SC612: Discrete Mathematics MSc(IT)-Autumn 2023

Lecture 1: Propositional Logic

Instructor: Gopinath Panda *July 27, 2023*

1 What?

- Discrete mathematics is the branch of mathematics devoted to the study of discrete objects.
- The objects are distinct, separate and can be counted.

Digital Watch

Analog Watch

  -
  - .... ________________
  - Please add some more examples.

- Calculus deals with continuous objects and is not part of discrete mathematics.

2 Why?

- to develop your ability to understand and create mathematical arguments.
- the gateway to more advanced courses in all parts of the mathematical sciences.
  - provides the mathematical foundations for many computer science courses including data structures, algorithms, database theory, automata theory, formal languages, compiler theory, computer security, and operating systems.
  - contains the necessary mathematical background for solving problems in operations research, chemistry, engineering, biology, and so on.

3 Important areas

- Logic $\longrightarrow$ writing codes, Algorithms
  - (Propositional Logic, Logical Implication, Logical Equivalence, Predicate Logic, Rules of Inference)

- Set theory $\longrightarrow$ Software engineering, Database
  - (Sets and Set Operations)
- Number theory
- Relation
- Function
- Combinatorics
- Graph theory $\longrightarrow$ Networks, OS, Compilers

4 Propositional Logic

Statements

1. Declarative sentences– assert or declare something, like "Gandhinagar is the capital of Gujarat.", "5 is a complex number",
2. Exclamatory sentences – emotional expressions, such as "Be careful!", "welcome to the course!", "Happy Birth Day!"
3. Interrogative sentences – ask questions, like "what time is it?" "why are you here?"
4. Imperative sentences – give commands, such as "stop at the red light." "Please pass the paper"

Logic

- Logic is the basis of all mathematical reasoning.
- The rules of logic specify the meaning (give precise meaning) of mathematical statements
- It began with the early Greeks and was extensively developed by Aristotle (384-322 BC).
- In computer science, logic has many applications in areas such as database theory, artificial intelligence, program language design, and automated verification of software and hardware.
- In database theory, logic is used to formalize the definitions of queries.
- In artificial intelligence, logic is used to formalize human inference.
- Proving a program to be correct can use logic-based notions such as loop invariants and both pre- and postconditions.
- logic plays a major role during many phases in the design of electronic computers, including the design of efficient combinatorial networks or circuits.

Proposition A proposition is a declarative sentence that is either true or false, but not both.

Note. • The truth value of a proposition is true (T), if it is a true proposition,
• The truth value of a proposition is false (F), if it is a false proposition.
• A proposition is the basic building block of logic.

Example 4.1. • *Washington, D.C., is the capital of the United States of America.*
• *Toronto is the capital of Canada.*
• *1 + 1 = 2.*
• *2 + 2 = 3.*
• *What time is it?*
• *Read this carefully.*
• $x + 1 = 2$.
• $x + y = z$.

Propositional variables Also called statement variables, are variables that represent propositions.
• propositional variables are denoted by $p, q, r, s, \ldots$ and their truth values are denoted by T or F.

Propositional logic The area of logic that deals with propositions. It was first developed systematically by the Greek philosopher Aristotle.

compound propositions New propositions formed from the existing propositions using logical operators (connectives-AND, OR, IF THEN, IF AND ONLY IF; and modifier-NOT).

4.1 Logical operators

Negation Let p be a proposition. The negation of p, denoted by $\neg p$, is the statement "It is not the case that p."
• The proposition $\neg p$ is read "not p."
• The truth value of $\neg p$, is the opposite of the truth value of p.

Table 1: Truth table for $\neg p$

p	$\neg p$
T	F
F	T

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

Remark 4.1. A conjunction is sometimes expressed without using the word "AND".
• I stepped into the bus, but got down near the college.
• Ram came in, while Hari went out.
• The night is dark, though stars are shining.
It is clear from the context of each that it is a conjunction.
There are also cases when 'AND' is used, but it is not a connective;
• "the concert is a combination of vocal and instrumental music".

Table 2: Truth table for $p \wedge q$

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

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

Table 3: Truth table for $p \vee q$

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

Exclusive OR Let p and q be propositions. The exclusive OR of p and q, denoted by $p \oplus q$, is the proposition "p XOR q."
• The disjunction $p \oplus q$ is true when exactly one of p and q is true and is false otherwise.

