Unit 1: Propositional Logic

Dr Maria Thomas

September 1, 2023

References

- Kenneth H Rosen, “Discrete Mathematics and its Applications”, Tata McGraw Hill, 2016.
- Y.N Singh, “Mathematical Foundation of computer science”, New Age International Publishers, New Delhi, 2005.

PROPOSITIONAL LOGIC

- To understand to construct correct mathematical arguments.
- The rules of logic give precise meaning to mathematical statements.
- These rules are used to distinguish between valid and invalid mathematical arguments.
- Applications: These rules are used in the design of computer circuits, the construction of computer programs, the verification of the correctness of programs, and in many other ways. Furthermore, software systems have been developed for constructing proofs automatically.

Propositions

- The basic building blocks of logic.
- A proposition is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.
- Examples:

  1. Bengaluru is the capital of Karnataka.
  2. 2+2=5.

3. Uttar Pradesh is the capital of India.
4. 1+1=2.

- Exercise:
  1. What time is it?
  2. $x + 1 = 2$
  3. $x + y = z$
  4. 1+1=2.
  5. Answer this question.
  6. There are no mosquitoes in this room.
- Exercise-Answers:
  1. Not declarative. Hence not a proposition.
  2. Not a proposition because it is neither true or false.
  3. Not a proposition because it is neither true or false.
  4. Proposition.
  5. Not declarative. Hence not a proposition.
  6. Proposition.
- Letters are used to denote propositions. The conventional letters are $p, q, r, s, ...$
- Examples:
  - p: I went to the university today.
  - q: $1 + 5 = 6$.
- The area of logic that deals with propositions is called the **propositional calculus** or **propositional logic**.
- It was first developed systematically by the Greek philosopher Aristotle more than 2300 years ago.

Definition: Negation Operator

- Let p be a proposition. The negation of p, denoted by $\neg p$ (also denoted by $\bar{p}$), is the statement "It is not the case that p."

- The proposition $\neg p$ is read "not p."

- The truth value of the negation of p is the opposite of the truth value of p.

- Example:
  - p: "Today is Friday."
  - $\neg p$: "Today is not Friday." or "It is not Friday today."
  - q: "At least 10mm of rain was received today in Bengaluru."
  - $\neg q$: "It is not the case that at least 10mm of rain was received today in Bengaluru." or "Less than 10mm of rain was received today in Bengaluru."

Remark

- Strictly speaking, sentences involving variable times such as those in the above examples are not propositions unless a fixed time is assumed. The same holds for variable places unless a fixed place is assumed and for pronouns unless a particular person is assumed. We will always assume fixed times, fixed places, and particular people in such sentences unless otherwise noted.

- Truth table: Negation of a Proposition

p	$\neg p$
T	F
F	T

Definition: Connectives

- Connectives are logical operators that are used to form new propositions from two or more existing propositions.

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

Definition: 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.

- Truth table: Conjunction of Two Propositions

- In logic the word "but" sometimes is used instead of "and" in a conjunction.
  - Example: "The sun is shining, but it is raining" is another way of saying "The sun is shining and it is raining."

Exercise

- Let p and q be the propositions: p : It is below freezing. q : It is snowing.

- $p \wedge q$: It is below freezing and it is snowing. This proposition is true only when it is below freezing and snowing.

Definition: 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.

- Disjunction is also termed as inclusive or.

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

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

- Truth table: Disjunction of Two Propositions

- The use of the connective or in a disjunction corresponds to one of the two ways the word "or" is used in English, namely, in an inclusive way.

Exercise

- Let p and q be the propositions: p : It is below freezing. q : It is snowing.

- $p \vee q$: It is below freezing or it is snowing. This proposition is true when it is below freezing or it is snowing.

Definition: Exclusive Or

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

- Truth table: Exclusive Or of Two Propositions

Conditional Statements

- Let p and q be propositions. The conditional statement $p \rightarrow q$ is the proposition "if p, then q." The conditional statement $p \rightarrow q$ is false when p is true and q is false, and true otherwise.

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

- In the conditional statement $p \to q$, p is called the hypothesis (or antecedent or premise) and q is called the conclusion (or consequence).

- A conditional statement $p \to q$ asserts that q is true on the condition that p holds. A conditional statement is also called an **implication**.

- Truth table: Implication

- Example: "If you get 100% on the final, then you will get an A."

  p: You get 100% on the final. q: You will get an A.

  - If you manage to get a 100 % on the final, then you would expect to receive an A.
  - If you do not get 100% you may or may not receive an A depending on other factors.
  - If you do get 100 %, but the professor does not give you an A, you will feel cheated.

