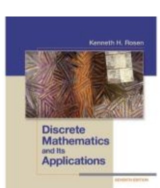
CSE2023 Discrete Computational Structures

Course goals

- Mathematical reasoning
  - Logic, inference, proof
- Combinatorial analysis
  - Count and enumerate objects
- Discrete structures
  - Sets, sequences, functions, graphs, trees, relations
- Algorithmic reasoning
  - Specifications and verifications
- Applications and modeling
  - Internet, business, artificial intelligence, etc.

Textbook

- *Discrete Mathematics and Its Applications*
  by Kenneth H. Rosen, 7th edition, McGraw Hill



Prerequisite

- Basic knowledge of calculus
- Basic knowledge in computer science

Grading

- Midterm %30
- Quiz(es) %15
- Homeworks %15
- Final 40%

1.1 Propositional logic

- Understand and construct correct mathematical arguments
- Give precise meaning to mathematical statements
- Rules are used to distinguish between valid (true) and invalid arguments
- Used in numerous applications: circuit design, programs, verification of correctness of programs, artificial intelligence, etc.

Proposition

- A declarative sentence that is either true or false, but not both
  - Washington, D.C., is the capital of USA
  - California is adjacent to New York
  - 1+1=2
  - 2+2=5
  - What time is it?
  - Read this carefully

Logical operators

- Negation operator
- Conjunction (and, ^)
- Disjunction (or v)
- Conditional statement →
- Biconditional statement ←→
- Exclusive Or

Negation

TABLE 1 The Truth Table for the Negation of a Proposition.

p	$\neg p$
T	F
F	T

Example

- "Today is Friday"
  - It is not the case that today is Friday
  - Today is not Friday
- At least 10 inches of rain fell today in Miami
  - It is not the case that at least 10 inches of rain fell today in Miami
  - Less than 10 inches of rain fell today in Miami

Conjunction

TABLE 2 The Truth Table for the Conjunction of Two Propositions.

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

Conjunction: p ^ q is true when both p and q are true. False otherwise

Example

- p: "Today is Friday", q: "It is raining today"
- p^q "Today is Friday and it is raining today"
  - true: on rainy Fridays
  - false otherwise:
    - Any day that is not a Friday
    - Fridays when it does not rain

Disjunction

TABLE 3 The Truth Table for the Disjunction of Two Propositions.

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

Disjunction: p v q is false when both p and q are false. True otherwise

Example

- p ˅ q: "Today is Friday or it is raining today"
  - True:
    - Today is Friday
    - It is raining today
    - It is a rainy Friday
  - False
    - Today is not Friday and it does not rain

Exclusive or

TABLE 4 The Truth Table for the Exclusive Or of Two Propositions.

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

Exclusive Or $p \oplus q$ is true when exactly one of p, q is true. False otherwise

Conditional statement

TABLE 5 The Truth Table for the Conditional Statement $p \rightarrow q$.

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

Conditional Statement:
•p is called the premise (or antecedent) and *q* is called the conclusion (or consequent)
•p→ q is false when p is true and q is false. True otherwise

Conditional statement p→q

- Also called an implication

if p, then q	**p implies q**
if p, q	**p only if q**
p is sufficient for q	**a sufficient condition for q is p**
q if p	q whenever p
q when p	q is necessary for p
a necessary condition for p is q	**q follows from p**
q unless ¬ p	

TABLE 5 The Truth Table for the Conditional Statement $p \to q$.

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

Conditional Statement: p→q is false when p is true and q is false. True otherwise
Example
p: you go, q: I go. p→q means "If you go, then I go" is equivalent to p only if q "You go only if I go" (not the same as "I go only if you go" which is q only if p)

p→q

- p only if q:
  - p cannot be true when q is not true
  - The statement is false if p is true but q is false
  - When p is false, q may be either true or false
  - Not to use "q only if p" to express p→q
- q unless ¬ p
  - If ¬ p is false, then q must be true
  - The statement is false when p is true but q is false, but the statement is true otherwise

TABLE 5 The Truth Table for the Conditional Statement $p \to q$.

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

Example

- If Maria learns discrete mathematics, then she will find a good job
  - Maria will find a good job when she learns discrete mathematics (q when p)
  - For Maria to get a good job, it is sufficient for her to learn discrete mathematics (sufficient condition for q is p)
  - Maria will find a good job unless she does not learn discrete mathematics (q unless not p)

Common mistake for p→q

- Correct: p only if q
- Mistake to think "q only if p"

Example

- “If today is Friday, then 2+3=6”
  - The statement is true every day except Friday even though 2+3=6 is false

TABLE 5 The Truth Table for the Conditional Statement $p \rightarrow q$.

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

Converse, contrapositive and inverse

- For conditional statement p→ q
  - Converse: q→ p
  - Contrapositive: ¬q → ¬ p
  - Inverse: ¬p → ¬ q
- Contrapositive and conditional statements are equivalent

Biconditional statement

TABLE 6 The Truth Table for the Biconditional $p \leftrightarrow q$.

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

- Biconditional Statement: “p if and only if q”
- p ←→ q is true when p, q have the same truth value. False otherwise
- Also known as bi-implications

Example

- P: “you can take the flight”, q: “you buy a ticket”
- P ←→ q: “You can take the flight if and only if you buy a ticket”
  - This statement is true
    - If you buy a ticket and take the flight
    - If you do not buy a ticket and you cannot take the flight

Truth table of compound propositions

TABLE 7 The Truth Table of $(p \vee \neg q) \rightarrow (p \wedge q)$.

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

Precedence of logic operators

TABLE 8 Precedence of Logical Operators.

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

$p \vee q \rightarrow r$ $\qquad (p \vee q) \rightarrow r$

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

Bit operations

TABLE 9 Table for the Bit Operators *OR*, *AND*, and *XOR*.

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

1.2 Translating English to logical expressions

Why?

English is often ambiguous and translating sentences into compound propositions removes the ambiguity

Using logical expressions, we can analyze them and determine their truth values

We can use rules of inferences to reason about them

Example

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

p : "You can access the internet from campus"

q : "You are a computer science major"

r : "You are freshmen"

$p \rightarrow (q \vee \neg r)$

System specification

- Translating sentences in natural language into logical expressions is an essential part of specifying both hardware and software systems.
- Consistency of system specification.
- Example: Express the specification "The automated reply cannot be sent when the file system is full"

Example

1. Let p denote "The automated reply can be sent"
2. Let q denote "The file system is full"

The logical expression for the sentence "The automated reply cannot be sent when the file system is full" is

$$q \longrightarrow \neg p$$

Example

Determine whether these system specifications are consistent:

1. The diagnostic message is stored in the buffer or it is retransmitted.
2. The diagnostic message is not stored in the buffer.
3. If the diagnostic message is stored in the buffer, then it is retransmitted.

Example

- Let p denote “The diagnostic message is stored in the buffer”
- Let q denote “The diagnostic message is retransmitted”

The three specifications are

$$p \vee q, \qquad \neg p, \qquad p \rightarrow q$$

Example

- If we add one more requirement “The diagnostic message is not retransmitted”

The new specifications now are

$$p \vee q, \qquad \neg p, \qquad p \rightarrow q \qquad \neg q$$

This is inconsistent! No truth values of p and q will make all the above statements true

Logic gates