Logic Concepts

Introduction

→ One of the prime activity of human intelligence is reasoning.

→ The activity of reasoning involves construction, organization and manipulation of statements to arrive at new conclusion.

→ Thus logic can be defined as a scientific study of the process of reasoning and the system of rules and proposition procedures that help in distinguish correct reasoning from incorrect reasoning.

→ Basically, the logic process takes in some function (called premises) and produces some output (called conclusions). There exist some basic laws that help in Theorem proving.

→ Logic is basically classified into 2 main categories:
① Propositional logic ② Predicate logic

Propositional Calculus (PC)

→ PC refers to a language of propositions in which a set of rules are used to combine simple propositions to form compound propositions with the help of certain logical operators.

→ These logical operators c/d "Connectives" ($\sim, \wedge, \vee, \rightarrow, \leftrightarrow$)

→ A well formed formula (w.f.f) is defined as a symbol or a string of symbols generated by a formal grammar of a formal language.
- A smallest unit (atom) is considered to be well formed formula.
- If α is wff then $\sim\alpha$ is also wff.
- If α & β are wff then $(\alpha \vee \beta)$, $(\alpha \wedge \beta)$, $(\alpha \rightarrow \beta)$ and $(\alpha \leftrightarrow \beta)$ are also well formed formulae.

Truth Table.

→ In PC, a truth table is used to provide oprational definitions of important logical operators; it elaborates all possible truth values of a formula.

→ Let us assume that A, B, C are prepositioned symbols.

Truth Table for logical connectives (operators)

A	B	$\sim A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

Q. Compute the truth value of $\alpha : (A \vee B) \wedge (\sim B \rightarrow A)$ using truth table approach.

Logical Equivalent

Two formulae α & β are said to be logically equivalent $(\alpha \cong \beta)$ if and only if the truth values of both are same for all possible assignments of logical constants (T or F) to the symbols appearing in the formulae.

Q. Which of the following pair of expressions are logical equivalent? Show by using truth table

1) $[A \wedge B \wedge C,\ A \wedge (B \vee C)]$
2) $[A \rightarrow (B \vee C),\ \sim A \vee B \vee C]$
3) $[(A \wedge \sim B) \rightarrow C,\ \sim (A \wedge \sim B \wedge \sim C)]$
4) $[(A \rightarrow B) \rightarrow C,\ A \rightarrow (B \rightarrow C)]$
5) $[A \vee \sim B \rightarrow C,\ A \vee (\sim B \rightarrow C)]$

Tautology: If and only if value of α is true for all its interpretations (row) DATE

Q. Consider A, B, C and D to be propositional symbols. Which of these formulae are tautologies? Show using truth table.

1) $\sim(A \vee B \wedge C)$
2) $(A \rightarrow B) \rightarrow (A \rightarrow \sim B)$
3) $(A \leftrightarrow B) \leftrightarrow (A \rightarrow B) \wedge (B \rightarrow A)$
4) $A \rightarrow (B \vee C) \rightarrow D$

Equivalence Laws: equivalence relation are used to reduce or simplify a given well formed formula or derive a new formula from the existing one.

These laws can be verified using truth table approach.

Name of Relation	Equivalence law
1 Commutative law	$A \vee B \cong B \vee A$ $A \wedge B \cong B \wedge A$
2 Associative law	$A \vee (B \vee C) \cong (A \vee B) \vee C$ $A \wedge (B \wedge C) \cong (A \wedge B) \wedge C$
3 Distributive law	$A \vee (B \wedge C) \cong (A \vee B) \wedge (A \vee C)$ $A \wedge (B \vee C) \cong (A \wedge B) \vee (A \wedge C)$
4 Double Negation	$\sim(\sim A) \cong A$
5 De Morgan's law	$\sim(A \vee B) \cong \sim A \wedge \sim B$ $\sim(A \wedge B) \cong \sim A \vee \sim B$
6 Absorption law	$A \vee (A \wedge B) \cong A$ $A \wedge (A \vee B) \cong A$ $A \vee (\sim A \wedge B) \cong A \vee B$ $A \wedge (\sim A \vee B) \cong A \wedge B$
7 Idempotence law	$A \vee A \cong A$ $A \wedge A \cong A$

Propositional logic

→ PL deals with the validity, satisfiability (also called consistency) & unsatisfiability (inconsistency) of a formula. and derivation of new formula using equivalence laws.