- Determine whether each of these conditional statements is true or false.

  - If $1+1=2$, then $2+2=5$.
  - If $1+1=3$, then $2+2=4$.
  - If $1+1=3$, then $2+2=5$.
  - If monkeys can fly, then $1+1=3$.
  - If monkeys cannot fly, then $1+1=2$.

  - False
  - True
  - True
  - True
  - True

- Determine whether each of these conditional statements is true or false.
  - If $1 + 1 = 3$, then donkeys exists .
  - If $1 + 1 = 3$, then dogs can fly.
  - If $1 + 1 = 2$, then dogs can fly.
  - If$2 + 2 = 4$, then $1 + 2 = 3$.

  - True
  - True
  - False
  - True

NOTES:

- **Propositional language is an artificial language; we only parallel English usage to make it easy to use and remember.**

- **The if-then construction used in many programming languages is different from that used in logic.**

  - Most programming languages contain statements such as "if p then S", where p is a proposition and S is a program segment (one or more statements to be executed).

  - S is executed if p is true, but S is not executed if p is false.

- **Converse**: The proposition $q \rightarrow p$ is called the converse of $p \rightarrow q$.

- **Contrapositive**: The contrapositive of $p \rightarrow q$ is the proposition $\neg q \rightarrow \neg p$.

- **Inverse**: The proposition $\neg p \rightarrow \neg q$ is called the inverse of $p \rightarrow q$.

- Truth Table of Converse.

- Truth Table of Contrapositive.

- Truth Table of Inverse

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

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

Equivalent Statements

- Two compound propositions are said to be equivalent if they have the same truth value.

- Examples:
  - A conditional statement and its contrapositive are equivalent.
  - The converse and the inverse of a conditional statement are also equivalent.

- **The converse and the inverse of a conditional statement are not equivalent to the original conditional statement.**

Exercise

- State the converse, contrapositive, and inverse of the conditional statement. "I come to class whenever there is going to be a quiz."

- Converse: "If I come to class, then there will be a quiz."

- Contrapositive: "If I do not come to class, then there will not be a quiz."

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

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

- Inverse: "If there is not going to be a quiz, then I don't come to class."
- Let p and q be propositions. The biconditional statement $p \longleftrightarrow q$ is the proposition "p if and only if q." The biconditional statement $p \longleftrightarrow q$ is true when p and q have the same truth values, and is false otherwise.

- Biconditional statements are also called bi-implications.

- $p \longleftrightarrow q$ can also be stated as
  - "p is necessary and sufficient for q ."
  - "if p then q , and conversely."
  - "p iff q ."

- $p \longleftrightarrow q$ has exactly the same truth value as $(p \rightarrow q) \wedge (q \rightarrow p)$.

Precedence of Logical Operators

Important Note:

- Consider the statement $p \rightarrow q$.
- p is a sufficient condition got q.
- q is a necessary condition for p.

- Example: "If I opened the door, I used the key."
- p: "I opened the door."
- q: "I used the key."
- Note: q can be true when p is false. However, p can be true "only" when q is true.

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

Truth Tables of Compound Propositions

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

-

Truth Tables of Compound Propositions

- $(p \rightarrow q) \longleftrightarrow (\neg q \rightarrow \neg p)$.

Application 1: Translating English Sentences

- Translating English sentences into expressions involving propositional variables and logical connectives to avoid ambiguity.

- The translated sentences from English into logical expressions can be analyzed to determine their truth values. Then we can manipulate them, and we can use 'rules of inference' to reason about them.

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

- Let q: "You can ride the roller coaster," r: "You are under 4 feet tall," s:"You are older than 1 6 years old,". Then the sentence can be translated

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

Application 2: System Specifications

- Translating sentences in natural language (such as English) into logical expressions is an essential part of specifying both hardware and software systems.

- System and software engineers take requirements in natural language and produce precise and unambiguous specifications that can be used as the basis for system development.

Example

- Express the specification "The automated reply cannot be sent when the file system is full" using logical connectives.

- Let p: "The automated reply can be sent" and q:"The file system is full."

- Then the given statement can be represented as $q \longrightarrow \neg p$.

- System specifications should be **consistent**, that is, they should not contain conflicting requirements that could be used to derive a contradiction.

- When specifications are not consistent, there would be no way to develop a system that satisfies all specifications.

Example

- Determine whether these system specifications are consistent: r:"The diagnostic message is stored in the buffer or it is retransmitted."

  s:"The diagnostic message is not stored in the buffer."

  t: "If the diagnostic message is stored in the buffer, then it is retransmitted."

- Solution:

- Let p: "The diagnostic message is stored in the buffer" and q: "The diagnostic message is retransmitted."

