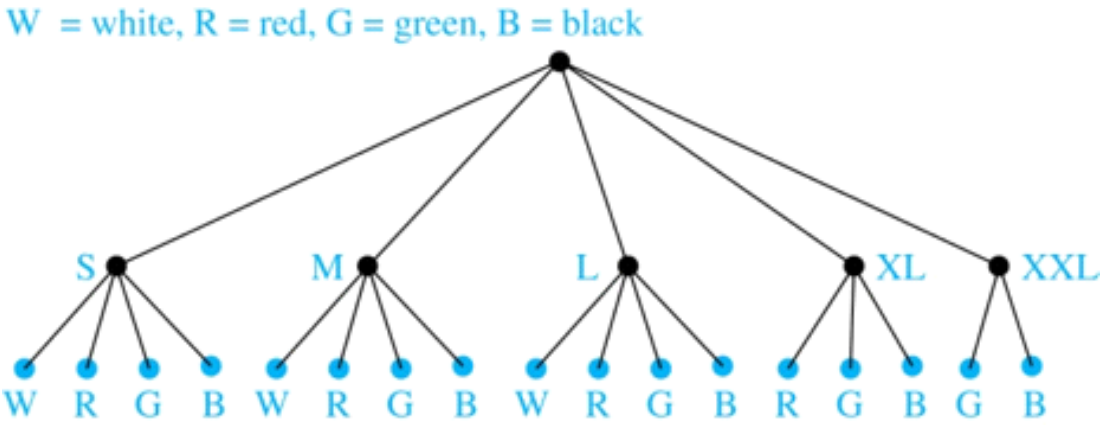
- Division Rule
  - There are n/d ways to do a task if
  - it can be done using a procedure that can be carried out in n ways
  - and for every way w, exactly d of the n ways correspond to way w.
- In terms of sets
  - If the finite set A is the union of n pairwise disjoint subsets each with d elements
  - then $n = |A|/d$.
- In terms of functions
  - If f is a function from A to B, where both are finite sets, and for every value $y \in$ B there

are exactly d values $x \in A$ such that $f(x) = y$, then $|B| = |A|/d$.

- Example
  - How many ways are there to seat four people around a circular table, where two seatings are considered the same when each person has the same left and right neighbor?
  - Number the seats around the table from 1 to 4 proceeding clockwise.
  - There are four ways to select the person for seat 1, 3 for seat 2, 2, for seat 3, and one way for seat 4.
  - Thus there are $4! = 24$ ways to order the four people.
  - But since two seatings are the same when each person has the same left and right neighbor, for every choice for seat 1, we get the same seating.
  - Therefore, by the division rule, there are $24/4 = 6$ different seating arrangements.

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, 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?
  - Draw the tree diagram. The store must stock 17 T-shirts.

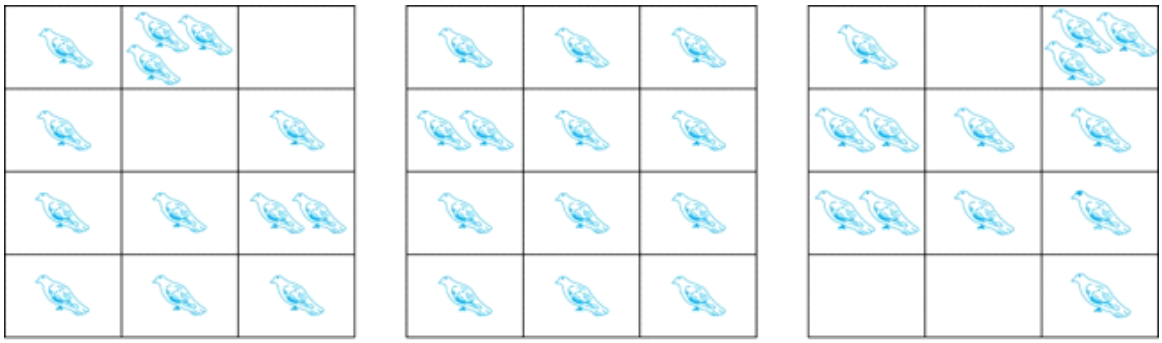


6.2 The Pigeonhole Principle

Monday, April 2, 2018 8:51 AM

The Pigeonhole Principle

- Introduction
  - If a flock of 20 pigeons roosts in a set of 19 pigeonholes
  - Then, one of the pigeonholes must have more than 1 pigeon.



(a) (b) (c)

- 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.
- 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.
- Example
  - Among any group of 367 people, there must be at least two with the same birthday
  - Because there are only 366 possible birthdays.
- Example
  - Show that for every integer n there is a multiple of n that has only 0s and 1s in its

decimal expansion.

- Let n be a positive integer.
- Consider the $n+1$ integers 1, 11, 111, … , 11…1 (where the last has $n+1$ 1s).
- There are n possible remainders when an integer is divided by n.
- By the pigeonhole principle, when each of the $n+1$ integers is divided by n, at least two must have the same remainder.
- Subtract the smaller from the larger and the result is a multiple of n that has only 0s and 1s in its decimal expansion.

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 $\left\lceil \frac{N}{k} \right\rceil$ objects.
- Proof
  - We use a proof by contraposition.
  - Suppose that none of the boxes contains more than $\left\lceil \frac{N}{k} \right\rceil - 1$ objects.
  - Then the total number of objects is at most
  - $k\left(\left\lceil \frac{N}{k} \right\rceil - 1\right) < k\left(\left(\left\lceil \frac{N}{k} \right\rceil + 1\right) - 1\right) = N$
  - where the inequality $\left\lceil \frac{N}{k} \right\rceil < \left\lceil \frac{N}{k} \right\rceil + 1$ has been used.
  - This is a contradiction because there are a total of n objects.
- Example
  - Among 100 people there are at least $\lceil 100/12 \rceil = 9$ who were born in the same month.
- Example
  - How many cards must be selected from a standard deck of 52 cards to guarantee that at least three cards of the same suit are chosen?
  - We assume four boxes; one for each suit.
  - Using the generalized pigeonhole principle, at least one box contains at least $\left\lceil \frac{N}{4} \right\rceil$ cards.
  - At least three cards of one suit are selected if $\left\lceil \frac{N}{4} \right\rceil \geq 3$.
  - The smallest integer N such that $\left\lceil \frac{N}{4} \right\rceil \geq 3$ is $N = 2 \cdot 4 + 1 = 9$.
- Example
  - How many must be selected to guarantee that at least three hearts are selected?
  - A deck contains 13 hearts and 39 cards which are not hearts.
  - If we select 41 cards, we may have 39 cards which are not hearts along with 2 hearts.
  - However, when we select 42 cards, we must have at least three hearts.
  - (Note that the generalized pigeonhole principle is not used here.)

6.3 Permutations and Combinations

Monday, April 2, 2018 8:51 AM

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, $P(3,2) = 6$.

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 $\big(n-(r-1)\big)$ 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
  - $P(n, r) = \dfrac{n!}{(n-r)!}$
- 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?
  - $P(100,3) = 100 \cdot 99 \cdot 98 = 970{,}200$

  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?
- The first city is chosen, and the rest are ordered arbitrarily.
- Hence the orders are: $7! = 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 5040$
- If she wants to find the tour with the shortest path that visits all the cities,
- she must consider 5040 paths!

Example

- How many permutations of the letters ABCDEFGH contain the string ABC ?
- We solve this problem by counting the permutations of six objects, ABC, D, E, F, G, H.
- $6! = 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 720$

Combinations

- Definition
  - An r-combination of elements of a set is an unordered selection of r elements from the set.
  - Thus, an r-combination is simply a subset of the set with r elements.
  - The number of r-combinations of a set with n distinct elements is denoted by $C(n, r)$.
  - The notation $\binom{n}{r}$ is also used and is called a binomial coefficient.
  - (We will see the notation again in the binomial theorem in Section 6.4.)
- Example:
  - Let S be the set {a, b, c, d}.
  - Then {a, c, d} is a 3-combination from S.
  - It is the same as {d, c, a} since the order listed does not matter.
  - C(4,2) = 6 because the 2-combinations of {a, b, c, d} are the six subsets
  - {a, b}, {a, c}, {a, d}, {b, c}, {b, d}, and {c, d}.
- Theorem 2
  - The number of r-combinations of a set with n elements, where $n \geq r \geq 0$, equals
  - $C(n, r) = \dfrac{n!}{(n-r)!\, r!}$
- Proof
  - By the product rule $P(n, r) = C(n, r) \cdot P(r, r)$. Therefore,
  - $C(n, r) = \dfrac{P(n, r)}{P(r, r)} = \dfrac{n!/(n-r)!}{r!/(r-r)!} = \dfrac{n!}{(n-r)!\, r!}$
- 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?
  - 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
- $C(52,47) = \frac{52!}{47!\,5!} = C(52,5) = 2{,}598{,}960$

- 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
  - $C(n,n-r) = \frac{n!}{(n-r)!\,\big(n-(n-r)\big)!} = \frac{n!}{(n-r)!\,r!} = C(n,r)$
  - Hence, $C(n,r) = C(n,n-r)$

Combinatorial Proofs

- Definition
  - A combinatorial proof of an identity is a proof that uses one of the following methods.
  - Double Counting Proof
    - A double counting proof uses counting arguments to prove that
    - both sides of an identity count the same objects, but in different ways.
  - Bijective Proof
    - A bijective proof shows that there is a bijection
    - between the sets of objects counted by the two sides of the identity.
- Example
  - Here are two combinatorial proofs that $C(n,r) = C(n,n-r)$
  - 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.

More Examples

- How many words can you formed by rearranging the letters in the word:

- Combine
  - 7!
- Permutation
  - Pick where t's go
  - Arrange remaining 9 letters
  - $\binom{11}{2} \cdot 9! = \frac{11!}{2!}$
- Rearrange
  - Pick where r's go
  - Pick where e's go
  - Pick where a's go
  - Arrange remaining 2 letters
  - $\binom{9}{3} \cdot \binom{6}{2} \cdot \binom{4}{2} \cdot 2! = \frac{9!}{3!\,2!\,2!}$

- In a game of cards a hand consists of 13 cards. How many possible hands are there with
  - Exactly one ace
    - Pick which ace
    - Pick the rest
    - $4 \cdot \binom{52-4}{12}$
  - At least one ace
    - 1 Ace + 2 Aces + 3 Aces + 4 Aces
    - $4 \cdot \binom{52-4}{12} + \binom{4}{2} \cdot \binom{52-4}{11} + \binom{4}{3} \cdot \binom{52-4}{10} + \binom{4}{4} \cdot \binom{52-4}{9}$
  - Exactly one ace and two diamonds
    - Case 1: We pick one diamond and a diamond ace
      - $1 \cdot 12 \cdot \binom{52-4-12}{11}$
    - Case 2: We pick two diamonds and a different ace
      - $3 \cdot \binom{12}{2} \cdot \binom{52-4-12}{10}$
    - So the total number of hands is $1 \cdot 12 \cdot \binom{52-4-12}{11} + 3 \cdot \binom{12}{2} \cdot \binom{52-4-12}{10}$

6.4 Binomial Coefficients and Identities

Wednesday, April 4, 2018 9:20 AM

Powers of Binomial Expressions

- 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 of $(x + y)^n$ where $n \in \mathbb{Z}^+$.
- 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 $x^3, x^2y. xy^2, y^3$ 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 $(x + y)^3 = x^3 + 3x^2y + 3xy^2 + y^3$

Binomial Theorem

- Binomial Theorem
  - Let x and y be variables, and n a nonnegative integer. Then:
  - $(x + y)^n = \sum_{j=0}^{n} \binom{n}{j} x^{n-j}y^j = \binom{n}{0}x^n + \binom{n}{1}x^{n-1}y + \cdots + \binom{n}{n-1}xy^{n-1} + \binom{n}{n}y^n$
- 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}, it is necessary to choose $n - j$ xs from the n sums.
  - Therefore, the coefficient of x^{n-j} is $\binom{n}{n-j}$ which equals $\binom{n}{j}$.

Using the Binomial Theorem

- What is the coefficient of $x^{12}y^{13}$ in the expansion of $(2x-3y)^{25}$?
- We view the expression as $\left(2x+(-3y)\right)^{25}$.
- By the binomial theorem
  - $\left(2x+(-3y)\right)^{25}=\sum_{j=0}^{25}\binom{25}{j}(2x)^{25-j}(-3y)^{j}$
- Consequently, the coefficient of $x^{12}y^{13}$ in the expansion is obtained when $j=13$.

A Useful Identity

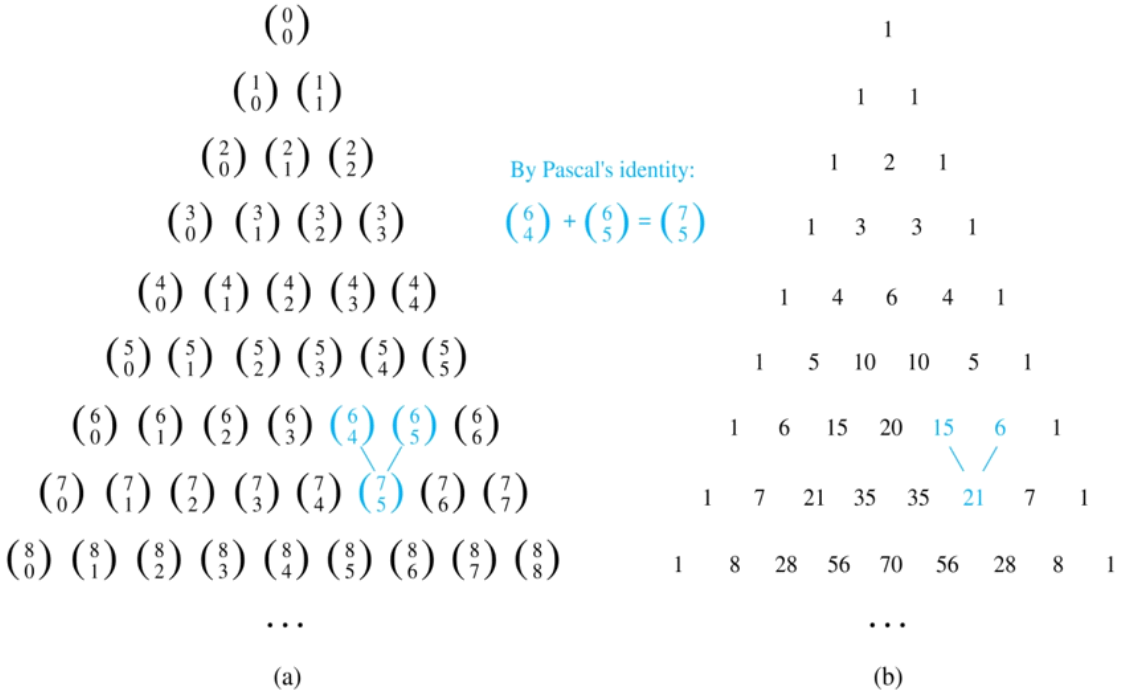
- Corollary 1
  - With $n\geq 0$, $\sum_{k=0}^{n}\binom{n}{k}=2^{n}$
- Proof (using binomial theorem)
  - With $x=1$ and $y=1$, from the binomial theorem we see that:
  - $2^{n}=(1+1)^{2}=\sum_{k=0}^{n}\binom{n}{k}1^{k}1^{n-k}=\sum_{k=0}^{n}\binom{n}{k}$
- Proof (combinatorial)
  - Consider the subsets 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:
  - $\sum_{k=0}^{n}\binom{n}{k}=2^{n}$

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
  - 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
  - $\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$

Pascal's Triangle



- The nth row in the triangle consists of the binomial coefficients $\binom{n}{k}$, $k = 0,1,\ldots,n$
- By Pascal's identity, adding two adjacent binomial coefficients results is
- the binomial coefficient in the next row between these two coefficients.

