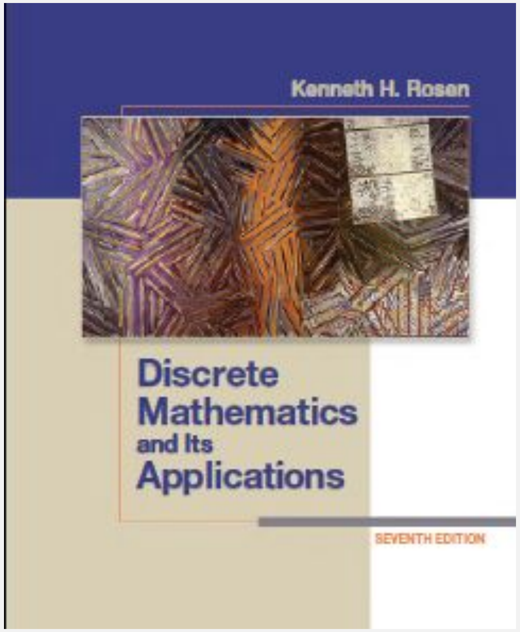
CSE 105: Discrete Mathematics



Course Evaluation

Topic	Marks
Attendance	05
Individual Presentations	05
Group Assignments	05
Class Tests (3)	15
Mid term	30
Final	40

Course Materials

Text Book:

- "*Discrete Mathematics and Its Application*", Kenneth H. Rosen, 7th Edition, McGraw-Hill.
- Lecture notes

Reference Materials:

- "*Schaum's Outlines Discrete Mathematics*", Seymore Lipschutz & Marc Lipson, 3rd Edition, McGraw-Hill.
- http://en.wikiversity.org/wiki/Introductory_Discrete_Mathematics_for_Computer_Science

Course Contents

No	Topic	Exams/Quiz
Lectures 1-2	Introduction to Discrete Mathematics+ Propositional Logic	
Lectures 3-4	Propositional Logic and Introduction to Set	
Lectures 5-6	Quiz-1 + Set	**Quiz-1**
Lectures 7-8	Function	
Lectures 9-10	Quiz-2 + Algorithm	**Quiz-2**
Lectures 11-12	**Midterm Exam**	
Lectures 13-14	Induction + Discrete Probability	GA-1
Lectures 15-16	Counting	Presentations
Lectures 17-18	Quiz-3+Relation	Quiz-3
Lectures 19-20	Number Theory	GA-2
Lectures 21-22	Quiz-4+Graph-Tree	Quiz-4
Lectures 23-24	Graph-Tree	Presentations
	Final Exam	

Discrete mathematics

Discrete mathematics
– study of mathematical structures and objects that are fundamentally **discrete rather than continuous.**

- **Examples of objects with discrete values are – integers, graphs, or statements in logic.**

- Discrete mathematics and **computer science.**

– Concepts from discrete mathematics are useful for describing **objects and problems in computer algorithms and programming languages. These** have applications in cryptography, automated theorem proving, and software development.

Logic

- Logic defines a formal language for representing knowledge and for making logical inference
- It helps us to understand how to construct a valid argument

Logic defines:

- Syntax of statements
- The meaning of statements
- The rules of logical inference (manipulation)

Propositional logic

The simplest logic

- **Definition:**

– **A proposition is a statement that is either true or false.**

- **Examples:**

– Airport is located in the North Part of Dhaka City.

- **(T)**

– 5 + 2 = 8.

- **(F)**

– It is raining today.

- **(either T or F)**

Propositional logic

Examples (cont.):

– How are you?

- **a question is not a proposition**

– $x + 5 = 3$

- **since x is not specified, neither true nor false**

– 2 is a prime number.

- **(T)**

– She is very talented.

- **since she is not specified, neither true nor false**

– There are other life forms on other planets in the universe.

- **either T or F**

Composite Statements

- More complex propositional statements can be built from elementary statements using **logical connectives.**

Example:

- Proposition A: It rains outside
- Proposition B: We will see a movie

- A new (combined) proposition:

If it rains outside then we will see a movie

Composite Statements

- More complex propositional statements can be built from elementary statements using **logical connectives.**

- **Logical connectives:**
  - Negation
  - Conjunction
  - Disjunction
  - Exclusive or
  - Implication
  - Biconditional

Logical connectives: Negation

Definition: Let p be a proposition. The statement "It is not the case that p." is another proposition, called the **negation of p. The** negation of p is denoted by $\neg$ p and read as "not p."

Example:

• Airport is located in the North Part of Dhaka City.

➡

• It is not the case that Airport is located in the North Part of Dhaka City.

Other examples:

– $5 + 2 \neq 8$.

– 10 is not a prime number.

– It is not the case that buses stop running at 9:00pm.

Logical connectives: Negation

Negate the following propositions:

– It is raining today.
• It is not raining today.

– 2 is a prime number.
• 2 is not a prime number

– There are other life forms on other planets in the universe.
• It is not the case that there are other life forms on other planets in the universe.