- Then, r can be expressed as $p \vee q$, s as $\neg p$ and t as $p \rightarrow q$.

Solution ctd.

- From s for $\neg p$ to be true, p must be false.

- Given p is false, $p \vee q$ can be true only when q is true. Hence we have that p is false and q is true from the statements r and s. Now, we check whether t holds true with p is false and q is true.

- Given p is false and q is true, $p \rightarrow q$ is true.

- Hence we can conclude that the system of specifications given in r, s and t are consistent.
- Do the above system specifications remain consistent if the specification "The diagnostic message is not retransmitted" is added?
- u: "The diagnostic message is not retransmitted"
- Then u: $\neg q$.
- We have from r, s and t that p is false and q are true.
- However, u is false when q as true.
- Hence these four specifications, r, s, t and u, are inconsistent.

Application 3: Boolean Searches

- Logical connectives are used extensively in searches of large collections of information, such as indexes of Web pages.
- Because these searches employ techniques from propositional logic, they are called Boolean searches.
- Example: Boolean searching to find Web pages about universities in New Mexico, we can look for pages matching NEW AND MEXICO AND UNIVERSITIES. The results of this search will include those pages that contain the three words NEW, MEXICO, and UNIVERSITIES.

Application 4: Logic Puzzles

- Puzzles that can be solved using logical reasoning are known as logic puzzles.
- Example: Each inhabitant of a remote village always tells the truth or always lies. A villager will only give a "Yes" or a "No" response to a question a tourist asks. Suppose you are a tourist visiting this area and come to a fork in the road. One branch leads to the ruins you want to visit; the other branch leads deep into the jungle. A villager is standing at the fork in the road. What one question can you ask the villager to determine which branch to take?

Application 5: Logic and Bit Operations

- Computer 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 $\oplus$ for the corresponding bit operations can be obtained. We use the notation OR, AND, and XOR for the operators, $\wedge$, $\vee$ and $\oplus$ respectively, as is done in various programming languages.

- 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 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.

- Find the bitwise OR, bitwise AND, and bitwise XOR of each of 11110000, 10101010.

Fuzzy Logic

- Fuzzy logic is used in artificial intelligence. In fuzzy logic, a proposition has a truth value that is a number between 0 and I , inclusive.

- A proposition with a truth value of 0 is false and one with a truth value of 1 is true.

- Truth values that are between 0 and 1 indicate varying degrees of truth. For instance, the truth value 0.8 can be assigned to the statement "Fred is happy," because Fred is happy most of the time, and the truth value 0.4 can be assigned to the statement "John is happy," because John is happy slightly less than half the time.

Propositional Equivalences

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

- Example: $p \vee \neg p$

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

- Example: $p \wedge \neg p$.

- **Contingency**: A compound proposition that is neither a tautology nor a contradiction is called a contingency.

Logical Equivalences

- The compound propositions p and q are called logically equivalent if $p \longleftrightarrow q$ is a tautology.

- The notation $p \equiv q$ denotes that p and q are logically equivalent.

- $p \equiv q$ is also represented as $p \Leftrightarrow q$.

- **Imp. Note**: $\equiv$ is not a logical connective and $p \equiv q$ is not a compound proposition.

- $p \equiv q$ is a statement "$p \longleftrightarrow q$ is a tautology."

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

List of Logical Equivalences

TABLE 6 Logical Equivalences.

Equivalence	*Name*
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

List of Logical Equivalences

TABLE 7 Logical Equivalences Involving Conditional Statements.
$p \to q \equiv \neg p \lor q$
$p \to q \equiv \neg q \to \neg p$
$p \lor q \equiv \neg p \to q$
$p \land q \equiv \neg(p \to \neg q)$
$\neg(p \to q) \equiv p \land \neg q$
$(p \to q) \land (p \to r) \equiv p \to (q \land r)$
$(p \to r) \land (q \to r) \equiv (p \lor q) \to r$
$(p \to q) \lor (p \to r) \equiv p \to (q \lor r)$
$(p \to r) \lor (q \to r) \equiv (p \land q) \to r$

List of Logical Equivalences

TABLE 8 Logical Equivalences Involving Biconditionals.
$p \leftrightarrow q \equiv (p \to q) \land (q \to p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \land q) \lor (\neg p \land \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Introduction to Predicate

Consider the statements:

- "$x > 3$"
- "computer x is functioning properly"
- These statements are neither true nor false when the values of the variables are not specified.
- We will discuss the ways that propositions can be produced from such statements.

Predicate

Consider the statement:

- "$x > 3$"

- The first part, the variable x, is the subject of the statement.