→ A formula α is said to be valid if and only if it is a tautology.

→ A formula α is said to be satisfiable if there exists at least one interpretation (row) for which α is true.

→ A formula α is said to be unsatisfiable if the value of α is false under all interpretations.

Q. Show that the following is valid argument:
If it is humid then it will rain and since it is humid today it will rain

Solution: A : It is humid B : It will rain
α : $[(A \to B) \land A] \to B$

A	B	$A \to B = (X)$	$X \land A = (Y)$	$Y \to B$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

Tautology

→ Limitation of this method lies in the fact that the size of the truth table goes exponentially. That is, if the formula contains n atoms, then truth table will contain 2^n entries.

classmate

→ Therefore, we require some other methods which can help in proving the validity of the formula directly.

1. Natural Deduction System
2. Axiomatic System
3. Semantic Tableau method
4. Resolution Refutation method.

Natural Deduction System (NDS)

→ NDC is thus called because of the fact that it mimics the pattern of natural reasoning. This system is based on a set of deductive inference rules.

→ Assuming that $A_1 -- A_k$, $1 \leq k \leq n$, are set of atoms and α_j, where $1 \leq j \leq m$ and β are well formed formulae, the inference rules may be stated as :

NDS Rules Table.

Rule Name	Symbol	Rule	Description.
Introducing $\wedge$	$(I:\wedge)$	If $A_1 -- A_n$ then $A_1 \wedge A_2 -- \wedge A_n$	If $A_1, -- A_n$ are true, then their conjunction $A_1 \wedge -- A_n$ is also true.
Eliminating $\wedge$	$(E:\wedge)$	If $A_1 \wedge -- \wedge A_n$ then A_i $(1 \leq i \leq n)$	If $A_1 \wedge -- \wedge A_n$ is true, then any A_i is also true.
Introducing $\vee$	$(I:\vee)$	If any (A_i) then $A_1 \vee -- \vee A_n$	If any A_i $(1 \leq i \leq n)$ is true, then $A_1 \vee -- \vee A_n$ is also true
Eliminating $\vee$	$(E:\vee)$	If $A_1 \vee -- \vee A_n$, $A_1 \rightarrow A$, -- $A_n \rightarrow A$ then A	If $A_1 \vee -- \vee A_n$, $A_1 \rightarrow A$, $A_2 \rightarrow A$ and $A_n \rightarrow A$ are true, then A is true.
Introducing $\rightarrow$	$(I:\rightarrow)$	If from α_1 -- α_m infer β is proved then $\alpha_1 \wedge -- \wedge \alpha_n \rightarrow \beta$ is proved	If given that α_1, α_2 -- and α_n are true and from these we deduce β then $\alpha_1 \wedge -- \wedge \alpha_n \rightarrow \beta$ is also true.

Eliminating $\rightarrow$	$(E : \rightarrow)$	If $A_1 \rightarrow A$, A_1 then A	If $A_1 \rightarrow A$ and A_1 are true then A is also true. This is cld Modus Ponen Rule.
Introducing $\leftrightarrow$	$(I : \leftrightarrow)$	If $A_1 \rightarrow A_2$, $A_2 \rightarrow A_1$ then $A_1 \leftrightarrow A_2$	If $A_1 \rightarrow A_2$ and $A_2 \rightarrow A_1$ are true then $A_1 \leftrightarrow A_2$ is also true.
Eliminating $(\leftrightarrow)$	$(E : \leftrightarrow)$	If $A_1 \leftrightarrow A_2$ then $A_1 \rightarrow A_2$, $A_2 \rightarrow A_1$	If $A_1 \leftrightarrow A_2$ is true then $A_1 \rightarrow A_2$ and $A_2 \rightarrow A_1$ are true.
Introducing $\sim$	$(I : \sim)$	If from A infer $A_1 \wedge \sim A_1$ is proved then $\sim A$ is proved.	If from A (which is true), a contradiction is proved then truth of $\sim A$ is also proved.
Eliminating $\sim$	$(E : \sim)$	If from $\sim A$ infer $A_1 \wedge \sim A_1$ is proved then A is proved	If from $\sim A$, a contradiction is proved then the truth of A is also proved.