Logical connectives: Negation

A **truth table** **displays the** **relationships between truth values** **(T or F) of different propositions.**

P	**¬ p**
T	F
F	T

Rows: all possible values of elementary proposition

Logical connectives: Conjunction

Definition: Let p and q be propositions. The proposition "p and q" denoted by p ∧ q, is true when both p and q are true and is false otherwise. The proposition p ∧ q is called the **conjunction of p and q.**

- **Examples:**
  - It is raining today and 2 is a prime number.
  - 2 is a prime number and $5 + 2 \neq 8$.
  - 13 is a perfect square and 9 is prime.

Logical connectives: Disjunction

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

- **Examples:**

– It is raining today or 2 is a prime number.
– 2 is a prime number or $5 + 2 \neq 8$.
– 13 is a perfect square or 9 is a prime.

Truth Tables

- **Conjunction and disjunction**
  - Four different combinations of values for p and q

p	**q**	**$p \wedge q$**	**$p \vee q$**
F	F		
F	T		
T	F		
T	T		

Rows: all possible combinations of values for elementary propositions: 2^n values

Truth Tables

- **Conjunction and disjunction**
  - Four different combinations of values for p and q

p	**q**	**$p \wedge q$**	**$p \vee q$**
F	F	F	
F	T	F	
T	F	F	
T	T	T	

Truth Tables

- **Conjunction and disjunction**
  - Four different combinations of values for p and q

p	**q**	**$p \wedge q$**	**$p \vee q$**
F	F	F	F
F	T	F	T
T	F	F	T
T	T	T	T

NB: $p \vee q$ (the or is used inclusively, i.e., $p \vee q$ is true when either p or q or both are true).

Exclusive or

Definition: Let p and q be propositions. The proposition "p exclusive or q" denoted by p q, is true when exactly one of p and q is true and it is false otherwise.

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

Implications

Definition: Let p and q be propositions. The proposition "p implies q" denoted by $p \to q$ is called implication. It is false when p is true and q is false and is true otherwise.

In $p \to q$, p is called the **hypothesis** and q is called the **conclusion**.

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

Implications

$p \rightarrow q$ is read in a variety of equivalent ways:

- if p then q
- p only if q
- p is sufficient for q
- q whenever p

Examples:

– if Germany won 2010 world cup then 2 is a prime.

- If F then T ? **T**

− if today is Monday then 2 * 3 = 8.

- $T \rightarrow F$? **F**

Implications

- The **converse of $p \to q$ is $q \to p$.**
- The **contrapositive of $p \to q$ is $\neg q \to \neg p$**
- The **inverse of $p \to q$ is $\neg p \to \neg q$**

Examples:

- If it jams, the traffic moves slowly.
- p: it jams
- q: traffic moves slowly.
- $\mathbf{p \to q}$

Implications

The converse:

If the traffic moves slowly then it jams.

- **$q \rightarrow p$**

The contrapositive:

- If the traffic does not move slowly then it does not jams.
- $\neg q \rightarrow \neg p$

The inverse:

- If it does not jams the traffic moves quickly.
- $\neg p \rightarrow \neg q$

Biconditional

Definition: Let p and q be propositions. The biconditional p $\leftrightarrow$ q (read p if and only if q), is true when p and q have the same truth values and is false otherwise.

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

Note: two truth values always agree.

Constructing the truth table

Example: Construct a truth table for $(p \rightarrow q) \wedge (\neg p \leftrightarrow q)$

- Simpler if we decompose the sentence to elementary and intermediate propositions

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

Constructing the truth table

Example: Construct a truth table for $(p \rightarrow q) \wedge (\neg p \leftrightarrow q)$

- Simpler if we decompose the sentence to elementary and intermediate propositions

p	**q**	$\neg p$	$p \rightarrow q$	$\neg p \leftrightarrow q$	$(p \rightarrow q) \wedge (\neg p \leftrightarrow q)$
F	F	T			
F	T	T			
T	F	F			
T	T	F			

Constructing the truth table

Example: Construct a truth table for $(p \rightarrow q) \wedge (\neg p \leftrightarrow q)$

- Simpler if we decompose the sentence to elementary and intermediate propositions

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

Constructing the truth table

Example: Construct a truth table for $(p \rightarrow q) \wedge (\neg p \leftrightarrow q)$

- Simpler if we decompose the sentence to elementary and intermediate propositions

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

Constructing the truth table

Example: Construct a truth table for $(p \rightarrow q) \wedge (\neg p \leftrightarrow q)$

- Simpler if we decompose the sentence to elementary and intermediate propositions

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

Computer Representation of True and False

We need to encode two values True and False:

- Computers represents data and programs using 0s and 1s
- Logical truth values – True and False

- A bit is sufficient to represent two possible values:

– 0 (False) or 1(True)

- A variable that takes on values 0 or 1 is called a Boolean variable.