Table 4: Truth table for $p \oplus q$

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

Remark 4.2. The connective 'OR' is used in both exclusive and inclusive sense. If $p \vee q$ is exclusive, then either of p or q can be true, not both.

4.2 Implication

Let p and q be propositions. The conditional statement $p \to q$ is

- the proposition "if p, then q."
- false when p is true and q is false, and true otherwise.
  - Here, p is called the hypothesis (or antecedent or premise)
  - q is called the conclusion (or consequence).
- commonly expressed as
  - "if p, then q" "p implies q" "if p, q" "p only if q" "q if p"

* p only if q" says that p cannot be true when q is not true.
*Be careful "q only if p" and p $\to$ q are different.

– "p is sufficient for q" "a sufficient condition for q is p" "q when p"
– "q whenever p" "q follows from p" "q unless ¬p"
– "q is necessary for p" "a necessary condition for p is q"

Table 5: Truth table for $p \to q$

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

Example. *If I am director, then I will install AC in classrooms.*

• If I am selected as the director, but didn't install ACs in the classroom. Then I am a liar. I didn't keep my promise. ($T \to F$ is F)
• If I am not selected as the director, whether or not ACs are installed in the classroom. I am not a liar. I didn't break my promise. ($F \to q$ always T)

Converse Let p and q be propositions. The converse statement of the conditional statement $p \to q$ is denoted by $q \to p$ and is the proposition
• "if q, then p."
• The converse statement $q \to p$ is false when p is false and q is true, and true otherwise.

Table 6: Truth table for $q \to p$

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

Contrapositive Let p and q be propositions. The contrapositive statement of the conditional statement $p \to q$ is denoted by $\neg q \to \neg p$ and is the proposition
• "if p, then q."
• The contrapositive statement $\neg q \to \neg p$ has the same truth value as $p \to q$.

Table 7: Truth table for $\neg q \to \neg p$

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

Inverse Let p and q be propositions. The inverse statement of the conditional statement $p \to q$ is denoted by $\neg p \to \neg q$ and is the proposition

- "if $\neg p$, then $\neg q$."
- The inverse statement $\neg p \to \neg q$ has the same truth value as $p \to q$.

Table 8: Truth table for $\neg q \to \neg p$

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

Biconditional Let p and q be propositions. The biconditional statement $p \leftrightarrow q$

- is the proposition "p if and only if q."
- is true when p and q have same truth values, and false otherwise.
- Common ways to express the biconditional statement:
  - "if p, then q and conversely"
  - "p iff q"
  - "p is necessary and sufficient for q"

Table 9: Truth table for $p \leftrightarrow q$

p	q	$p \to q$	$q \to p$	$p \leftrightarrow q$	$(p \to q) \wedge (q \to p)$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	T	F	F	F
F	F	T	T	T	T

Note. • Necessary but not sufficient condition and vice versa:
p is necessary condition for q : $q \to p$ p is not necessary condition for q : $\neg(q \to p)$
p is sufficient condition for q : $p \to q$ p is not sufficient condition for q : $\neg(p \to q)$

- p is necessary but not sufficient condition for q : $(q \to p) \wedge \neg(p \to q)$
- p is not necessary but sufficient condition for q : $(p \to q) \wedge \neg(q \to p)$

Precedence of Logical Operators

1. Negation $\neg$
2. Conjunction $\wedge$
3. Disjunction $\vee$
4. Conditional $\to$
5. Biconditional $\leftrightarrow$

Example 4.2. *Construction of Truth table for compound propositions.*

Construct the truth table of the compound proposition

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

Solution: Because this truth table involves two propositional variables p and q, there are four rows in this truth table, one for each of the pairs of truth values TT, TF, FT, and FF. The first two columns are used for the truth values of p and q, respectively. In the third column we find the truth value of $\neg q$, needed to find the truth value of $p \vee \neg q$, found in the fourth column. The fifth column gives the truth value of $p \wedge q$. Finally, the truth value of $(p \vee \neg q) \rightarrow (p \wedge q)$ is found in the last column.