6.5 Generalized Permutations and Combinations

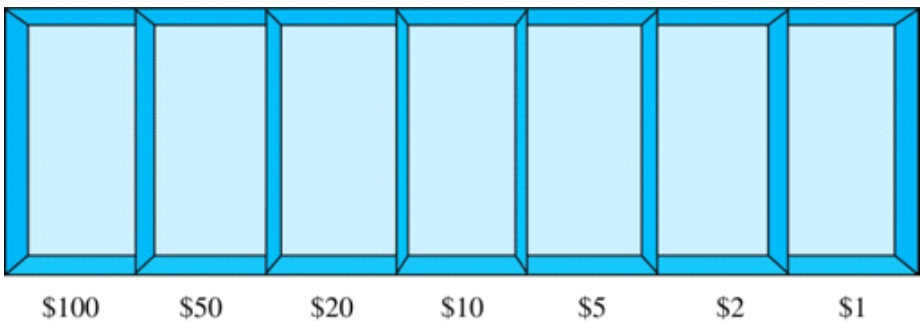
Friday, April 6, 2018 9:09 AM

Permutations with Repetition

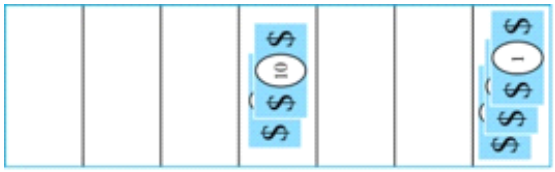
- Theorem 1
  - The number of r-permutations of a set of n objects with repetition allowed is
  - $GP(n, r)n^r$
- Proof
  - There are n ways to select an element of the set
  - for each of the r positions in the r-permutation when repetition is allowed
  - Hence, by the product rule there are n^r r-permutations with repetition.
- Example
  - How many strings of length r can be formed from the uppercase letters of the English alphabet?
  - The number of such strings is 26^r
  - which is the number of r-permutations of a set with 26 elements
- Example 2
  - How many function are there $f: A \to B$ where $|A| = k$, and $|B| = m$? m^k

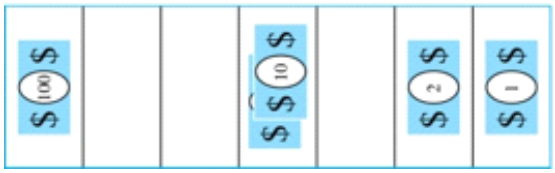
Combinations with Repetition

- Example
  - How many ways are there to select five bills from a box containing at least five of each of the following denominations: \$1, \$2, \$5, \$10, \$20, \$50, and \$100?
  - Place the selected bills in the appropriate position of a cash box illustrated below:



  - Some possible ways of placing the five bills:





- The number of ways to select five bills corresponds to
- the number of ways to arrange six bars and five stars in a row.
- This is the number of unordered selections of 5 objects from a set of 11.
- Hence, there are
  - $C(11,5) = \frac{11!}{5!\,6!} = 462$
- ways to choose five bills with seven types of bills

- Theorem 2
  - The number of r-combinations from a set with n elements when repetition of elements is allowed
  - $C(n+r-1, r) = C(n+r-1, n-1)$
- Proof
  - Each r-combination of a set with n elements with repetition allowed

    can be represented by a list of n – 1 bars and r stars.
  - The bars mark the n cells containing a star for each time the ith element of the set occurs in the combination.
  - The number of such lists is $C(n+r-1, r)$
  - because each list is a choice of the r positions to place the stars
  - from the total of $n+r-1$ positions to place the stars and the bars.
  - This is also equal to $C(n+r-1, n-1)$
  - which is the number of ways to place the n – 1 bars
- Example 1
  - How many solutions does the equation $x_1 + x_2 + x_3 = 11$ $(x_1, x_2, x_3 \in \mathbb{Z}^+)$ have
  - Each solution corresponds to a way to select 11 items from a set with 3 elements
  - x_1 elements of type one, x_2 of type two, and x_3 of type three.
  - By Theorem 2 it follows that the number of solution is

- $C(3+11-1,11) = C(13,11) = C(13,2) = \dfrac{13 \cdot 12}{1 \cdot 2} = 78$

- Example 2
  - Suppose that a cookie shop has four different kinds of cookies.
  - How many different ways can six cookies be chosen?
  - The number of ways to choose six cookies is
  - the number of 6-combinations of a set with four elements.
  - By Theorem 2 the number of ways to choose six cookies from the four kinds is
  - $C(9,6) = C(9,3) = \dfrac{9 \cdot 8 \cdot 7}{1 \cdot 2 \cdot 3} = 84$

Permutations and Combinations with and without Repetition

TABLE 1 Combinations and Permutations With and Without Repetition.

Type	*Repetition Allowed?*	*Formula*
r-permutations	No	$\dfrac{n!}{(n-r)!}$
r-combinations	No	$\dfrac{n!}{r!\,(n-r)!}$
r-permutations	Yes	n^r
r-combinations	Yes	$\dfrac{(n+r-1)!}{r!\,(n-1)!}$

7.1 An Introduction to Discrete Probability

Monday, April 9, 2018 5:30 PM

Probability of an Event

- Introduction
  - We first study Pierre-Simon Laplace's classical theory of probability
  - which he introduced in the 18th century, when he analyzed games of chance.
- Experiment
  - A procedure that yields one of a given set of possible outcomes.
- Sample space
  - The sample space of the experiment is the set of possible outcomes.
- Event
  - An event is a subset of the sample space.
- Probability
  - If S is a finite sample space of equally likely outcomes and E is an event
  - Then the probability of E is $p(E) = \frac{|E|}{|S|}$
  - For every event E, we have $0 \leq p(E) \leq 1$
  - This follows directly from the definition because
  - $0 \leq p(E) = \frac{|E|}{|S|} \leq \frac{|S|}{|S|} \leq 1$, since $0 \leq |E| \leq |S|$

Applying Laplace's Definition

- Example 1
  - An urn contains four blue balls and five red balls.
  - What is the probability that a ball chosen from the urn is blue?
  - The probability that the ball is chosen is $4/9$
  - since there are nine possible outcomes, and four of these produce a blue ball.
- Example 2
  - What is the probability that when two dice are rolled, the sum of the numbers on the two dice is 7?
  - By the product rule there are $62 = 36$ possible outcomes.
  - Six of these sum to 7.
  - Hence, the probability of obtaining a 7 is $6/36 = 1/6$.
- Example 3
  - In a lottery, a player wins a large prize when they pick four digits that match, in correct order, four digits selected by a random mechanical process.
  - What is the probability that a player wins the prize?

- By the product rule there are $10^4 = 10{,}000$ ways to pick four digits.
- Since there is only 1 way to pick the correct digits,
- the probability of winning the large prize is $1/10{,}000 = 0.0001$.

- Example 4
  - A smaller prize is won if only three digits are matched.
  - What is the probability that a player wins the small prize?
  - If exactly three digits are matched, one of the four digits must be incorrect and the other three digits must be correct.
  - For the digit that is incorrect, there are 9 possible choices.
  - Hence, by the sum rule, there a total of 36 possible ways to choose four digits that match exactly three of the winning four digits.
  - The probability of winning the small price is
    - $36/10{,}000 = 9/2500 = 0.0036$

- Example 5
  - There are many lotteries that award prizes to people who correctly choose a set of six numbers out of the first n positive integers, where n is usually between 30 and 60.
  - What is the probability that a person picks the correct six numbers out of 40?
  - The number of ways to choose six numbers out of 40 is
    - $C(40,6) = 40!/(34!6!) = 3{,}838{,}380$.
  - Hence, the probability of picking a winning combination is
    - $1/\ 3{,}838{,}380 \approx 0.00000026$.

- Example 6
  - What is the probability that the numbers 11, 4, 17, 39, and 23 are drawn in that order from a bin with 50 balls labeled with the numbers 1,2, …, 50 if
  - The ball selected is not returned to the bin.
    - Sampling without replacement:
    - The probability is $1/254{,}251{,}200$ since there are
    - $50 \cdot 49 \cdot 47 \cdot 46 \cdot 45 \ = 254{,}251{,}200$ ways to choose the five balls.
  - The ball selected is returned to the bin before the next ball is selected.
    - Sampling with replacement:
    - The probability is $1/50^5 = 1/312{,}500{,}000$ since $50^5 = 312{,}500{,}000$.

The Probability of Complements and Unions of Events

- Theorem 1
  - Let E be an event in sample space S.
  - The probability of the complementary event of E: $\bar{E} = S - E$ is given by
  - $p(\bar{E}) = 1 - p(E)$

- Proof

- Using the fact that $|\bar{E}| = |S| - |E|$
- $p(\bar{E}) = \frac{|S| - |E|}{|S|} = 1 - \frac{|E|}{|S|} = 1 - p(E)$

- Example
  - A sequence of 10 bits is chosen randomly.
  - What is the probability that at least one of these bits is 0?
  - Let E be the event that at least one of the 10 bits is 0.
  - Then $\bar{E}$ is the event that all of the bits are 1s.
  - The size of the sample space S is 210. Hence,
  - $p(E) = 1 - p(\bar{E}) = 1 - \frac{|\bar{E}|}{|S|} = 1 - \frac{1}{2^{10}} = 1 - \frac{1}{1024} = \frac{1023}{1024}$

- Theorem 2
  - Let E_1 and E_2 be events in the sample space S. Then
  - $p(E_1 \cup E_2) = p(E_1) + p(E_2) - p(E_1 \cap E_2)$

- Proof
  - Given the inclusion-exclusion formula from Section 2.2
  - $|A \cup B| = |A| + |B| - |A \cap B|$, it follows that
    - $p(E_1 \cup E_2) = \frac{|E_1 \cup E_2|}{|S|}$
    - $= \frac{|E_1| + |E_2| - |E_1 \cap E_2|}{|S|}$
    - $= \frac{|E_1|}{|S|} + \frac{|E_2|}{|S|} - \frac{|E_1 \cap E_2|}{|S|}$
    - $= p(E_1) + p(E_2) - p(E_1 \cap E_2)$

- Example
  - What is the probability that a positive integer selected at random from the set of positive integers not exceeding 100 is divisible by either 2 or 5?
  - Let E_1 be the event that the integer is divisible by 2
  - Let E_2 be the event that it is divisible 5.
  - Then the event that the integer is divisible by 2 or 5 is $E_1 \cup E_2$
  - And $E_1 \cap E_2$ is the event that it is divisible by 2 and 5.
  - It follows that:
  - $p(E_1 \cup E_2) = p(E_1) + p(E_2) - p(E_1 \cap E_2) = \frac{50}{100} + \frac{20}{100} - \frac{10}{100} = \frac{3}{5}$

Monty Hall Puzzle

- You are asked to select one of the three doors to open.
- There is a large prize behind one of the doors and if you select that door, you win the prize.

- After you select a door, the game show host opens one of the other doors (which he knows is not the winning door).
- The prize is not behind the door and he gives you the opportunity to switch your selection.
- Should you switch?

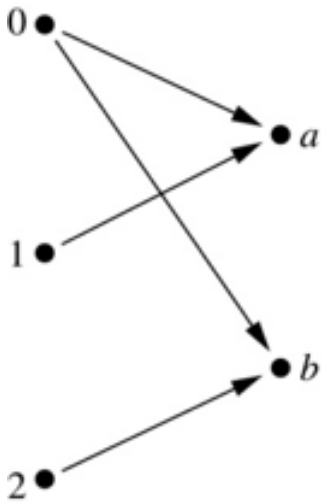
- You should switch.
- The probability that your initial pick is correct is 1/3.
- This is the same whether or not you switch doors.
- But since the game show host always opens a door that does not have the prize,
- If you switch the probability of winning will be 2/3
- Because you win if your initial pick was not the correct door
- And the probability your initial pick was wrong is 2/3.

9.1 Relations and Their Properties

Wednesday, April 11, 2018 8:59 AM

Binary Relations

- Definition
  - A binary relation R from a set A to a set B is a subset $R \subseteq A \times B$.
- 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 this relation graphically or using a table:



R	a	b
0	×	×
1	×	
2		×

- Note
  - Relations are more general than functions
  - A function is a relation where exactly one element of B is related to each element of A

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 1
  - Suppose that $A = \{a, b, c\}$
  - Then $R = \{(a, a), (a, b), (a, c)\}$ is a relation on A
- Example 2
  - Let $A = \{1, 2, 3, 4\}$
  - The ordered pairs in the relation $R = \{(a, b) | a \text{ divides } b\}$ are
  - $\{(1,1), (1, 2), (1,3), (1, 4), (2, 2), (2, 4), (3, 3), (4,4)\}$
- Question: How many relations are there on a set A?
  - Because a relation on A is the same thing as a subset of $A \times A$

- We count the subsets of $A \times A$.
- Since $A \times A$ has n^2 elements when A has n elements
- And a set with m elements has 2^m subsets
- There are $2^{|A|^2}$ subsets of $A \times A$.
- Therefore, there are $2^{|A|^2}$ relations on a set A.

- Example 3
  - Consider these relations on the set of integers:
    - $R_1 = \{(a, b) | a \leq b\}$
    - $R_2 = \{(a, b) | a > b\}$
    - $R_3 = \{(a, b) | a = b \text{ or } a = -b\}$
    - $R_4 = \{(a, b) | a = b\}$
    - $R_5 = \{(a, b) | a = b + 1\}$
    - $R_6 = \{(a, b) | a + b \leq 3\}$
  - Note
    - These relations are on an infinite set and each of these relations is an infinite set
    - R_5 can be viewed as a function
    - Our definition of a function $f: A \to B$ is a subset of $A \times B$
    - Therefore every function is a relation
  - Which of these relations contain each of the pairs
    - (1,1), (1, 2), (2, 1), (1, −1), and (2, 2)?
  - Solution
    - (1,1) is in R_1, R_3, R_4, and R_6
    - (1,2) is in R_1 and R_6
    - (2,1) is in R_2, R_5, and R_6
    - (1, −1) is in R_2, R_3, and R_6
    - (2,2) is in R_1, R_3, and R_4

Reflexive Relations

- Definition
  - R is reflexive if and only if $(a, a) \in R$ for every element $a \in A$
  - Written symbolically, R is reflexive if and only if
  - $\forall x[x \in U \to (x, x) \in R]$
- Note
  - If $A = \emptyset$ then the empty relation is reflexive vacuously.
  - That is the empty relation on an empty set is reflexive!
- Example
  - The following relations on the integers are reflexive:

        - $R_1 = \{(a,b)|a \leq b\}$
        - $R_3 = \{(a,b)|a = b \text{ or } a = -b\}$
        - $R_4 = \{(a,b)|a = b\}$
    - The following relations are not reflexive:
        - $R_2 = \{(a,b)|a > b\}$ (note that $3 \not> 3$)
        - $R_5 = \{(a,b)|a = b + 1\}$ (note that $3 \neq 3 + 1$)
        - $R_6 = \{(a,b)|a + b \leq 3\}$ (note that $4 + 4 \not\leq 3$)

Antireflexive Relations

- Definition
    - R is antireflexive if and only if $(a,a) \notin R$ for every element $a \in A$
    - Written symbolically, R is antireflexive if and only if
    - $\forall x[x \in U \rightarrow (x,x) \notin R]$
- Note
    - Antireflexive is different from not reflexive
- Example
    - The following relations on the integers are antireflexive
        - $R_2 = \{(a,b)|a > b\}$
        - $R_5 = \{(a,b)|a = b + 1\}$
    - $R_6 = \{(a,b)|a + b \leq 3\}$ is neither reflexive nor antireflexive