→ A theorem in the NDS written as from $\alpha_1 --- \alpha_n$ infer β leads to the interpretation that β is deducted from the set of hypotheses $\{\alpha_1 --- \alpha_n\}$.
A theorem that is written as infer β implies that there are no hypotheses and β is true under all interpretations.

Q. Prove the following theorems using deductive inference rule

1) from $A \rightarrow B \wedge C$, A infer C
2) from $Q \rightarrow P$, $Q \rightarrow R$ infer $Q \rightarrow (P \wedge R)$.
3) from $A \leftrightarrow B$, B infer A.

4) from $A \wedge B$ infer $A \wedge (B \vee C)$

or Q: prove that $A \wedge (B \vee C)$ is deduced from $A \wedge B$.

Solution

Description	Formula	Comments.
Theorem	from $A \wedge B$ infer $A \wedge (B \vee C)$	To be proved.
Hypothesis (given)	$A \wedge B$	1
E: $\wedge$ (1)	A	2
E: $\wedge$ (1)	B	3
I: $\vee$ (3)	$B : C$	4
I: $\wedge$ (2,4)	$A \wedge (B \vee C)$	proved.

Deduction Theorem in NDS

To prove a formula $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ it is sufficient to prove a theorem from $\alpha_1, \ldots \alpha_n$ infer β. Conversely, if $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ is proved then the theorem from $\alpha_1, \ldots \alpha_n$ infer β is assumed to be proved.

Q. prove the following theorems in natural deduction system.

1) infer $(A \rightarrow B) \rightarrow (\sim A \rightarrow \sim A)$
2) infer $\sim A \rightarrow (A \rightarrow B)$
3) infer $A \rightarrow (\sim B \rightarrow \sim A)$
4) infer $(A \rightarrow B) \rightarrow [(\sim A \rightarrow B) \rightarrow B]$
5) infer $A \wedge B \leftrightarrow B \wedge A$.

DATE

Q. Prove the theorem infer $[(A \to B) \wedge (B \to C)] \to (A \to C)$

Solⁿ: The theorem infer $[(A \to B) \wedge (B \to C)] \to (A \to C)$ is reduced to the theorem from $(A \to B), (B \to C)$ using deduction theorem. further, to infer $A \to C$ we will have to prove a subtheorem prove '$A \to C$' from A infer C.

Discription	Formula	Comments
Theorem	from $A \to B, B \to C$ infer $A \to C$	To be proved
Hypothesis 1	$A \to B$	1
Hypothesis 2	$B \to C$	2
Subtheorem	from A infer C	3
Hypothesis	A	3.1
E: $\to$(1, 3.1)	B	3.2
E: $\to$ (2, 3.2)	C	3.3
I: $\to$ (3)	$A \to C$	proved.

Axiomatic System

→ The axiomatic system is based on a set of 3 axioms & one rule of deduction.

→ In this system, only two logical operators not ($\sim$) and implies ($\rightarrow$) are allowed to form a formula.