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

Bitwise operation

- T and F replaced with 1 and 0

p	**q**	**p ∧ q**	**p ∨ q**
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

p	**¬p**
0	1
1	0

Bitwise operation

- T and F replaced with 1 and 0

$$\begin{array}{cc} & 1\ 0\ 1\ 1\ 0\ 0\ 1\ 1 \\ \vee & 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1 \\ \hline & \boxed{1\ 1\ 1\ 1\ 1\ 0\ 1\ 1} \end{array} \qquad \begin{array}{cc} & 1\ 0\ 1\ 1\ 0\ 0\ 1\ 1 \\ \wedge & 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1 \\ \hline & \boxed{0\ 0\ 0\ 0\ 0\ 0\ 0\ 1} \end{array} \qquad \begin{array}{cc} & 1\ 0\ 1\ 1\ 0\ 0\ 1\ 1 \\ \oplus & 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1 \\ \hline & \boxed{1\ 1\ 1\ 1\ 1\ 0\ 1\ 0} \end{array}$$

Applications of propositional logic

- **Translation of English sentences**

- **Inference and reasoning:**

  – new true propositions are inferred from existing ones

  – Used in Artificial Intelligence:

  - Builds programs that act intelligently
  - Programs often rely on symbolic manipulations

- **Design of logic circuit**

Translation

Assume a sentence:
If you are older than 13 or you are with your parents then you can attend a PG-13 movie.

Parse:

- If (you are older than 13 or you are with your parents) then (you can attend a PG-13 movie)

Atomic (elementary) propositions:

– A= you are older than 13
– B= you are with your parents
– C=you can attend a PG-13 movie

- Translation: $A \vee B \rightarrow C$

Application of Inference

Assume we know the following sentences are true:
If you are older than 13 or you are with your parents then you can attend a PG-13 movie. You are older than 13.

Translation:

- If (you are older than 13 or you are with your parents) then (you can attend a PG-13 movie) . (You are older than 13).
  – A= you are older than 13
  – B= you are with your parents
  – C=you can attend a PG-13 movie
- $(A \vee B \rightarrow C)$, A
- $(A \vee B \rightarrow C) \wedge A$ is true
- With the help of the logic we can infer the following statement (proposition):
  – You can attend a PG-13 movie or C is True

Tautology and Contradiction

Some propositions are interesting since their values in the truth table are always the same

Definitions:

- A compound proposition that is always true for all possible truth values of the propositions is called a **tautology.**
- A compound proposition that is always false is called a **contradiction.**
- A proposition that is neither a tautology nor contradiction is called a **contingency.**

Example: $p \lor \neg p$ is a **tautology**.
$p \land \neg p$ is a **contradiction**.

p	$\neg p$	$p \lor \neg p$	$p \land \neg p$
1	0	1	0
0	1	1	0

Equivalence

How do we determine that two propositions are equivalent?

Their truth values in the truth table are the same.

- Example: **$p \rightarrow q$ is equivalent to $\neg q \rightarrow \neg p$ (contrapositive)**

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

Equivalent statements are important for logical reasoning since they can be substituted and can help us to make a logical argument.

Logical Equivalence

Definition: The propositions p and q are called logically equivalent if p ↔ q is a tautology (alternately, if they have the same truth table). The notation **p <=> q denotes p and q are** logically equivalent.

p	**q**	$p \rightarrow q$	$\neg q \rightarrow \neg p$	$(p \rightarrow q) <=> (\neg q \rightarrow \neg p)$
F	F	T	T	T
F	T	T	T	T
T	F	F	F	T
T	T	T	T	T

Important Logical Equivalence

- **Double negation**
  - $\neg(\neg p) \Leftrightarrow p$

- **Commutative**
  - $p \vee q \Leftrightarrow q \vee p$
  - $p \wedge q \Leftrightarrow q \wedge p$

- **Associative**
  - $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
  - $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$

Important Logical Equivalence

- **Distributive**
  - $p \lor (q \land r) \Leftrightarrow (p \lor q) \land (p \lor r)$
  - $p \land (q \lor r) \Leftrightarrow (p \land q) \lor (p \land r)$

- **De Morgan**
  - $\neg(p \lor q) \Leftrightarrow \neg p \land \neg q$
  - $\neg(p \land q) \Leftrightarrow \neg p \lor \neg q$

- **Other useful equivalences**
  - $p \lor \neg p \Leftrightarrow T$
  - $p \land \neg p \Leftrightarrow F$
  - $\mathbf{p \rightarrow q \Leftrightarrow (\neg p \lor q)}$

Showing Logical Equivalence

Show $(p \land q) \rightarrow p$ is a tautology

In other words $((p \land q) \rightarrow p \Leftrightarrow T)$

$(p \land q) \rightarrow p$	$\Leftrightarrow \neg(p \land q) \lor p$	Useful
	$\Leftrightarrow [\neg p \lor \neg q] \lor p$	DeMorgan
	$\Leftrightarrow [\neg q \lor \neg p] \lor p$	Commutative
	$\Leftrightarrow \neg q \lor [\neg p \lor p]$	Associative
	$\Leftrightarrow \neg q \lor [T]$	Useful
	$\Leftrightarrow T$	Domination

You can also use truth table to show this

Thank You