Symmetric Relations

- Definition
    - R is symmetric if and only if $(b,a) \in R$ whenever $(a,b) \in R, \forall a,b \in A$
    - Written symbolically, R is symmetric if and only if
    - $\forall x \forall y[(x,y) \in R \rightarrow (y,x) \in R]$
- Example
    - The following relations on the integers are symmetric:
        - $R_3 = \{(a,b)|a = b \text{ or } a = -b\}$
        - $R_4 = \{(a,b)|a = b\}$
        - $R_6 = \{(a,b)|a + b \leq 3\}$
    - The following are not symmetric:
        - $R_1 = \{(a,b)|a \leq b\}$ (note that $3 \leq 4$, but $4 \not\leq 3$)
        - $R_2 = \{(a,b)|a > b\}$ (note that $4 > 3$, but $3 \not> 4$)
        - $R_5 = \{(a,b)|a = b + 1\}$ (note that $4 = 3 + 1$, but $3 \neq 4 + 1$)

Antisymmetric Relations

- Definition
    - R is antisymmetric if and only if $a = b$ whenever $(a,b),(b,a) \in R, \forall a,b \in A$

- Written symbolically, R is antisymmetric if and only if
- $\forall x \forall y \left[(x, y) \in R \wedge (y, x) \in R \rightarrow x = y \right]$

- Note
  - For any integer, if $a \geq b$ and $a \leq b$, then $a = b$
- Example
  - The following relations on the integers are antisymmetric:
    - $R_1 = \{(a, b) | a \leq b\}$
    - $R_2 = \{(a, b) | a > b\}$
    - $R_4 = \{(a, b) | a = b\}$
    - $R_5 = \{(a, b) | a = b + 1\}$
  - The following relations are not antisymmetric:
    - $R_3 = \{(a, b) | a = b \text{ or } a = -b\}$ (note that $(1, -1), (-1,1) \in R_3$)
    - $R_6 = \{(a, b) | a + b \leq 3\}$ (note that $(1,2), (2,1) \in R_6$)

Transitive Relations

- Definition
  - R is transitive if and only if $(a, c) \in R$ whenever $(a, b), (b, c) \in R, \forall a, b, c \in A$
  - Written symbolically, R is transitive if and only if
  - $\forall x \forall y \forall z \left[(x, y) \in R \wedge (y, z) \in R \rightarrow (x, z) \in R \right]$
- Note
  - For every integer, $a \leq b$ and $b \leq c$, then $b \leq c$
- Example
  - The following relations on the integers are transitive:
    - $R_1 = \{(a, b) | a \leq b\}$
    - $R_2 = \{(a, b) | a > b\}$
    - $R_3 = \{(a, b) | a = b \text{ or } a = -b\}$
    - $R_4 = \{(a, b) | a = b\}$
  - The following are not transitive:
    - $R_5 = \{(a, b) | a = b + 1\}$ (note that $(3,2), (4,3) \in R_5$, but $(3,3) \notin R_5$),
    - $R_6 = \{(a, b) | a + b \leq 3\}$ (note that $(2,1), (1,2) \in R_6$, but $(2,2) \notin R_6$).

Combining Relations

- Given two relations R_1 and R_2
- We can combine them using basic set operations to form new relations
- Example
  - Let $A = \{1,2,3\}$ and $B = \{1,2,3,4\}$
  - The relations $R_1 = \{(1,1),(2,2),(3,3)\}$ and $R_2 = \{(1,1),(1,2),(1,3),(1,4)\}$ can be combined using basic set operations to form new relations:
  - $R_1 \cup R_2 = \{(1,1),(1,2),(1,3),(1,4),(2,2),(3,3)\}$

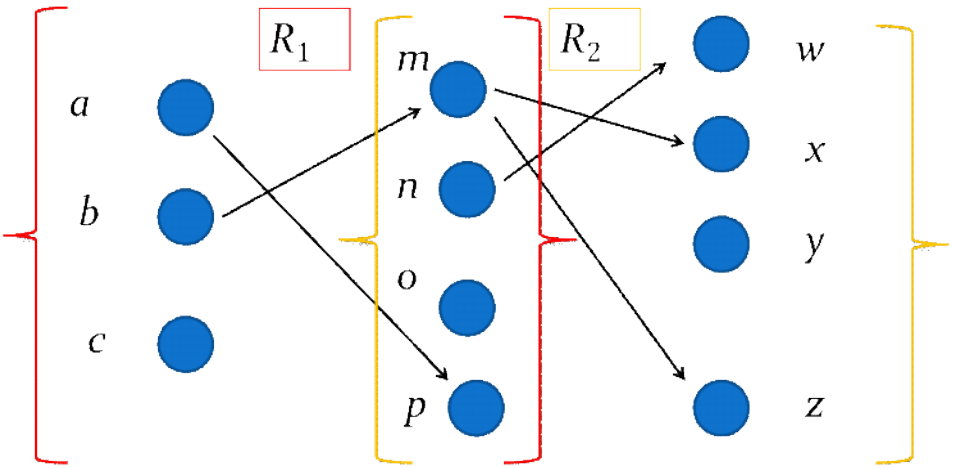
- $R_1 \cap R_2 = \{(1,1)\}$
- $R_1 - R_2 = \{(2,2),(3,3)\}$
- $R_2 - R_1 = \{(1,2),(1,3),(1,4)\}$

Inverse

- Definition
  - Let R be a relation from A to B
  - The inverse of R is the relation
    - $R^{-1} = \{(a,b)|(b,a) \in R\}$
- Proposition
  - R is symmetric if and only if $R = R^{-1}$

Composition

- 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 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 $R_2 \circ R_1$
- Example



  - $R_2 \circ R_1 = \{(b,x),(b,z)\}$

Powers of a Relation

- Definition
  - Let R be a binary relation on A
  - Then the powers R_n of the relation R can be defined inductively by:
  - Basis Step: $R_1 = R$
  - Inductive Step: $R_{n+1} = R_n \circ R$

- The powers of a transitive relation are subsets of the relation
- This is established by the following theorem:
- Theorem 1
    - The relation R on a set A is transitive iff $R_n \subseteq R$ for $n = 1,2,3\ldots$
    - (see the text for a proof via mathematical induction)

9.3 Representing Relations

Friday, April 13, 2018 9:31 AM

Representing Relations Using Matrices

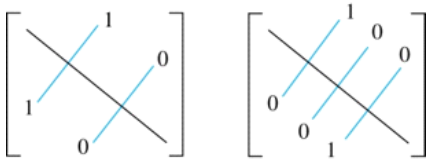
- A relation between finite sets can be represented using a zero-one matrix.
- Suppose R is a relation from $A = \{a_1, a_2, \dots, a_m\}$ to $B = \{b_1, b_2, \dots, b_n\}$
  - The elements of the two sets can be listed in any arbitrary order
  - When $A = B$, we use the same ordering.
- The relation 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
  - a 0 if a_i is not related to b_j.

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 $a > b$.
  - What is the matrix representing R (with increasing numerical order)
- Solution
  - Because $R = \{(2,1), (3,1), (3,2)\}$, the matrix is
  - $M_R = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$
- Example 2
  - Let $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3, b_4, b_5\}$.
  - Which ordered pairs are in the relation R represented by the matrix
  - $M_R = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$
- Solution
  - Because R consists of those ordered pairs (a_i, b_j) with $m_{ij} = 1$
  - $R = \{(a_1, b_2), (a_2, b_1), (a_2, b_3), (a_2, b_4), (a_3, b_1), \{(a_3, b_3), (a_3, b_5)\}$

Matrices of Relations on Sets

- If R is a reflexive relation, all the elements on the main diagonal of M_R are equal to 1
- R is a symmetric relation, if and only if $m_{ij} = 1$ whenever $m_{ji} = 1$
- R is an antisymmetric relation, if and only if $m_{ij} = 0$ or $m_{ji} = 0$ when $i \neq j$



(a) Symmetric (b) Antisymmetric

Example of a Relation on a Set

- Example 3: Suppose that the relation R on a set is represented by the matrix
  - $M_R = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$
- Is R reflexive, symmetric, and/or antisymmetric?
- Because all the diagonal elements are equal to 1, R is reflexive
- Because M_R is symmetric, R is symmetric
- R not antisymmetric because both $m_{1,2}$ and $m_{2,1}$ are 1

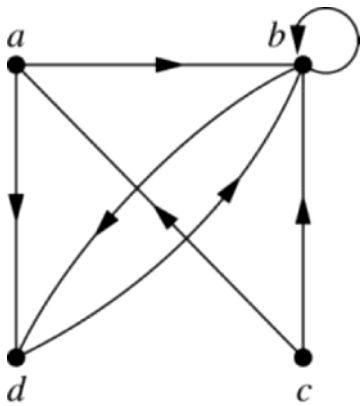
Representing Relations Using Digraphs

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)
- The vertex b is called the terminal vertex of this edge.
- An edge of the form (a, a) is called a loop.

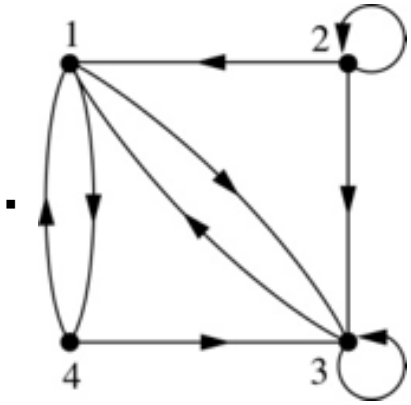
Example 1

- 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.



Example 2

- What are the ordered pairs in the relation represented by this directed graph?



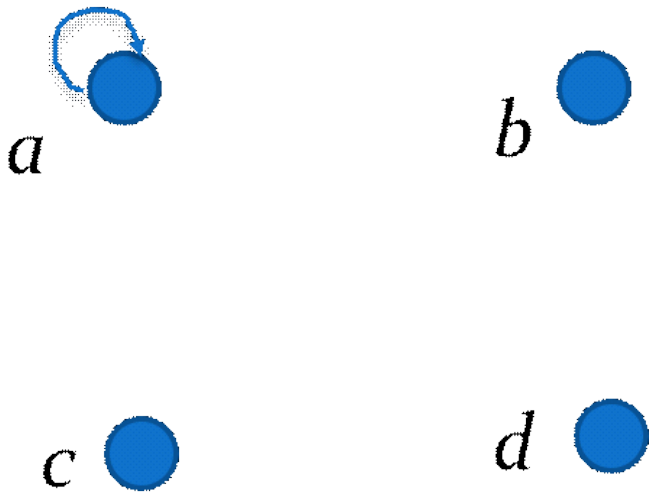
- The ordered pairs in the relation are
- (1, 3), (1, 4), (2, 1), (2, 2), (2, 3), (3, 1), (3, 3), (4, 1), (4, 3)

Determining which Properties a Relation has from its Digraph

- Reflexivity: A loop must be present at all vertices in the graph.
- Symmetry: If (x, y) is an edge, then so is (y, x).
- Antisymmetry: If (x, y) with $x \neq y$ is an edge, then (y, x) is not an edge.
- Transitivity: If (x, y) and (y, z) are edges, then so is (x, z)

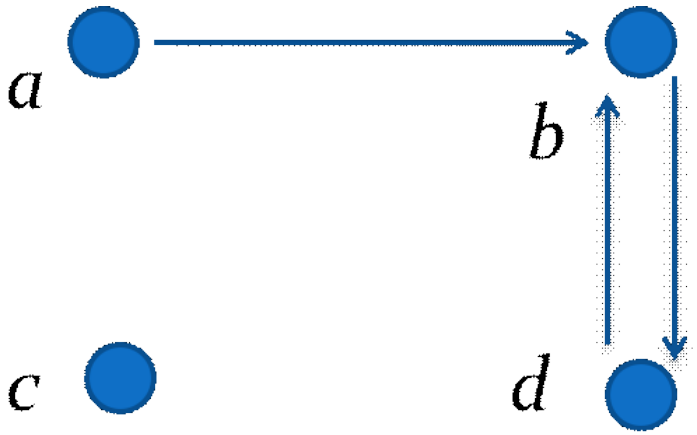
Determining which Properties a Relation has from its Digraph

- Example 1



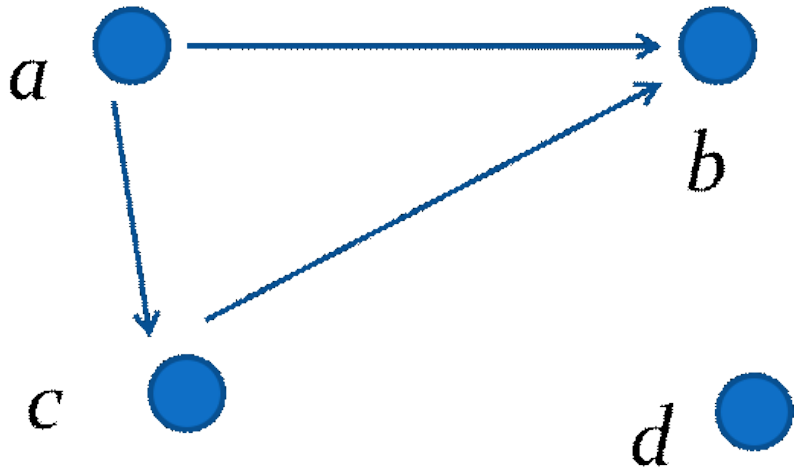
- Reflexive? No, not every vertex has a loop
- Symmetric? Yes (trivially), there is no edge from one vertex to another
- Antisymmetric? Yes (trivially), there is no edge from one vertex to another
- Transitive? Yes, (trivially) since there is no edge from one vertex to another

- Example 2



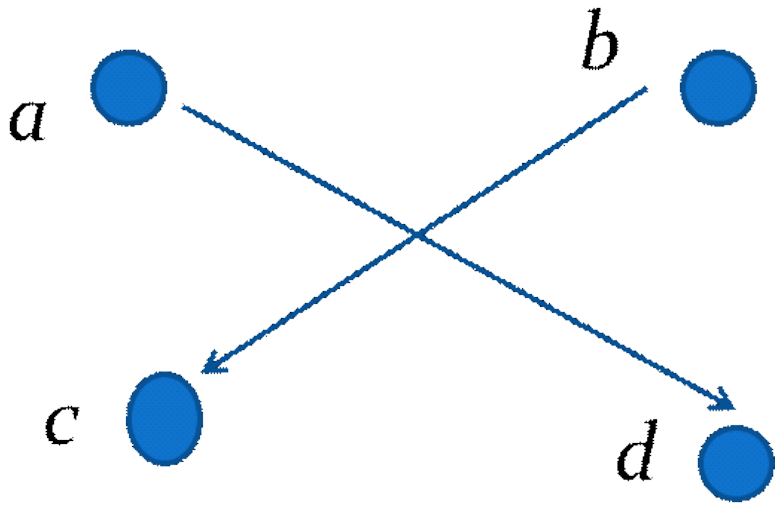
- Reflexive? No, there are no loops
- Symmetric? No, there is an edge from a to b, but not from b to a
- Antisymmetric? No, there is an edge from d to b and b to d
- Transitive? No, there are edges from a to c and from c to b, but there is no edge from a to d

- Example 3



- Reflexive? No, there are no loops
- Symmetric? No, for example, there is no edge from c to a
- Antisymmetric? Yes, whenever there is an edge from one vertex to another, there is not one going back
- Transitive? No, there is no edge from a to b

- Example 4
  - Reflexive? No, there are no loops
  - Symmetric? No, for example, there is no edge from d to a
  - Antisymmetric? Yes, whenever there is an edge from one vertex to another, there is not one going back
  - Transitive? Yes (trivially), there are no two edges where the first edge ends at the vertex where the second edge begins