→ Other logical operators can be easily expressed in terms of $\sim$ and $\rightarrow$ using equivalence law:

$$A \wedge B \cong \sim(\sim A \vee \sim B) \cong \sim(A \rightarrow \sim B)$$
$$A \vee B \cong \sim A \rightarrow B$$
$$A \leftrightarrow B \cong (A \rightarrow B) \wedge (B \rightarrow A) \cong \sim[(A \rightarrow B) \rightarrow \sim(B \rightarrow A)]$$

→ Axiom 1 $\alpha \rightarrow (\beta \rightarrow \alpha)$

Axiom 2 $[\alpha \rightarrow (\beta \rightarrow \gamma)] \rightarrow [(\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma)]$

Axiom 3 $(\sim\alpha \rightarrow \sim\beta) \rightarrow (\beta \rightarrow \alpha)$

Modus Ponen Rule. Hypotheses : $\alpha \rightarrow \beta$ and α
Consequent : β.

Interpretation of MP Rule: If given that $\alpha \rightarrow \beta$ and α are hypotheses (assumed to be true), β is inferred (i.e. true) as a consequent.

→ Let $\Sigma = \{\alpha_1 --- \alpha_n\}$ be a set of hypothesis.
The formula α is defined to be a "deductive consequence" of Σ if either α is an axiom or hypothesis or is derived from α_j where $1 \leq j \leq n$, using MP rule.

It is represented as $\{\alpha_1 -- \alpha_n\} \vdash \alpha$
$\Sigma \vdash \alpha$

Q. Establish that $A \to C$ is a deductive consequence of $\{A \to B, B \to C\}$ i.e,

$$\{A \to B, B \to C\} \vdash (A \to C)$$

Description	Formula	Comments
Theorem	$\{A \to B, B \to C\} \vdash (A \to C)$	Prove
Hypothesis 1	$A \to B$	1
Hypothesis 2	$B \to C$	2
Instance of Axiom 1	$(B \to C) \to [A \to (B \to C)]$	3
MP (2, 3)	$[A \to (B \to C)]$	4
Instance of Axiom 2	$[A \to (B \to C)] \to [(A \to B) \to (A \to C)]$	5
MP (4, 5)	$(A \to B) \to (A \to C)$	6
MP (1, 6)	$A \to C$	Proved.

Q. Prove the following in axiomatic system

① $\{A \to (\sim B \to C), \sim B\} \vdash A \to C$

② $\{A \to B, A\} \vdash (C \to B)$

③ $\{A \to \sim B, B \to \sim A\} \vdash (A \to \sim B)$

④ $\{A, (A \to B)\} \vdash (B \to C) \to C$

⑤ $\{A, (B \to (A \to C))\} \vdash B \to C$

Deduction Theorem : Given that Σ is a set of hypothesis and α and β are well formed formula. If β is proved from $\{\Sigma \cup \alpha\}$, then $(\alpha \to \beta)$ is proved from Σ.

$$\{\Sigma \cup \alpha\} \vdash \beta \text{ implies } \Sigma \vdash (\alpha \to \beta)$$

Q. Prove $\vdash \sim A \rightarrow (A \rightarrow B)$ by using deduction theorem.

Solution : If we can prove $\{\sim A\} \vdash (A \rightarrow B)$ then using deduction theorem, we have proved $\vdash \sim A \rightarrow (A \rightarrow B)$

Description	formula	Comments
Theorem	$\{\sim A\} \vdash (A \rightarrow B)$	Prove
H1	$\sim A$	1
Instance of A1	$\sim A \rightarrow (\sim B \rightarrow \sim A)$	2
MP (1, 2)	$\sim B \rightarrow \sim A$	3
Instance of A3	$(\sim B \rightarrow \sim A) \rightarrow (A \rightarrow B)$	4
(MP 3, 4)	$A \rightarrow B$	Proved.

Semantic Tableau System in Propositional Logic

Semantic tableau system, we follow backward chaining approach.
In this, a set of rules are applied systematically on a formula or a set of formulae in order to establish consistency or inconsistency.

Semantic tableau is a binary tree which is constructed by using ~~this method are discussed in detail in the~~ of semantic tableau rules with a formula as a root.