- The second part, called the predicate- "is greater than 3"-refers to a property that the subject of the statement can have.

- We take P to be the predicate "is greater than 3" and x as the variable.

- Then can denote the statement "x is greater than 3" by $P(x)$.

- The statement $P(x)$ is also said to be the value of the propositional function P at x.

Predicate ctd.

- Once a value has been assigned to the variable x , the statement $P(x)$ becomes a proposition and has a truth value.

- Let $P(x)$ denote the statement "$x > 3$." What are the truth values of $P(4)$ and $P(2)$?

- Let $A(x)$ be the statement "the word x contains the letter a." What are the truth values of 1. $A(orange)$ 2. $A(lemon)$ 3. $A(true)$ 4. $A(false)$.

- Let $Q(x, y)$ denote the statement "$x = y + 3$." What are the truth values of the propositions $Q(1, 2)$ and $Q(3, O)$?

Definition

- A statement of the form $P(x_1, x_2, x_3, ..., x_n)$ is the value of the propositional function P at the n-tuple $(x_1, x_2, x_3, ..., x_n)$, and P is also called a n-place predicate or a n-ary predicate.

Propositional functions in computer programs

- Consider the statement
  - "if $x > 0$ then $x := x + 1$."

  - Let $P(x)$ be "$x > 0$."

  - When this statement is encountered in a program, the value of the variable x at that point in the execution of the program is inserted into $P(x)$.

  - If $P(x)$ is true for this value of x, the assignment statement $x := x+1$ is executed, so the value of x is increased by 1 .

  - If $P(x)$ is false for this value of x , the assignment statement is not executed, so the value of x is not changed.

Quantification

- Quantification expresses the extent to which a predicate is true over a range of elements.

- Two types:
  1. Universal quantification
  2. Existential quantification

Universal Quantification

- The universal quantification of $P(x)$ is the statement "$P(x)$ for all values of x in the domain."

- Notation: $\forall x\ P(x)$ denotes the universal quantification of $P(x)$.

- $\forall$ is called the universal quantifier.

- Example: Let $P(x)$ b e the statement "$x + 1 > x$." What is the truth value of the quantification $\forall x\ P(x)$ where the domain consists of all real numbers?

- Example: Suppose that $P(x)$ is "$x^2 > 0$." What is the truth value of the quantification $\forall x\ P(x)$ where the domain consists of all integers?

Existential Quantification

- The existential quantification of $P(x)$ is the proposition "There exists an element x in the domain such that $P(x)$."

- Notation: $\exists x\ P(x)$ denotes the existential quantification of $P(x)$.

- $\exists$ is called the existential quantifier.

- Example: Let $P(x)$ b e the statement "$x > 3$." What is the truth value of the quantification $\exists x\ P(x)$ where the domain consists of all real numbers?

- Example: Suppose that $Q(x)$ is "$x = x + 1$." What is the truth value of the quantification $\exists x\ Q(x)$ where the domain consists of all real numbers?

- An **argument** in propositional logic is a sequence of propositions. All but the final proposition in the argument are called premises and the final proposition is called the conclusion.

- An argument is valid if the truth of all its premises implies that the conclusion is true.

- An argument form in propositional logic is a sequence of compound propositions involving propositional variables.

- An argument form is valid if no matter which particular propositions are substituted for the propositional variables in its premises, the conclusion is true if the premises are all true.

Thoery of Inference/Rules of Inference

- Rules of inference gives the validity of some relatively simple argument forms.

- Rule of Inferennce 1: The tautology $(p \wedge (p \rightarrow q)) \rightarrow q$ is the basis of the rule of inference called modus ponens or the law of detachment.

  $p\ p \rightarrow q$ (0,0) – (1,0);
  $\therefore q$

Example

1 The statement "If it snows today, then we will go skiing" and and its hypothesis, "It is snowing today," are true. Then, by modus ponens, it follows that the conclusion of the conditional statement, "We will go skiing," is true.

2 " If $\sqrt{2} > \frac{3}{2}$, then $(\sqrt{2})^2 > \left(\frac{3}{2}\right)^2$." p $p \to q$ (0,0) – (1,0);

$\therefore$ q This argument is valid because it is constructed by using modus ponens, a valid argument form. But one of its premises, $\sqrt{2} > \frac{3}{2}$, is false. Consequently, we cannot conclude that the conclusion is true.

Rule of Inferennce 2:

- Rule of Inferennce 2: Modus tollens- The tautology $(\neg q \wedge (p \to q)) \to \neg p$

$\neg q$ $p \to q$ (0,0) – (1,0);

$\therefore \neg p$

Rules of Inteference

TABLE 1 Rules of Inference.

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