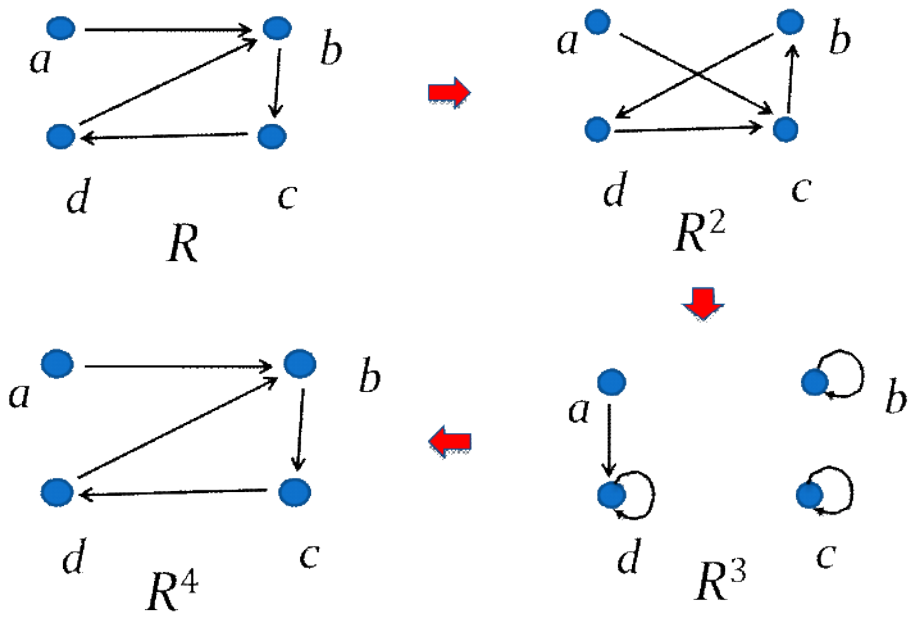


Closures

- Definition
  - The closure of a relation R on A with respect to property P
  - is the least relation on A that contains R and has property P
- Note
  - Least relation R' on A s.t.
  - $R \subseteq R'$
  - R' has property P
  - If S is a relation that safisfies the condition above, then $R' \subseteq S$
- Example
  - The reflexive closure of R is just $R \cup \{(a,a)|a, \in A\}$
  - The symmetric closure of R is $R \cup R^{-1}$
  - The transitive closure of R is $R \cup R^2 \cup R^3 \cup \cdots$

Path in Directed Graphs

- Definition
  - A path from a to b is a directed graph G is a sequence of edges
  - $(a = x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, x_n = b)$ where $n > 0$
  - We denote the path by $x_0, x_1, \ldots, x_n$ say that the path has length n
- Theorem
  - Let R be a relation on a set A
  - There is a path of length n from a to b if and only if (a, b) is an element of R^n



The Connectivity Relation

- Definition
  - Let R be a relation on A
  - The connectivity relation R^* cibsusts if all elements (a, b) s.t.
  - There is a path from a to b in R
  - In other words, R^* is the union of $R, R^2, R^3, \ldots$
  - $R^* = \bigcup_{i=1}^{\infty} R^{(i)}$
- Example 1
  - Let R be the relation between US state such that (a, b) is in R if a and b share a border. What is R^*?
  - All pairs of states except Alaska and Hawaii
- Example 2
  - Let R be the relation between integers s.t. (a, b) is in R if $b = a + 1$
  - What is R^2, R^n, R^*
  - $R^2 = \{(a, b) | b = a + 2\}$
  - $R^n = \{(a, b) | b = a + n\}$
  - $R^* = \{(a, b) | a < b\}$
- Transitive closure
  - The connectivity relation R^* is exactly the transitive closure of R
  - We need to show that R is a subset of R^*, R^* is transitive and least with that property.
  - The first two are easy
  - Let S be a transitive relation containing R

- By induction we show that S contains R^n for every n

Computing The Connectivity Relation

- Theorem
  - Let R be a relation on A and let n be the number of elements in A.
  - The connectivity relation R^* is the union of $R, R^2, \dots, R^n$
- Proof
  - Let (a, b) be the element of R^*
  - Let $a_0 = x_0, x_1, \dots, x_m = b$ be the shortest path witnessing this.
  - If $m > n$, then two of the vertices among $x_1, \dots, x_m$ must be the same, say $x_i = x_j$
  - But then we can find a shorter path $x_0, x_1, \dots, x_i = x_j, x_{j+1}, x_n$
- Corollary
  - $M_{R^*} = M_R \vee M_R^{[2]} \vee \cdots \vee M_R^{[n]}$
- Example
  - Compute M_{R^*} for the relation $R = \{(a, b), (b, c), (c, d), (d, b)\}$
  - $M_R = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$
  - $M_R^{[2]} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \odot \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
  - Similarly, compute $M_R^{[3]}, M_R^{[4]}$
  - Then $M_{R^*} = M_R \vee M_R^{[2]} \vee M_R^{[3]} \vee M_R^{[4]}$

9.5 Equivalence Relations

Friday, April 20, 2018 9:06 AM

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, 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.

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.

Congruence Modulo m

- Example
  - Let m be an integer with $m > 1$
  - Show that the relation
  - $R = \{(a, b) | a \equiv b \pmod{m}\}$
  - 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, $a \equiv c \pmod{m}$

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.

Equivalence Classes

- Let R be an equivalence relation on a set A.
- The set of all elements that are related to an element a of A is called the equivalence class of a
- The equivalence class of a with respect to R is denoted by $[a]_R$.
- When only one relation is under consideration, we can write $[a]$, without the subscript R
- Note that $[a]_R = \{s|(a, s) \in R\}$
- If $b \in [a]_R$, then b is called a representative of this equivalence class.
- Any element of a class can be used as a representative of the class.
- The equivalence classes of the relation congruence modulo m are called the congruence

classes modulo m.

- The congruence class of an integer a modulo m is denoted by $[a]_m$
- So $[a]_m = \{\dots, a-2m, a-m, a+2m, a+2m, \dots\}$
- For example,
  - $[0]_4 = \{\dots, -8, -4, 0, 4, 8, \dots\}$
  - $[1]_4 = \{\dots, -7, -3, 1, 5, 9, \dots\}$
  - $[2]_4 = \{\dots, -6, -2, 2, 6, 10, \dots\}$
  - $[3]_4 = \{\dots, -5, -1, 3, 7, 11, \dots\}$

Equivalence Classes and Partitions

- Theorem 1
  - Let R be an equivalence relation on a set A.
  - These statements for elements a and b of A are equivalent:
    i) aRb
    ii) $[a] = [b]$
    iii) $[a] \cap [b] = \emptyset$
- Proof
  - We show that (i) implies (ii).
  - Assume that aRb.
  - Now suppose that $c \in [a]$.Then aRc. Because aRb and R is symmetric, bRa.
  - Because R is transitive and bRa and aRc, it follows that bRc.
  - Hence, $c \in [b]$. Therefore, $[a] \subseteq [b]$.
  - A similar argument (omitted here) shows that $[b] \subseteq [a]$.
  - Since $[a] \subseteq [b]$ and $[b] \subseteq [a]$, we have shown that $[a] = [b]$.

Partition of a Set

- A partition of a set S is a collection of disjoint nonempty subsets of S that have S as their union.
- In other words, the collection of subsets A_i, where $i \in I$, forms a partition of S if and only if
  - $A_i \neq \emptyset$ for $i \in I$,
  - $A_i \cap A_j = \emptyset$ when $i \neq j$,
  - $\bigcup_{i \in I} A_i = S$

An Equivalence Relation Partitions a Set

- Let R be an equivalence relation on a set A.
- The union of all the equivalence classes of R is all of A
- Since an element a of A is in its own equivalence class $[a]_R$. In other words,

- $\bigcup_{a \in A} [a]_R = A$

- From Theorem 1, it follows that these equivalence classes are either equal or disjoint
- So $[a]_R \cap [b]_R = \emptyset$ when $[a]_R \neq [b]_R$.
- Therefore, the equivalence classes form a partition of A
- Because they split A into disjoint subsets.

Equivalence Relation and Partition

- Theorem 2
  - Let R be an equivalence relation on a set S.
  - Then the equivalence classes of R form a partition of S.
  - Conversely, given a partition $\{A_i | i \in I\}$ of the set S
  - There is an equivalence relation R that has the sets A_i, $i \in I$, as its equivalence classes.
- Proof
  - We have already shown the first part of the theorem.
  - For the second part, assume that $\{A_i | i \in I\}$ is a partition of S.
  - Let R be the relation on S consisting of the pairs (x, y)
  - where x and y belong to the same subset A_i in the partition.
  - We must show that R satisfies the properties of an equivalence relation.
  - Reflexivity
    - For every $a \in S$, $(a, a) \in R$, because a is in the same subset as itself.
  - Symmetry
    - If $(a, b) \in R$, then b and a are in the same subset of the partition, so $(b, a) \in R$
  - Transitivity
    - If $(a, b) \in R$ and $(b, c) \in R$, then a and b are in the same subset of the partition, as are b and c.
    - Since the subsets are disjoint and b belongs to both, the two subsets of the partition must be identical.
    - Therefore, $(a, c) \in R$ since a and c belong to the same subset of the partition.

9.6 Partial Orderings

Monday, April 23, 2018 9:11 AM

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.
- 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 $a \geq c$.
- 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.
  - $(Z^+, |)$ is a poset.
- 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 $A \subseteq C$.

Comparability

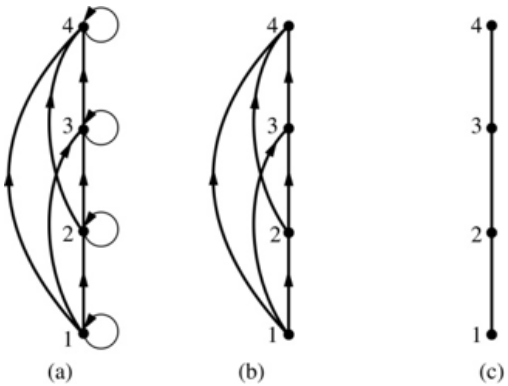
- Definition 2
  - The elements a and b of a poset $(S, \preccurlyeq)$ are comparable if either $a \preccurlyeq b$ or $b \preccurlyeq a$.

- $a, b \in S$ so that neither $a \preccurlyeq b$ nor $b \preccurlyeq a$, then a and b are called incomparable.

- Definition 3
  - If $(S, \preccurlyeq)$ is a poset and every two elements of S are comparable
  - Then S is called a totally ordered or linearly ordered set
  - And $\preccurlyeq$ is called a total order or a linear order.
  - A totally ordered set is also called a chain.
- Definition 4
  - (S,$\preccurlyeq$) is a well-ordered set if it is a poset such that
    - $\preccurlyeq$ is a total ordering
    - every nonempty subset of S has a least element.

Hasse Diagrams

- A Hasse diagram is a visual representation of a partial ordering that leaves out edges that must be present because of the reflexive and transitive properties.



- A partial ordering is shown in (a) of the figure above.
- The loops due to the reflexive property are deleted in (b).
- The edges that must be present due to the transitive property are deleted in (c).
- The Hasse diagram for the partial ordering (a), is depicted in ©

Lexicographic Order

- Definition
  - Given two posets $(A_1, \preccurlyeq_1)$ and $(A_2, \preccurlyeq_2)$
  - The lexicographic ordering on $A_1 \times A_2$ is defined by specifying that
  - (a_1, a_2) is less than $(b1, b2)$, that is, $(a_1, a_2) \prec (b_1, b_2)$,
  - either if $a_1 \prec_1 b_1$ or if $a_1 = b_1$ and $a_2 \prec_2 b_2$.
  - This definition can be easily extended to a lexicographic ordering on strings (see text).
- Example
  - Consider strings of lowercase English letters
  - A lexicographic ordering can be defined using the ordering of the letters in the alphabet
  - This is the same ordering as that used in dictionaries.
  - discreet $\prec$ discrete, because these strings differ in the seventh position and $e \prec t$.

- discreet $\prec$ discreetness, because the first eight letters agree, but the second string is longer.

Well Ordered Induction

- Theorem
  - If $(S, \leq)$ is a well ordered poset and P is a property s.t.
  - If $P(y)$ is true for all $y < x$, then $P(x)$ is true
  - Then P is true for all elements in the poset
- Example
  - Suppose that $a_{m,n}$ is defined for $(m, n) \in \mathbb{N} \times \mathbb{N}$
    - $a_{0,0} = 0$
    - $a_{m,n} = \begin{cases} a_{m-1,n} + 1 & \text{if } n = 0 \text{ and } m > 0 \\ a_{m,n-1} + n & \text{if } n > 0 \end{cases}$
  - Show that $a_{m,n} = m + \frac{n(n+1)}{2}$ is defined for all $(m, n) \in \mathbb{N} \times \mathbb{N}$
- Solution
  - Use induction
  - Basis Step
    - $a_{0,0} = 0 + \dfrac{0 \cdot 1}{2}$
  - Inductive Step
    - Assume that $a_{m',n'} = m' + \frac{n'(n'+1)}{2}$ whenever
    - (m', n') is less than (m, n) in the lexicographic ordering of $\mathbb{N} \times \mathbb{N}$
    - If $n = 0$, by the inductive hypothesis, we can conclude
      - $a_{m,n} = a_{m-1,n} + 1 = m - 1 + \dfrac{n(n+1)}{2} + 1 = m + \dfrac{n(n+1)}{2}$
    - If $n > 0$, by the inductive hypothesis, we can conclude
      - $a_{m,n} = a_{m-1,n} + 1 = m + \dfrac{n(n-1)}{2} + n = m + \dfrac{n(n+1)}{2}$

Maximal and Minimal Elements

- Definition
  - If $(S, \leq)$ is a ordered poset then an element a is
  - Minimal if there is no element b s.t. $b \leq a$, and b is not equal to a
  - Maximal if there is no element b s.t. $a \leq b$, and b is not equal to a
  - Greatest if for every element $b \in S$, $b \leq a$
  - Least if for every element $b \in S$, $a \leq b$
- Intuition
  - Minimal: No element is strictly smaller
  - Least: Every other element is bigger
  - Maximal: No element is strictly bigger

- Greatest: Every other element is smaller

- Example 1
  - Let {2,4,5,10,12,20,25} ordered by divisibility relation
  - Which element of {2,4,5,10,12,20,25} are maximal and which are minimal?
  - Is there a least or greatest element?
  - Maximal: 12, 20, 25
  - Minimal: 2,5
  - No least or greatest element.
- Example 2
  - Given an example of a poset with no minimal element and two maximal elements?
  - $\{1,1^*,0,-1,-2,-3,\dots\}$ where $1 \geq 0 \geq -1 \geq -2 \geq \cdots$, and $1^* \geq 0 \geq -1 \geq -2 \geq \cdots$

Topological Sorting

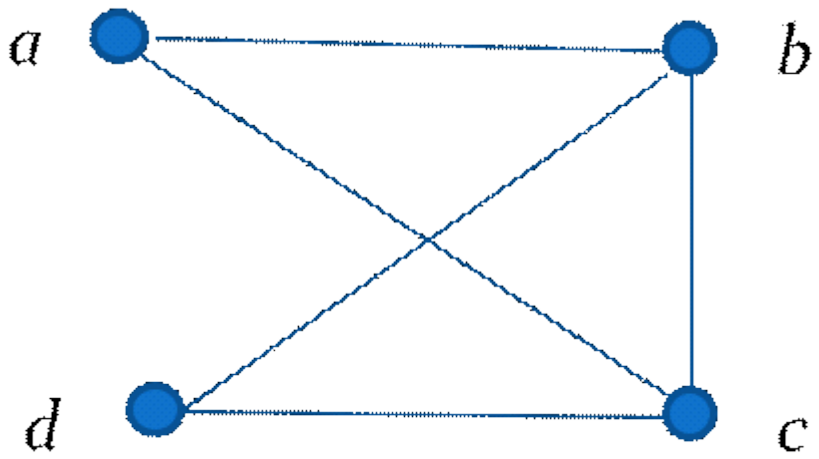
- Definition
  - If (S,R) is a partial ordering and (S,R^*) is a total ordering then we say that
  - The two are compatible if R is a subset of R^*
- Theorem
  - Let S be a finite set
  - Every partial order R on S has a compatible total order R^*
- Proof
  - Every nonempty finite partial order has a minimal element
  - This can be proved by induction on the size of S
  - We should build a total ordering of S be selecting
    - a_1 to be a minimal element of S
    - a_2 to be the minimal element of $S - \{a_1\}$
    - a_i to be the minimal element of $S - \{a_1, a_2, \dots, a_{i-1}\}$
    - a_n to be the least available element of S
  - We set $R^* = \{(a_i, a_i) | j \geq i\}$

10.1 Graphs and Graph Models

Friday, April 27, 2018 9:05 AM

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.