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

Classification of compound propositions

- Tautology: A compound proposition that is always true, no matter what the truth values of the propositional variables that occur in it.
- Contradiction: A compound proposition that is always false.
- Contingency: A compound proposition that is neither a tautology nor a contradiction

Table 10: Examples of Tautology and Contradiction

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

4.3 Logic and Bit Operations

Bit

- A bit is a symbol with two possible values, namely, 0 (zero) and 1 (one).
  - It comes from Binary digIT, because zeros and ones are the digits used in binary representations of numbers.
  - Statistician John Tukey introduced this terminology in 1946.
- A bit can be used to represent a truth value, because there are two truth values, namely, true and false.
  - a 1 bit represent true and
  - a 0 bit to represent false.

- That is, 1 represents T (true), 0 represents F (false).

Truth Value	Bit
T	1
F	0

Boolean variable

- A variable is called a Boolean variable if its value is either true or false.
- Consequently, a Boolean variable can be represented using a bit.
- bit operations correspond to the logical connectives.
- By replacing true by a one and false by a zero in the truth tables for the operators $\wedge$, $\vee$, and $\oplus$,

Table 11: 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

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

Example 4.3. • *1011 is a bit string of length four.*
• *111010011 is a bit string of length nine.*

Bitwise operators Bit operations can be extended to bit strings.
• Define the bitwise OR, bitwise AND, and bitwise XOR of two strings of the same length to be the strings that have as their bits the OR, AND, and XOR of the corresponding bits in the two strings, respectively.

Example 4.4. *Find the bitwise OR, bitwise AND, and bitwise XOR of the bit strings 01 1011 0110 and 11 0001 1101.*

4.4 Applications of Propositional Logic

- Statements in mathematics and the sciences and in natural language often are imprecise or ambiguous. To make such statements precise, they can be translated into the language of logic.
- Logic can be used to analyze and solve many familiar puzzles.
- Propositional logic and its rules can be used to
  - design computer circuits,
  - construct computer programs,
  - verify the correctness of programs, and
  - build expert systems.

Table 12: Table for the Bit Operators OR, AND, and XOR.

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

1. Translating English Sentences There are many reasons to translate English sentences into expressions involving propositional variables and logical connectives.

• In particular, English is often ambiguous. Translating sentences into compound statements removes the ambiguity.

• Once we have translated sentences from English into logical expressions, we can analyze these logical expressions to determine their truth values, we can manipulate them, and we can use rules of inference to reason about them.

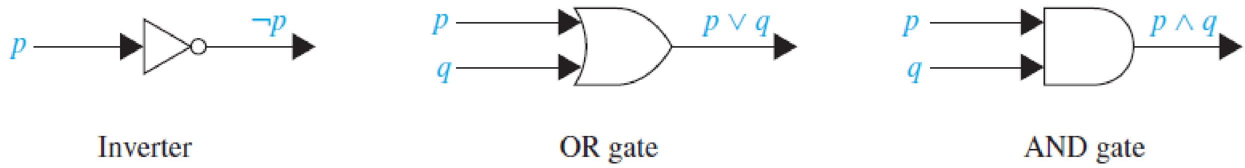
Example 4.5. *"You can access the Internet from campus only if you are a computer science major or you are not a freshman."*

- Let p:= "You can access the Internet from campus,"
- q:= "You are a computer science major," and
- r:= "You are a freshman" $\neg r$:= "You are not a freshman"
- Connectives are "only if" and "or"
- This sentence can be represented as $p \to (q \vee \neg r)$.

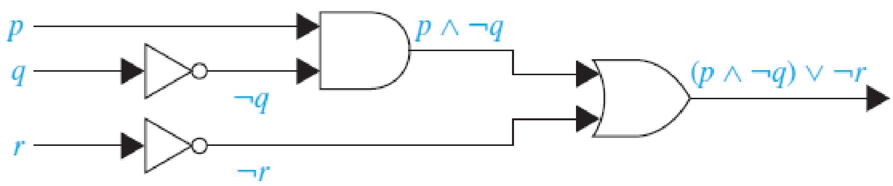
2. Boolean searches Logical connectives are used extensively in searches of large collections of information, such as indexes of Web pages.

3. Logic puzzles

4. Logic circuits A logic circuit (or digital circuit) receives input signals $p_1, p_2, \ldots, p_n$, each a bit, and produces output signals $s_1, s_2, \ldots, s_n$, each a bit.



Basic logic gates.



A combinatorial circuit.