: Semantic Tableau Rules :

Rule	Tableau Tree	Explanation
Rule 1	$\alpha \wedge \beta$ is true if both α and β are true $\alpha \wedge \beta$ | α | β	A tableau for formula $(\alpha \wedge \beta)$ is constructed by adding both α and β to the same path.
Rule 2	$\sim(\alpha \wedge \beta)$ is true if either $\sim\alpha$ or $\sim\beta$ is true $\sim(\alpha \wedge \beta)$ / \ $\sim\alpha$ $\sim\beta$	A tableau for the formula $\sim(\alpha \wedge \beta)$ is constructed by adding two new paths. $\sim\alpha$ and $\sim\beta$
Rule 3	$\alpha \vee \beta$ is true if either α or β is true. $\alpha \vee \beta$ / \ α β	$\alpha \vee \beta$ is constructed by adding two new paths α and β.

DATE

Rule 4 $\sim(\alpha \vee \beta)$ is true if both $\sim\alpha$ and $\sim\beta$ are true.

$$\begin{array}{c} \sim(\alpha \vee \beta) \\ | \\ \sim\alpha \\ | \\ \sim\beta \end{array}$$

Rule 5 $\sim(\sim\alpha)$ is true then α is true

$$\begin{array}{c} \sim(\sim\alpha) \\ | \\ \alpha \end{array}$$

Rule 6 $\alpha \rightarrow \beta$ then $\sim\alpha \vee \beta$ is true

$$\begin{array}{ccc} & \alpha \rightarrow \beta & \\ \swarrow & & \searrow \\ \sim\alpha & & \beta \end{array}$$

Rule 7 $\sim(\alpha \rightarrow \beta)$ true then $\alpha \wedge \sim\beta$ is true

$$\begin{array}{c} \sim(\alpha \rightarrow \beta) \\ | \\ \alpha \\ | \\ \sim\beta \end{array}$$

Rule 8 $\alpha \leftrightarrow \beta$ true then $(\alpha \wedge \beta) \vee (\sim\alpha \wedge \sim\beta)$ is true

$$\begin{array}{ccc} & \alpha \leftrightarrow \beta & \\ \swarrow & & \searrow \\ \alpha \wedge \beta & & \sim\alpha \wedge \sim\beta \end{array}$$

Rule 9 $\sim(\alpha \leftrightarrow \beta)$ is true then $(\alpha \wedge \beta) \vee (\sim\alpha \wedge \beta)$ is true

$$\begin{array}{ccc} & \sim(\alpha \leftrightarrow \beta) & \\ \swarrow & & \searrow \\ \alpha \wedge \beta & & \sim\alpha \wedge \beta \end{array}$$

Q. Construct a semantic tableau for a formula
$(A \wedge \sim B) \wedge (\sim B \rightarrow C)$

Description	formula	line number
Tableau Root	$(A \wedge \sim B) \wedge (\sim B \rightarrow C)$	1
Rule 1 (1)	$A \wedge \sim B$	2
	$\sim B \rightarrow C$	3
Rule 1 (2)	A	4
	$\sim B$	5
Rule 6 (3)	$\sim(B)$ C	6
	B ✓ (open)	
	× closed $\{B, \sim B\}$	

Satisfiability & Unsatisfiability

→ A path is said to be contradictory or closed (finished) whenever complementary atoms appear on the same path of semantic tableau.

→ A formula α is said to be satisfiable if a tableau with root α ~~are found to be closed~~ is not a contradictory tableau. (at least one open path)

→ A formula α is said to be unsatisfiable if a tableau with root α is a contradictory tableau.

→ A set of formulae $S = \{\alpha_1, \ldots \alpha_n\}$ is said to be unsatisfiable if a tableau with root $(\alpha_1 \wedge \alpha_2 \wedge \ldots \alpha_n)$ is a contradictory tableau.

→ A set of formulae $S = \{\alpha_1 -- \alpha_n\}$ is said to be satisfiable if the formulae in a set $\alpha_1 \wedge \alpha_2 - \wedge \alpha_n$ has at least one open (or non contradictory path).