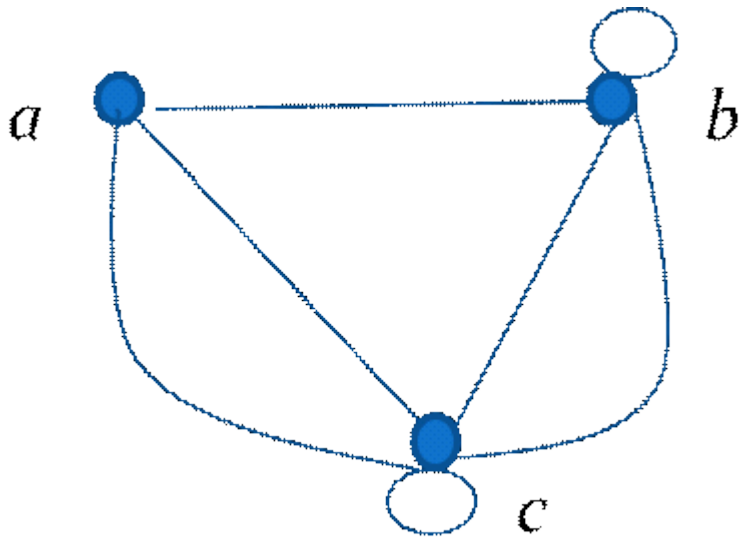


- Remarks
  - The graphs we study here are unrelated to graphs of functions studied in Chapter 2.
  - We have a lot of freedom when we draw a picture of a graph.
  - All that matters is the connections made by the edges, not the particular geometry depicted.
  - For example, the lengths of edges, whether 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.

Graph Terminology

- Definition
  - 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.

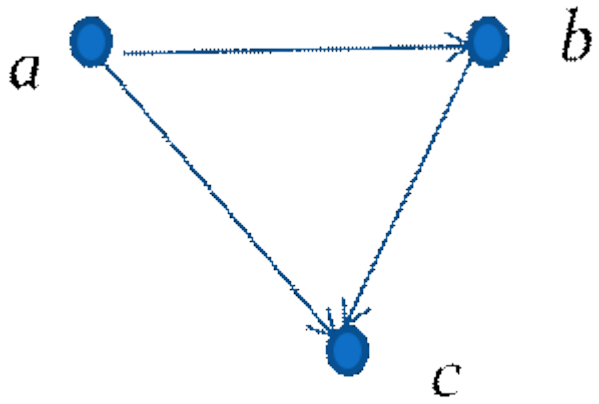


Directed Graphs

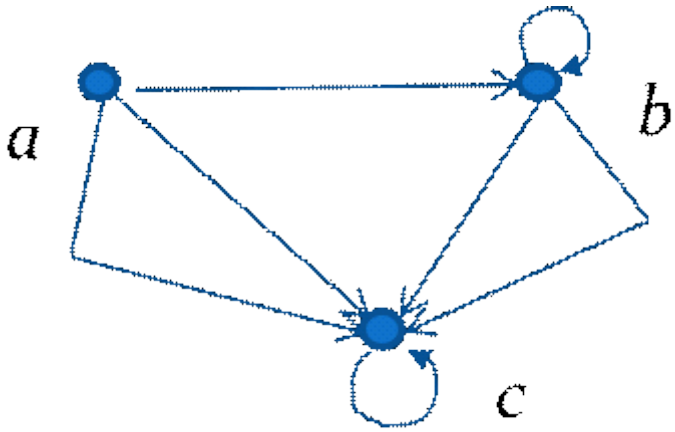
- Definition:
    - An directed graph (or digraph) $G = (V, E)$ consists of
        - a nonempty set V of vertices (or nodes)
        - and a set E of directed edges (or arcs).
    - Each edge is associated with an ordered pair of vertices.
    - The directed edge associated with the ordered pair (u, 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.

Graph Terminology (continued)

- Definition
    - A simple directed graph has no loops and no multiple edges.
- Example
    - This is a directed graph with three vertices and four edges.

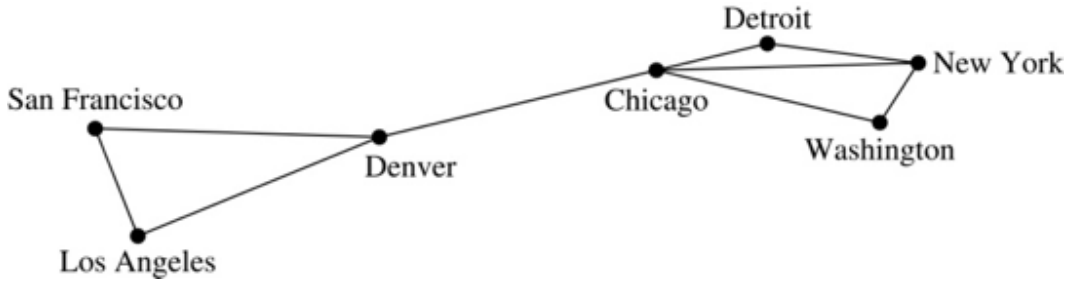


- Definition
  - A directed multigraph may have multiple directed edges.
  - When there are m directed edges from the vertex u to the vertex v,
  - We say that (u, v) is an edge of multiplicity m.
- Example
  - In this directed multigraph the multiplicity of (a, b) is 1 and the multiplicity of (b, c) is 2.

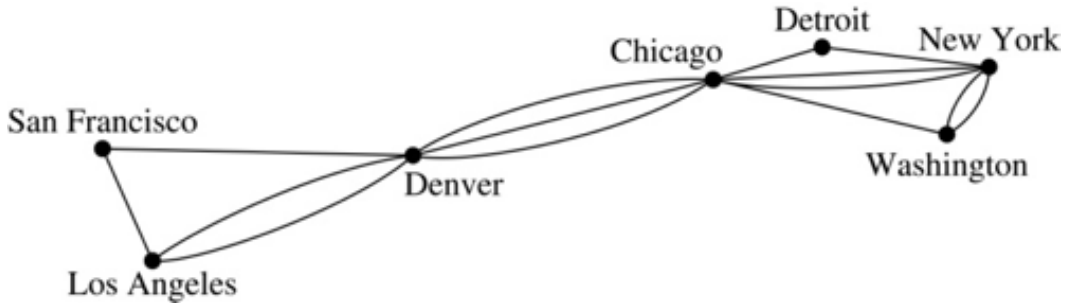


Graph Models: Computer Networks

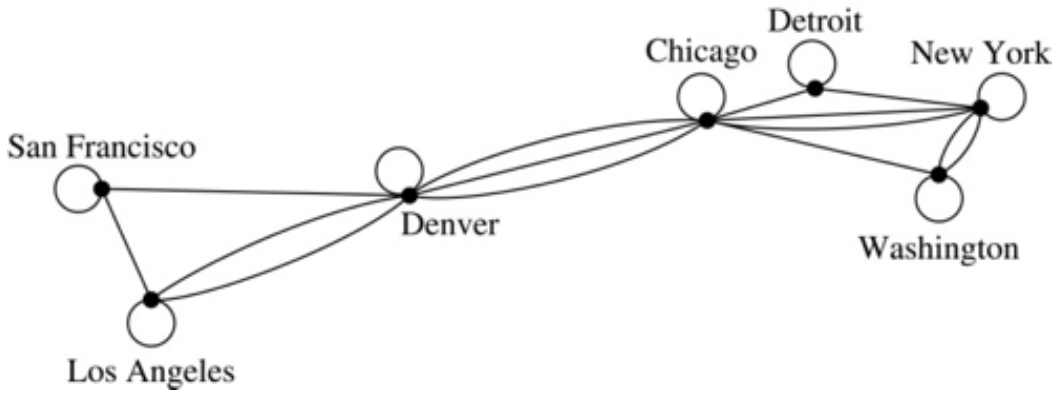
- When we build a graph model, we use the appropriate type of graph to capture the important features of the application.
- We illustrate this process using graph models of different types of computer networks.
- In all these graph models, the vertices represent data centers and the edges represent communication 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.



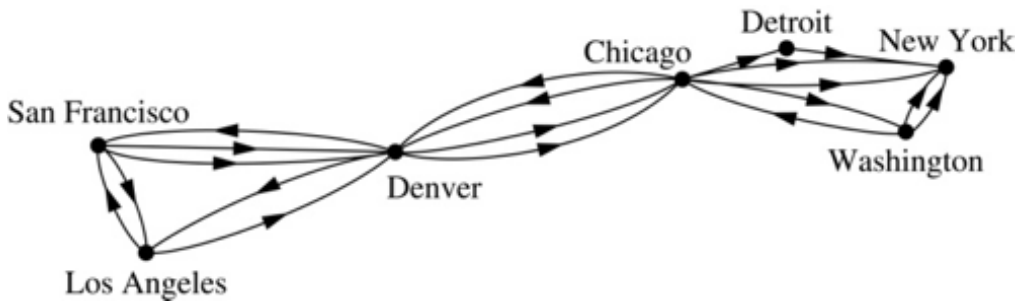
- To model a computer network where we care about the number of links between data centers, we use a multigraph.



- To model a computer network with diagnostic links at data centers, we use a pseudograph, as loops are needed.



- 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.



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

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?

Examples of Collaboration Graphs

- The Hollywood graph models the collaboration of actors in films.
  - We represent actors by vertices and we connect two vertices if the actors they represent have appeared in the same movie.
  - We will study the Hollywood Graph in Section 10.4 when we discuss Kevin Bacon numbers.
- An academic collaboration graph models the collaboration of researchers who have jointly written a paper in a particular subject.
  - We represent researchers in a particular academic discipline using vertices.
  - We connect the vertices representing two researchers in this discipline if they are coauthors of a paper.
  - We will study the academic collaboration graph for mathematicians when we discuss Erdős numbers in Section 10.4.

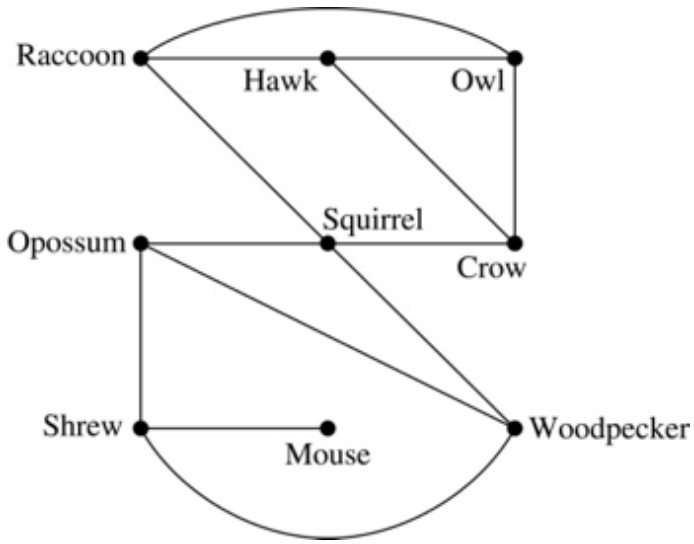
Transportation Graphs

- 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

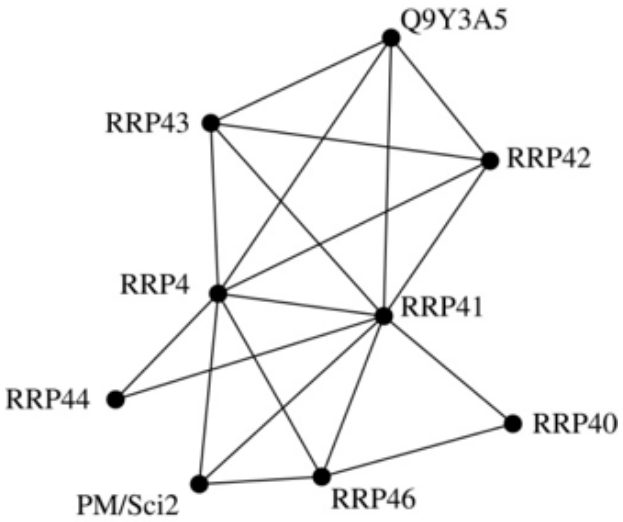
- vertices represent intersections and edges represent roads.
- undirected edges represent two-way roads and directed edges represent one-way roads.

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.



- 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.
- Example: This is a module of the protein interaction graph of proteins that degrade RNA in a human cell.


Q9Y3A5
RRP43
RRP42
RRP4
RRP41
RRP44
RRP40
PM/Sci2
RRP46

10.2 Graph Terminology and Special Types of Graphs

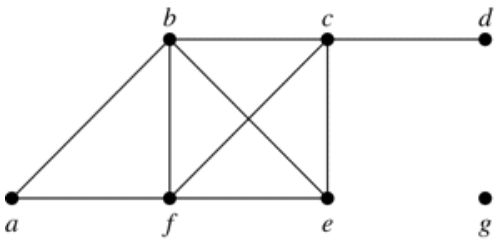
Friday, April 27, 2018 9:27 AM

Basic Terminology

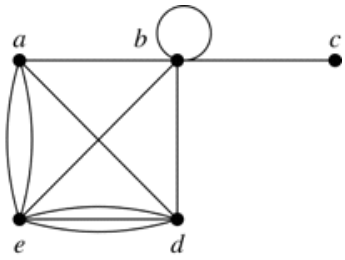
- Definition 1
  - Two vertices u, v in an undirected graph G are called adjacent (or neighbors) in G if
    - there is an edge e between u and v
  - Such an edge e is called incident with the vertices u and v and e is said to connect u and v
- Definition 2
  - The set of all a vertices v of $G = (V, E)$, 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 $\deg(v)$.

Degrees and Neighborhoods of Vertices

- What are the degrees and neighborhoods of the vertices in the graphs G and H?



G



H

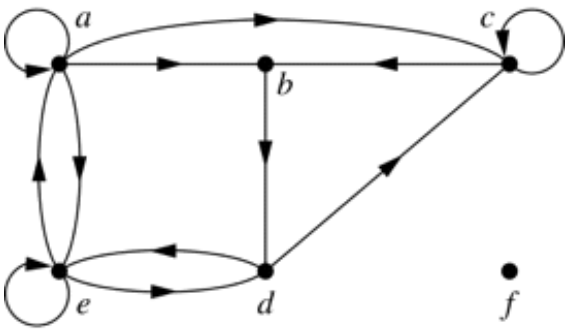
- For graph G
  - $\deg(a) = 2, \deg(b) = \deg(c) = \deg(f) = 4, \deg(d) = 1, \deg(e) = 3, \deg(g) = 0.$
  - $N(a) = \{b, f\}, N(b) = \{a, c, e, f\}, N(c) = \{b, d, e, f\}, N(d) = \{c\},$
  - $N(e) = \{b, c, f\}, N(f) = \{a, b, c, e\}, N(g) = \emptyset.$
- For graph H
  - $\deg(a) = 4, \deg(b) = \deg(e) = 6, \deg(c) = 1, \deg(d) = 5.$
  - $N(a) = \{b, d, e\}, N(b) = \{a, b, c, d, e\}, N(c) = \{b\}, N(d) = \{a, b, e\}, N(e) = \{a, b, d\}.$

Degrees of Vertices

- Theorem 1 (Handshaking Theorem)
  - If $G = (V, E)$ is an undirected graph with m edges, then
  - $2m = \sum_{v \in V} \deg v$
- Proof
  - Each edge contributes twice to the degree count of all vertices.
  - Both the left-hand and right-hand sides of this equation equal twice the number of edges.
  - Think about the graph where
    - vertices represent the people at a party and
    - an edge connects two people who have shaken hands.
- 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.
- Theorem 2
  - An undirected graph has an even number of vertices of odd degree.
- Proof
  - Let $G = (V, E)$ be an undirected graph with m edges
  - Let V_1 be the vertices of even degree and V_2 be the vertices of odd degree in G
  - Then $2m = \sum_{v \in V} \deg v = \sum_{v \in V_1} \deg v + \sum_{v \in V_2} \deg v$, where
    - $\sum_{v \in V_1} \deg v$ must be even since $\deg v$ is even for each $v \in V_1$
    - $\sum_{v \in V_2} \deg v$ 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

Directed Graphs

- 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.
- Definition:
  - The in-degree of a vertex v, denoted $\deg^-(v)$, is the number of edges which terminate at
  - The out-degree of v, denoted $\deg^+(v)$, is the number of edges with v as their initial vertex.
  - Note
    - a loop at a vertex contributes 1 to both the in-degree and the out-degree
- Example

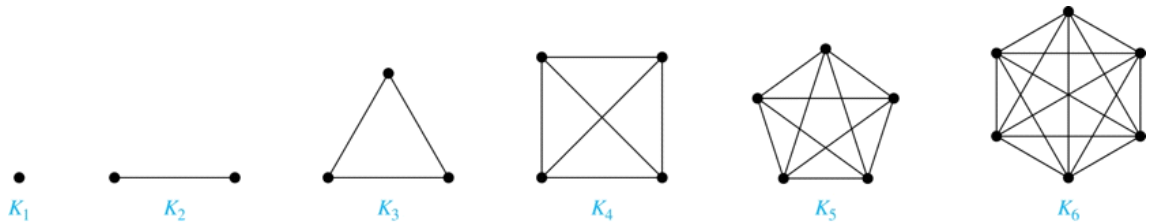


G

  - In the graph G we have
    - $\deg^-(a) = 2, \deg^-(b) = 2, \deg^-(c) = 3, \deg^-(d) = 2, \deg^-(e) = 3, \deg^-(f) = 0.$
    - $\deg^+(a) = 4, \deg^+(b) = 1, \deg^+(c) = 2, \deg^+(d) = 2, \deg^+(e) = 3, \deg^+(f) = 0.$
- Theorem 3
  - Let $G = (V, E)$ be a graph with directed edges
  - Then, $|E| = \sum_{v \in V} \deg^-(v) + \deg^+(v)$
- Proof
  - The first sum counts the number of outgoing edges over all vertices
  - 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.

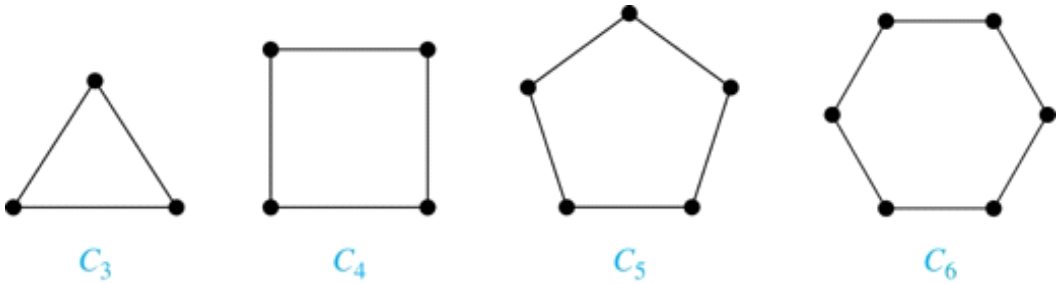
Special Types of Simple Graphs: Complete Graphs

- 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.

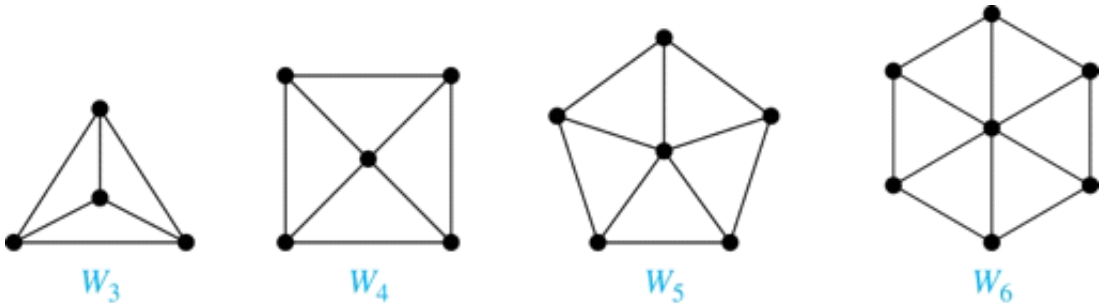


Special Types of Simple Graphs: Cycles and Wheels

- A cycle C_n for $n \geq 3$ consists of
  - n vertices $v_1, v_2, \ldots, v_n$, and
  - edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$

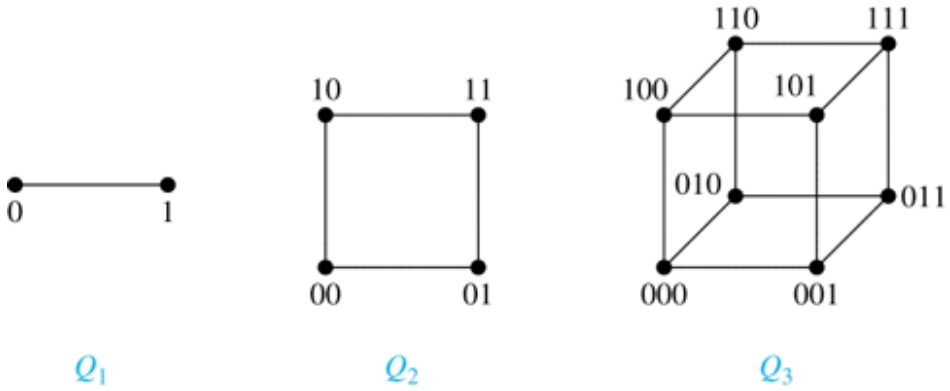


- A wheel W_n is obtained by
  - adding an additional vertex to a cycle C_n for $n \geq 3$, and
  - connecting this new vertex to each of the n vertices in C_n by new edges.



Special Types of Simple Graphs: n-Cubes

- An n-dimensional hypercube, or n-cube, Q_n, is
  - a graph with $2n$ vertices representing all bit strings of length n, where
  - there is an edge between two vertices that differ in exactly one bit position.

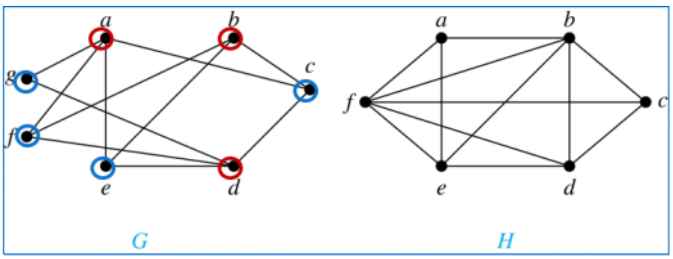


Q_1 Q_2 Q_3

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.

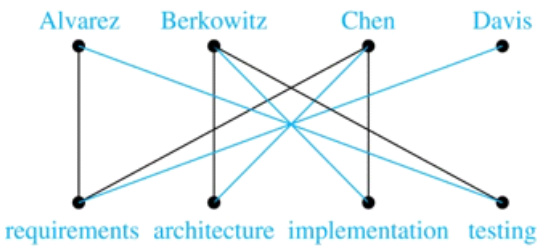
G is bipartite



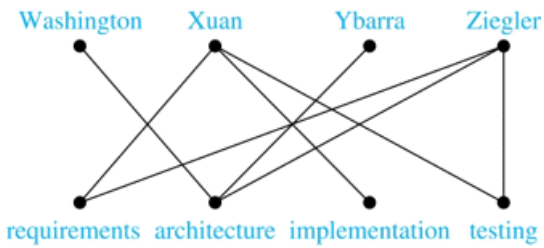
H is not bipartite since if we color a red, then the adjacent vertices f and b must both be blue.

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.



(a)

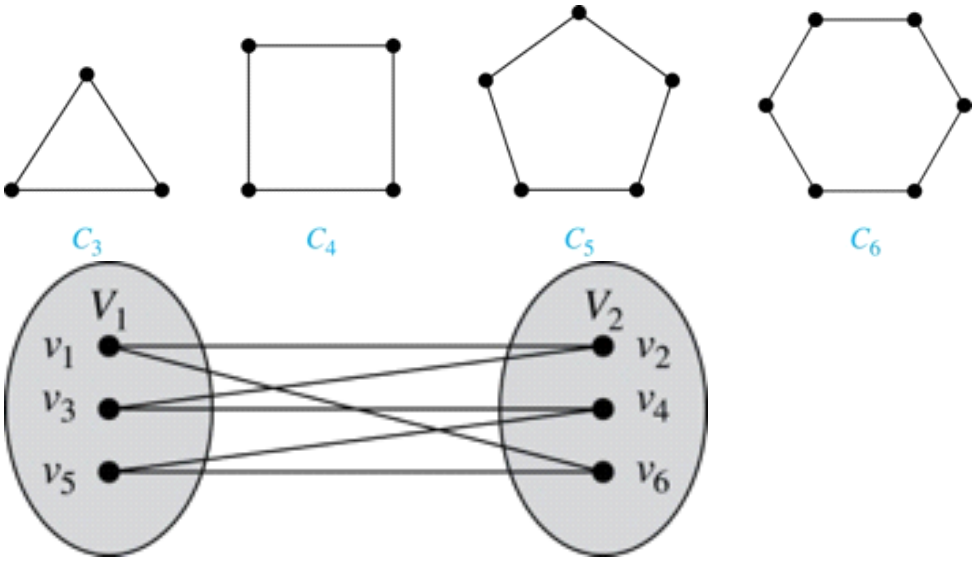


(b)

Examples of Bipartite Graphs

- Example
  - Show that C_6 is bipartite.
- Solution

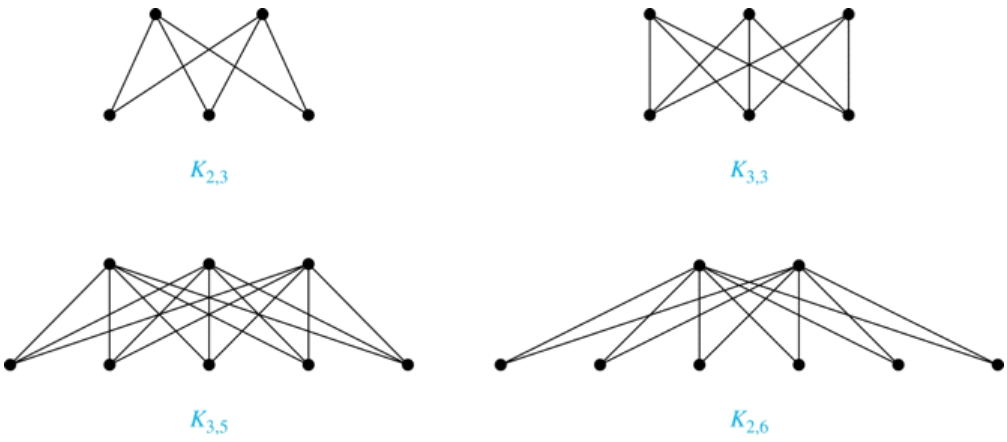
- We can partition the vertex set into $V_1 = \{v_1, v_3, v_5\}$ and $V_2 = \{v_2, v_4, v_6\}$ so that
- every edge of C_6 connects a vertex in V_1 and V_2



- Example
  - Show that C_2 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.

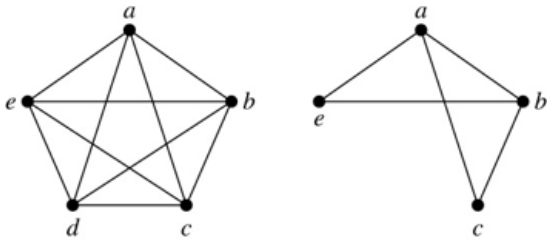
Complete Bipartite Graphs

- 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.

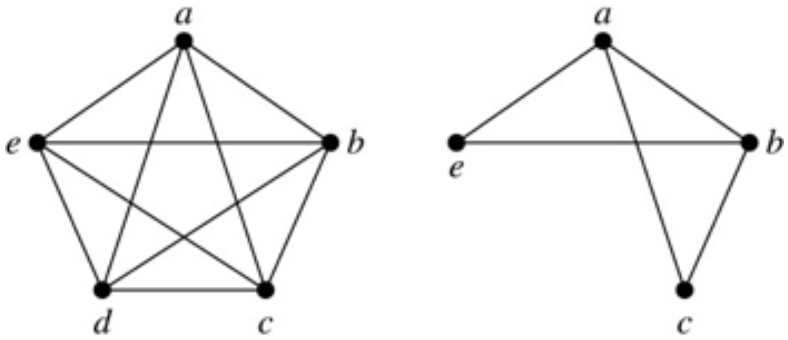


New Graphs from Old

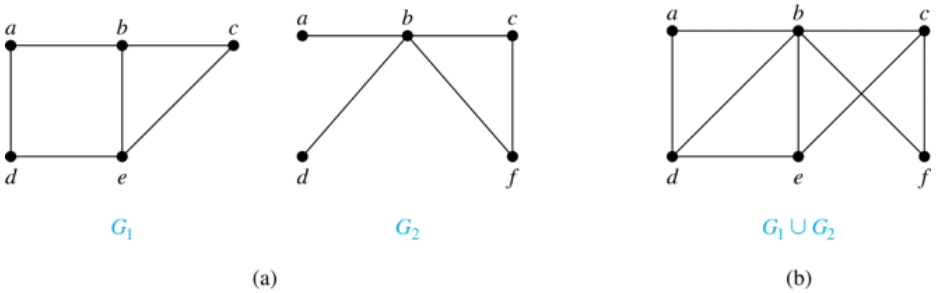
- Definition
  - A subgraph of a graph $G = (V, E)$ is a graph (W, F), where $W \subset V$ and $F \subset E$.
  - A subgraph H of G is a proper subgraph of G if $H \neq G$.
- Example
  - Here we show K_5 and one of its subgraphs.



- 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 $W = \{a, b, c, e\}$.