I.M.P.

→ A formula is said to be logical consequence of a set S if and only if α is tableau provable from S.

→ The formula α is said to be tableau provable from S ($S \vdash \alpha$) if there is a contradictory tableau from S with $\sim\alpha$ as a root.

Q. Show $\alpha : (A \wedge \sim B) \wedge (\sim B \rightarrow C)$ is satifiable.
(Check solⁿ solved before.)

Q. Show $\alpha : (A \wedge B) \wedge (B \rightarrow \sim A)$ is unsatifiable using the tableau method.

Discription	Formula	Line no.
Tableau Root	$(A \wedge B) \wedge (B \rightarrow \sim A)$	1
Rule (1) 1	$(A \wedge B)$	2
	$(B \rightarrow \sim A)$	3
Rule (1) 2	A	4
	B	5
Rule 6 (3)	$\sim B$ $\sim A$	
	$\{B, \sim B\}$ ✗ Closed $\{A, \sim A\}$ Closed.	

DATE

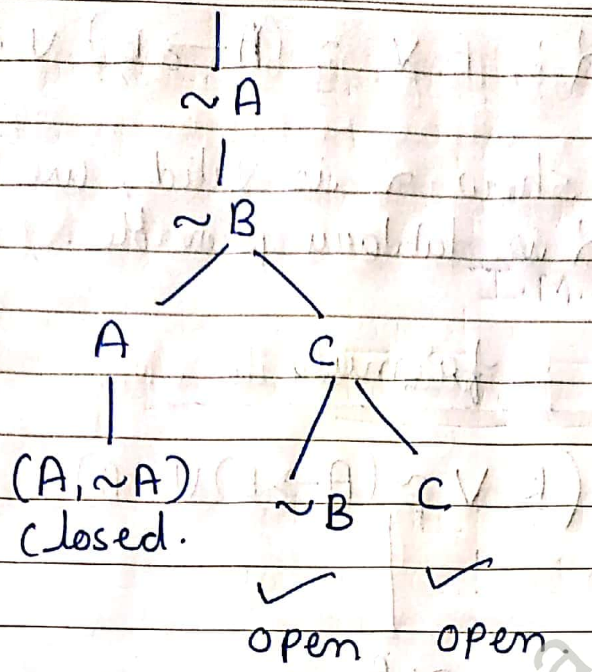
Q. $S = \{\sim(A \vee B), (C \to B), (A \vee C)\}$ show S is unsatisfiable.

Discription	formula	line number
	$\sim(A \vee B) \wedge (C \to B) \wedge (A \vee C)$	1
Rule (1)	$\sim(A \vee B)$	2
	$(C \to B)$	3
	$(A \vee C)$	4
Rule 4(2)	$\sim A$	
	$\sim B$	
Rule 3(4)	$A \quad C$	
Rule 6(3)	$(\sim A, A) \quad \sim C \quad B$	
	X closed $\quad (\sim C, C)$ X closed $\quad (\sim B, B)$ X closed.	

Q. $S = \{\sim(A \vee B), (B \to C), (A \vee C)\}$ is consistent

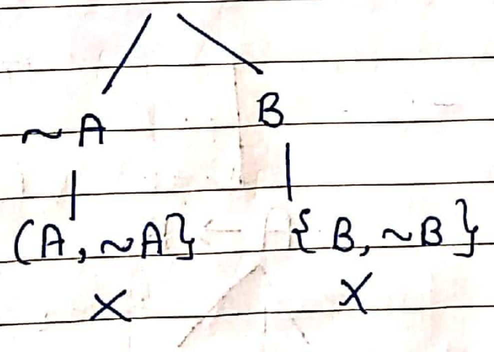
Discription	formula	line number
	$\sim(A \vee B) \wedge (B \to C) \wedge (A \vee C)$	
	$\sim(A \vee B)$	
	$(B \to C)$	
	$(A \vee C)$	

classmate



Q. Show that B is logical consequence of $S = \{A \to B, A\}$

Discription	formula	line number.
Tableau Root	$\sim B$	1
premise 1	$A \to B$	2
Premise 2	A	
		3



~B as root gives contradictory tableau; thus B is logical consequence of S.