- 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 $E_1 \cap E_2$
  - The union of G_1 and G_2 is denoted by $G_1 \cup G_2$.
- Example



10.3 Representing Graphs and Graph Isomorphism

May 2, 2018 9:05 AM

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 1

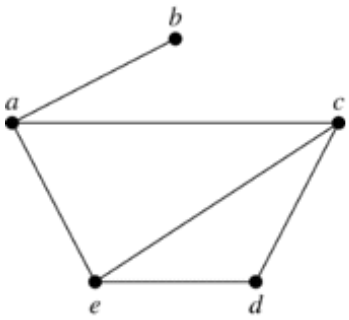


TABLE 1 An Adjacency List for a Simple Graph.

Vertex	*Adjacent Vertices*
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

- Example 2

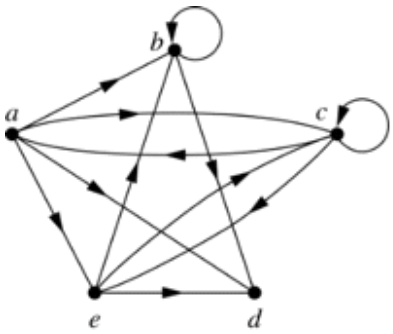


TABLE 2 An Adjacency List for a Directed Graph.

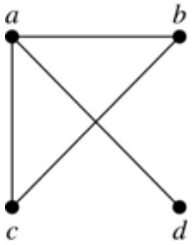
Initial Vertex	*Terminal Vertices*
a	b, c, d, e
b	b, d
c	a, c, e
d	
e	b, c, d

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 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 $A_G = [a_{ij}]$, then

- $a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G \\ 0 & \text{otherwise} \end{cases}$

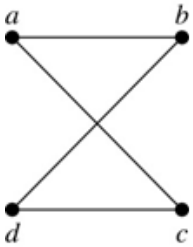
- Example 1



$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

  - The ordering of vertices is a, b, c, d.

- Example 2



$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

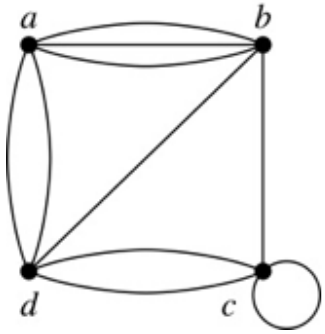
  - The ordering of vertices is a, b, c, d.

- 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_{ii} for $i = 1, 2, 3, \ldots, n$, is 0.

- Adjacency list vs adjacency matrix
  - When a graph is sparse (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
  - But for a dense graph, which includes a high percentage of possible edges
    - An adjacency matrix is preferable

Adjacency Matrices: Graphs with Loops and Multiple Edges

- Adjacency matrices can also be used to represent graphs with loops and multiple edges.
- A loop at the vertex v_i is represented by a 1 at the (i, j)th position of the matrix.
- When multiple edges connect the same pair of vertices v_i and v_j, (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.



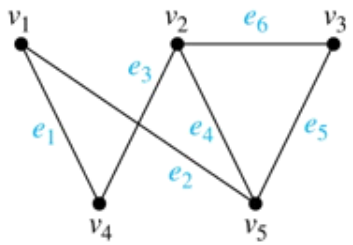
$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}$$

Adjacency Matrices: Directed graphs

- 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 $v_1, v_2, \dots, v_n$ is a list of the vertices.
- In other words, if the graphs adjacency matrix is $A_G = [a_{ij}]$, 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.

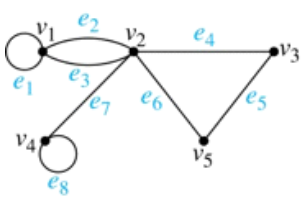
Representation of Graphs: Incidence Matrices

- Definition
  - Let $G = (V, E)$ be an undirected graph with
    - vertices $v_1, v_2, \dots, v_n$ and
    - edges $e_1, e_2, \dots, e_m$.
  - The incidence matrix with respect to the ordering of V and E is the $n \times m$ matrix
    - $M = [m_{ij}]$, where $m_{ij} = \begin{cases} 1 & \text{when edge } e_j \text{ is incident with } v_i \\ 0 & \text{otherwise} \end{cases}$
- 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
  - The columns going from left to right represent e_1 through e_6.
- Example



$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

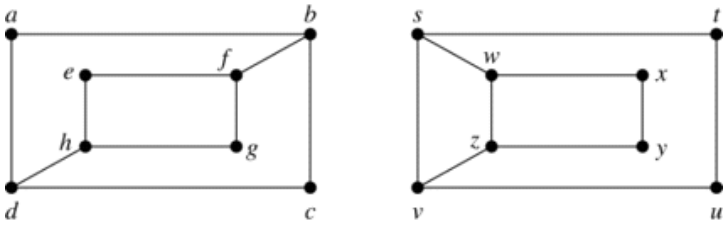
- The rows going from top to bottom represent v_1 through v_5
- The columns going from left to right represent e_1 through e_8.

Isomorphism of Graphs

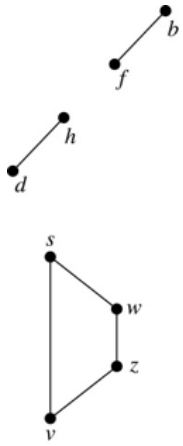
- Definition
  - The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if
    - there is a one-to-one and onto function f from V_1 to V_2 with the property that
    - a and b are adjacent in $G_1 \Leftrightarrow f(a)$ and $f(b)$ are adjacent in G_2, for all a and b in V_1.
  - Such a function f is called an isomorphism.
  - Two simple graphs that are not isomorphic are called nonisomorphic.
- Example
  - Show that the graphs $G = (V, E)$ and $H = (W, F)$ are isomorphic.
- Solution
  - The function f defined below is a one-to-one correspondence between V and W
    - $f(u_1) = v_1$
    - $f(u_2) = v_4$
    - $f(u_3) = v_3$
    - $f(u_4) = v_2$
  - 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 below consists of two adjacent vertices in H
    - $f(u_1) = v_1, f(u_2) = v_4$
    - $f(u_1) = v_1, f(u_3) = v_3$
    - $f(u_2) = v_4, f(u_4) = v_2$
    - $f(u_3) = v_3, f(u_4) = v_2$
- Note
  - 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 whether 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.

- Example
  - Determine whether these two graphs are isomorphic

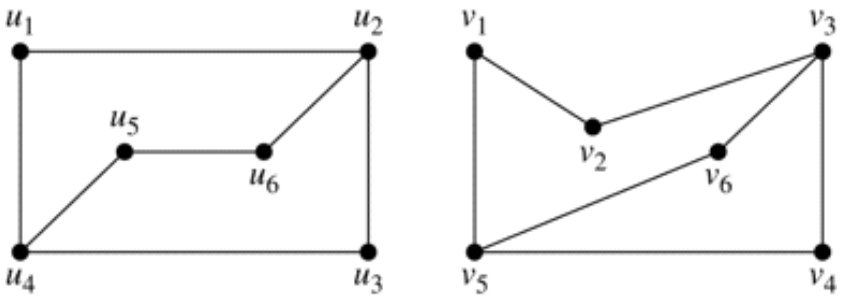


G H



- 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.

- Example
  - Determine whether these two graphs are isomorphic.



G H

- 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.

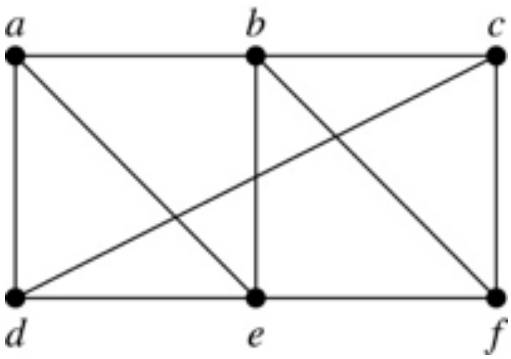
- We define an injection f from the vertices of G to the vertices of H that preserves the degree of vertices.
- We will determine whether it is an isomorphism.
- The function f defined below is a one-to-one correspondence between G and H
  - $f(u_1) = v_6, f(u_2) = v_3, f(u_3) = v_4, f(u_4) = v_5, f(u_5) = v_1, f(u_6) = v_2$
- 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.
- See the text for an illustration of how adjacency matrices can be used for this verification.

10.4 Connectivity

May 2, 2018 9:07 AM

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
    - determining whether a message can be sent between two computers.
    - efficiently planning routes for mail delivery
- 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 $e_1, \dots, e_n$ of G for which
    - there exists a sequence $x_0 = u, x_1, \dots, x_{n-1}, x_n = v$ of vertices such that
    - e_i has, for $i = 1, \dots, n$, the endpoints x_{i-1} and x_i.
  - When the graph is simple, we denote this path by its vertex sequence
    - $x_0, x_1, \dots, x_n$(since listing the vertices uniquely determines the path).
  - The path is a circuit if
    - it begins and ends at the same vertex and has length greater than zero.
  - The path or circuit is said to
    - pass through the vertices $x_1, x_2, \dots, x_{n-1}$ and
    - traverse the edges $e_1, \dots, e_n$
  - A path or circuit is simple if it does not contain the same edge more than once.
- Example

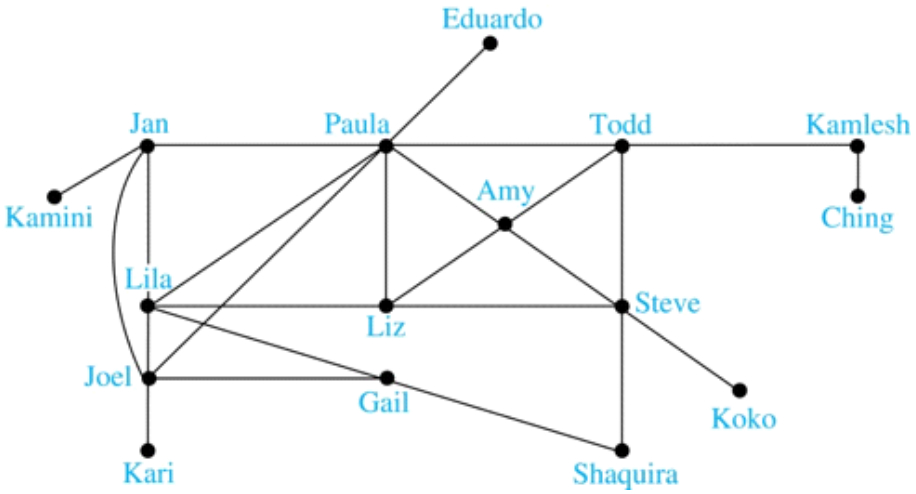


  - In the simple graph here:

- 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.

Degrees of Separation

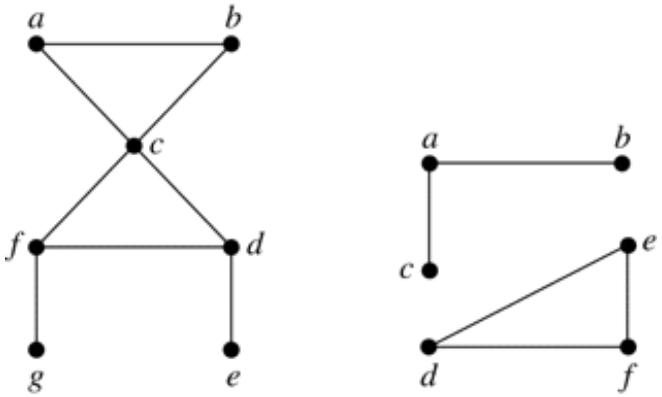
- Paths in Acquaintanceship Graphs
  - In an acquaintanceship graph there is a path between two people if there is a chain of people linking these people, where two people adjacent in the chain know one another.
  - In this graph there is a chain of six people linking Kamini and Ching.



- Note:
  - Some have speculated that almost every pair of people in the world are linked by a small chain of no more than six, or maybe even, five people.
  - The play Six Degrees of Separation by John Guare is based on this notion.

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.

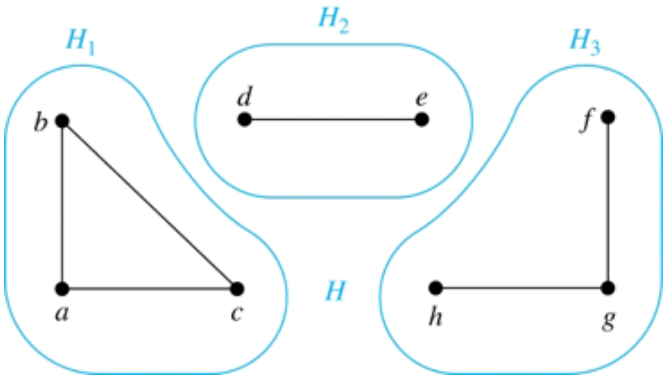


G_1 G_2

- Theorem
  - Every pair of distinct vertices in a connect graph is connected by a simple path
- Proof
  - Let u and v be two distinct vertices of the connected undirected graph $G = (V, E)$
  - Because G is connected, there is at least one path between u and v
  - Let $x_0, x_1, \dots, x_n$, where $x_0 = u$ and $x_n = v$, be the vertex sequence of a path of least length.
  - This path of least length is simple.
  - To see this, suppose is not simple, then $x_i = x_j$ for some i and j with $0 \leq i < j$
  - This means that there is a path from u to v of shorter length with vertex sequence
    - $x_0, x_1, \dots, x_{i-1}, x_j, \dots, x_n$
  - Obtained by deleting the edges corresponding to the vertex sequence $x_i, \dots, x_{j-1}$

Connected Components

- Definition
  - A connected component of a graph G is
    - a connected subgraph of G that is
    - not a proper subgraph of another connected subgraph of G
  - A graph G that is not connected has two or more connected components that are disjoint and have G as their union.
- Example
  - The graph H is the union of three disjoint subgraphs H_1, H_2, and H_3
  - None of which are proper subgraphs of a larger connected subgraph of G
  - These three subgraphs are the connected components of H.



Connectedness in Directed Graphs

- Definition
  - A directed graph is strongly connected if
  - there is a path from a to b and a path from b to a whenever a and b are vertices in the graph
- Definition
  - A directed graph is weakly connected if
  - there is a path between every two vertices in the underlying undirected graph, which is
  - the undirected graph obtained by ignoring the directions of the edges of the directed graph

Counting Paths between Vertices

- We can use the adjacency matrix of a graph to find the number of paths between two vertices in the graph.
- Theorem
  - Let G be a graph with adjacency matrix A with respect to the ordering $v_1, \ldots, v_n$ of vertices
  - (with directed or undirected edges, multiple edges and loops allowed).
  - The number of different paths of length r from v_i to v_j, where $r > 0$ is a positive integer,
  - equals the (i, j)th entry of A^r.
- Proof by mathematical induction:
  - Basis Step
    - By definition of the adjacency matrix
    - The number of paths from v_i to v_j of length 1 is the (i, j)th entry of A.
  - Inductive Step
    - For the inductive hypothesis, we assume that
      - the (i, j)th entry of A^r is the number of different paths of length r from v_i to v_j.
    - Because $A^{r+1} = A^r A$, the (i, j)th entry of A^{r+1} equals
      - $b_{i1}a_{1j} + b_{i2}a_{2j} + \cdots + b_{in}a_{nj}$, where b_{ik} is the (i, k)th entry of A^r.
    - By the inductive hypothesis, b_{ik} is the number of paths of length r from v_i to v_k.
    - A path of length $r + 1$ from v_i to v_j is made up of
      - a path of length r from v_i to some v_k, and
      - an edge from v_k to v_j.

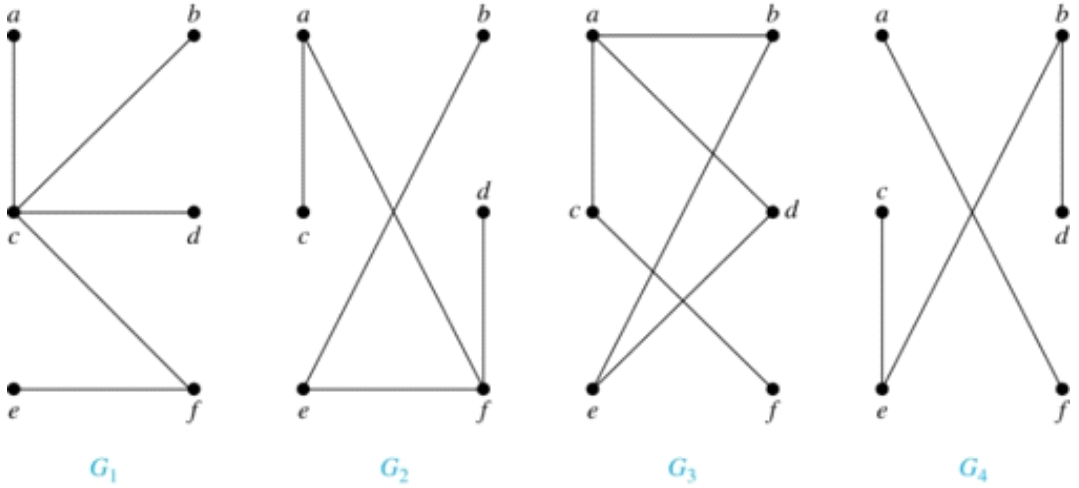
- By the product rule for counting, the number of such paths is the product of
  - the number of paths of length r from v_i to v_k (i.e., b_{ik}) and
  - the number of edges from v_k to v_j (i.e, akj).
- The sum over all possible intermediate vertices v_k is $b_{i1}a_{1j} + b_{i2}a_{2j} + \cdots + b_{in}a_{nj}$

11.1 Introduction to Trees

May 4, 2018 9:01 AM

Trees

- Definition
  - A tree is a connected undirected graph with no simple circuits.
- Example
  - Which of these graphs are trees?

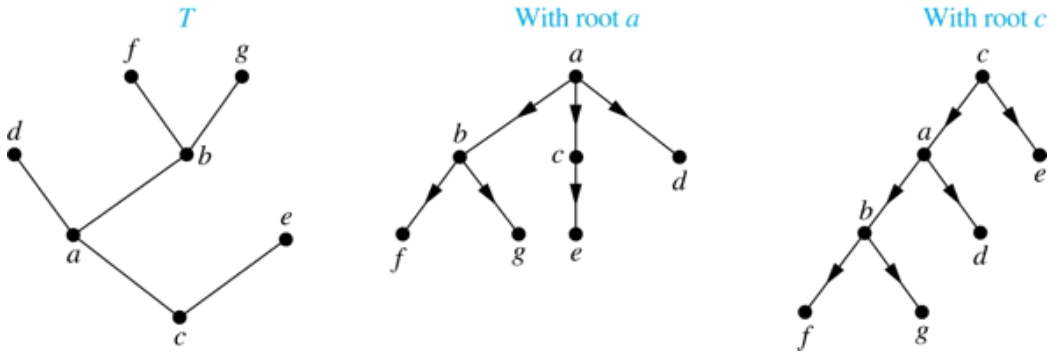


- Solution
  - G_1 and G_2 are trees - both are connected and have no simple circuits
  - Because e, b, a, d, e is a simple circuit, G_3 is not a tree.
  - G_4 is not a tree because it is not connected.
- Definition
  - A forest is a graph that has no simple circuit, but is not connected.
  - Each of the connected components in a forest is a tree.
- Theorem
  - An undirected graph is a tree if and only if
  - There is a unique simple path between any two of its vertices.
- Proof
  - ($\Longrightarrow$) Assume that T is a tree.
    - Then T is connected with no simple circuits.
    - Hence, if x and y are distinct vertices of T
    - There is a simple path between them (by Theorem 1 of Section 10.4).
    - This path must be unique
      - If there were a second path
      - There would be a simple circuit in T (by Exercise 59 of Section 10.4).

        - Hence, there is a unique simple path between any two vertices of a tree.
    - (⟸) Assume that there is a unique simple path between any two vertices of graph T
        - T is connected because there is a path between any two of its vertices.
        - Furthermore, T can have no simple circuits since
            - If there were a simple circuit
            - There would be two paths between some two vertices
        - Therefore T is a tree
    - Hence, a graph with a unique simple path between any two vertices is a tree.

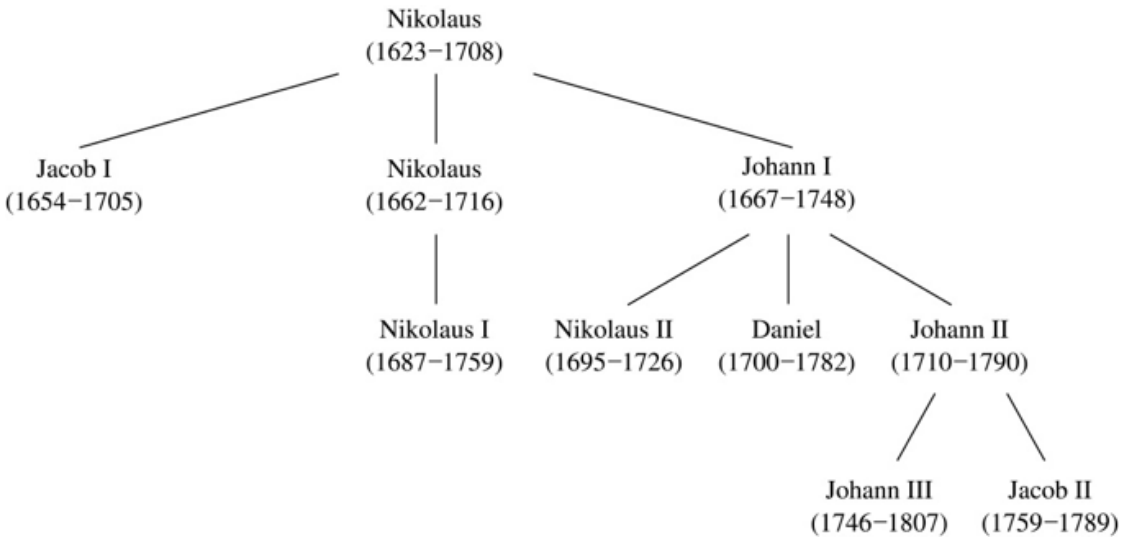
Rooted Trees

- 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



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)

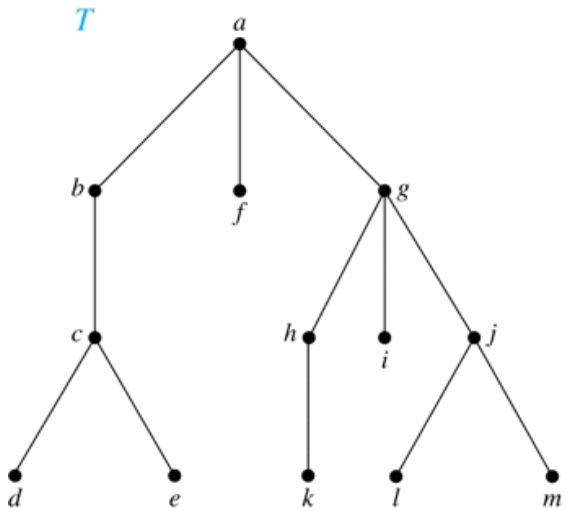


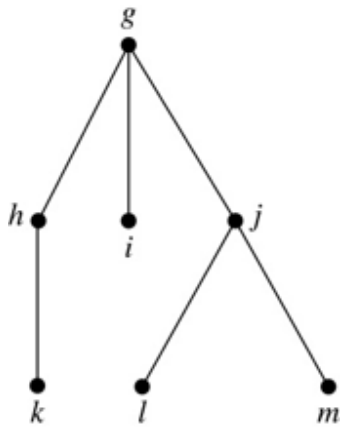
- If v is a vertex of a rooted tree other than the root, the parent of v is
    - the unique vertex u such that there is a directed edge from u to v

- When u is a parent of v, v is called a child of u.
- Vertices with the same parent are called siblings.
- 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.
- A vertex of a rooted tree with no children is called a leaf.
- Vertices that have children are called internal vertices.
- If a is a vertex in a tree, the subtree with a as its root is the subgraph of the tree
  - consisting of a, and
  - its descendants, and
  - all edges incident to these descendants

Examples of Rooted Trees

- Example: In the rooted tree T (with root a):
  - Find the parent of c, the children of g, the siblings of h, the ancestors of e, and the descendants of b.
  - Find all internal vertices and all leaves.
  - What is the subtree rooted at G?





- Solution:
  - 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
  - The internal vertices are a, b, c, g, h, and j
  - The leaves are d, e, f, i, k, l, and m

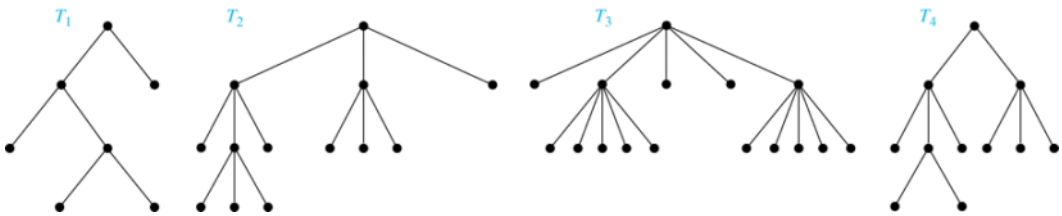
- We display the subtree rooted at g

Properties of Trees

- Theorem 2
  - A tree with n vertices has $n - 1$ edges.
- Proof (by mathematical induction):
  - Basis Step
    - When $n = 1$, a tree with one vertex has no edges.
    - Hence, the theorem holds when $n = 1$.
  - Inductive Step
    - Assume that every tree with k vertices has $k - 1$ edges.
    - Suppose that a tree T has $k + 1$ vertices and that v is a leaf of T
    - Let w be the parent of v
    - Removing the vertex v and the edge connecting w to v
    - This produces a tree T' with k vertices.
    - By the inductive hypothesis, T' has $k - 1$ edges.
    - Because T has one more edge than T', we see that T has k edges.
    - This completes the inductive step

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?



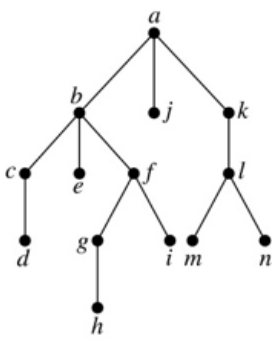
- Solution
  - T_1 is a full binary tree because each of its internal vertices has two children.
  - T_2 is a full 3-ary tree because each of its internal vertices has three children.
  - In T_3 each internal vertex has five children, so T_3 is a full 5-ary tree
  - T_4 is not a full m-ary tree for any m because
    - some of its internal vertices have two children and others have three children

Counting Vertices in Full m-Ary Trees

- Theorem 3
  - A full m-ary tree with i internal vertices has $n = mi + 1$ vertices.
- Proof
  - Every vertex, except the root, is the child of an internal vertex.
  - Because each of the i internal vertices has m children,
  - there are mi vertices in the tree other than the root.
  - Hence, the tree contains $n = mi + 1$ vertices.
- Theorem 4
  - A full m-ary tree with
    - n vertices has $i = \frac{n-1}{m}$ internal vertices and $l = \frac{(m-1)n+1}{m}$ leaves
    - i internal vertices has $n = mi + 1$ vertices and $l = (m-1)i + 1$ leaves
    - l leaves has $n = \frac{ml-1}{m-1}$ vertices and $i = \frac{l-1}{m-1}$ internal vertices
  - Proof (vertices)
    - Solving for i in $n = mi + 1$ (from Theorem 3) gives $i = \frac{n-1}{m}$.
    - Since each vertex is either a leaf or an internal vertex, $n = l + i$.
    - By solving for l and using the formula for i, we see that
    - $l = n - i = n - \frac{n-1}{m} = \frac{(m-1)n+1}{m}$

Level of vertices and height of trees

- When working with trees, we often want to have rooted trees where
  - the subtrees 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.
- Example:
  - Find the level of each vertex in the tree to the right.
  - What is the height of the tree?

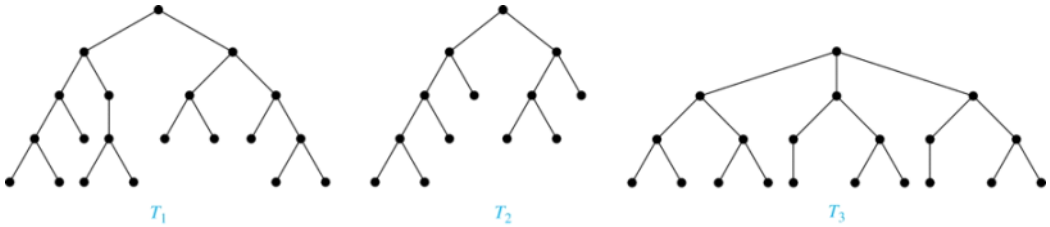


- Solution:

- 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.
- The height is 4, since 4 is the largest level of any vertex.

Balanced m-Ary Trees

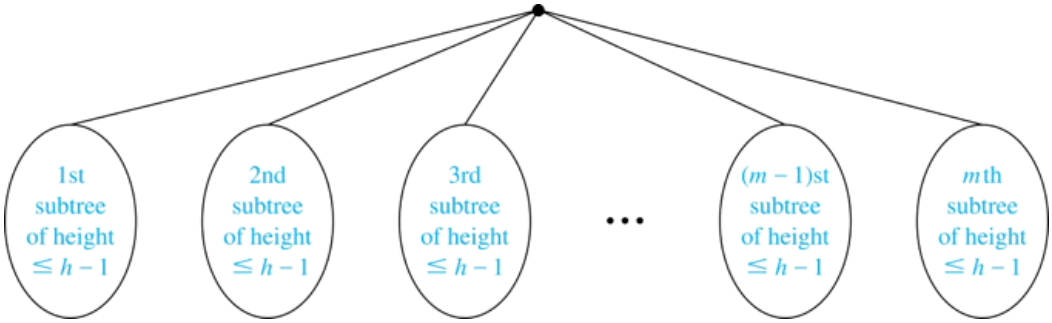
- Definition
  - A rooted m-ary tree of height h is balanced if all leaves are at levels h or $h-1$.
- Example
  - Which of the rooted trees shown below is balanced?



- Solution
  - T_1 and T_3 are balanced, but T_2 is not because it has leaves at levels 2, 3, and 4.

The Bound for the Number of Leaves in an m-Ary Tree

- Theorem 5
  - There are at most m^h leaves in an m-ary tree of height h.
- Proof (by mathematical induction on height):
  - Basis Step
    - Consider an m-ary trees of height 1.
    - The tree consists of a root and no more than m children, all leaves.
    - Hence, there are no more than $m^1 = m$ leaves in an m-ary tree of height 1.
  - Inductive Step
    - Assume the result is true for all m-ary trees of height $< h$.
    - Let T be an m-ary tree of height h.
    - The leaves of T are the leaves of the subtrees of T we get when we delete the edges from the root to each of the vertices of level 1.



        - Each of these subtrees has height $\leq h - 1$.
        - By the inductive hypothesis, each of these subtrees has at most m^{h-1} leaves.
        - Since there are at most m such subtrees
        - There are at most $m \cdot m^{h-1} = m^h$ leaves in the tree.
- Corollary 1
    - If an m-ary tree of height h has l leaves, then $h \geq \lceil \log_m l \rceil$.
    - If the m-ary tree is full and balanced, then $h = \lceil \log_m l \rceil$. (see text for the proof)