Q. Show that α : $B \vee \sim(A \to B) \vee \sim A$ is valid.

In order to show α is valid, we have to show that α is tableau provable.

Discription	formula	Comments.
Tableau Root	$\sim(B \vee \sim(A \to B) \vee \sim A)$	1
Rule 4 (1)	$\sim B$	2
	$\sim[\sim(A \to B) \vee \sim A]$	3
Rule 4 (3)	$\sim[\sim(A \to B)]$	4
	$\sim(\sim A)$	5
	A	6
Rule 5 (5)		
Rule 5 (4)	$A \to B$	7

$$A \to B$$

$\sim A$ B

$\times \{A, \sim A\}$ $\times \{B, \sim B\}$

Q. Show that following formula is valid by giving tableau proof

1. $A \to (B \to A)$
2. $\sim(A \vee B) \leftrightarrow (\sim A \wedge \sim B)$
3. $(\sim A \vee B) \leftrightarrow (A \to B)$

Resolution Refutation in Propositional Logic

DATE

→ Here to prove a formula or derive a goal from a given set of clauses by contradiction, is the resolution refutation method.

→ The term clause is used to denote a special formula containing the boolean operators $\sim$ and $\vee$.

→ Any given formula can be easily converted into a set of clauses.

→ It uses a single inference rule which is known as resolution based on modus ponen inference rule.

→ Resolution produces proofs by refutation. In other words to prove a statement, resolution attempts to show that the negation of the statement prereduces a contradiction with the known statements.

Conversion of a formula into a set of clauses

→ In Propositional logic, there are 2 Normal forms mainly
- Disjunctive NF (DNF)
- Conjunctive NF (CNF).

→ A formula is said to be in its Normal form if it is constructed using only natural connectives $\{\sim, \wedge, \vee\}$

→ In DNF, the formula is represented as disjunction of conjunctions

$$(L_{11} \wedge \dots \wedge L_{1m}) \vee \dots \vee (L_{p1} \wedge \dots \wedge L_{pk})$$

Where as in CNF it is reprented as conjunction of disjunction

$$(L_{11} \vee \dots \vee L_{1m}) \wedge \dots \wedge (L_{p1} \vee \dots \vee L_{pk})$$

where L_{ij} are literals (positive or negative litrals) atoms.

→ So we can write CNF form of a given formula as $(C_1 \wedge .. \wedge C_n)$, where C_K $(1 \leq K \leq n)$ is a disjunction of conjunctions literals and is called a "clause".
So, formally, a clause is $(L_1 \vee .. \vee L_m)$

→ Conversion of a formula to its CNF.

Eliminate double negation signs by using
$\sim(\sim A) \cong A$.

Use de morgan's law to push $(\sim)$ immediately before the atomic formula.

$\sim(A \wedge B) = \sim A \vee \sim B$
$\sim(A \vee B) = \sim A \wedge \sim B$.

Use distributive law to get CNF

$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$

Eliminate $\rightarrow$ and $\leftrightarrow$ by using the following equivalence law

$A \rightarrow B = \sim A \vee B$
$A \leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A)$

Q. Convert the formula $(\sim A \rightarrow B) \wedge (C \wedge \sim A)$ into its equivalent CNF representation.

Soln. $(\sim A \rightarrow B) \wedge (C \wedge \sim A) \cong (\sim(\sim A) \cup B) \wedge (C \wedge \sim A)$
$\cong (A \cup B) \wedge (C \wedge \sim A)$
$\cong (A \cup B) \wedge C \wedge \sim A$

set of clauses
$A \cup B$, C, $\sim A$.

DATE

Resolution of Clauses

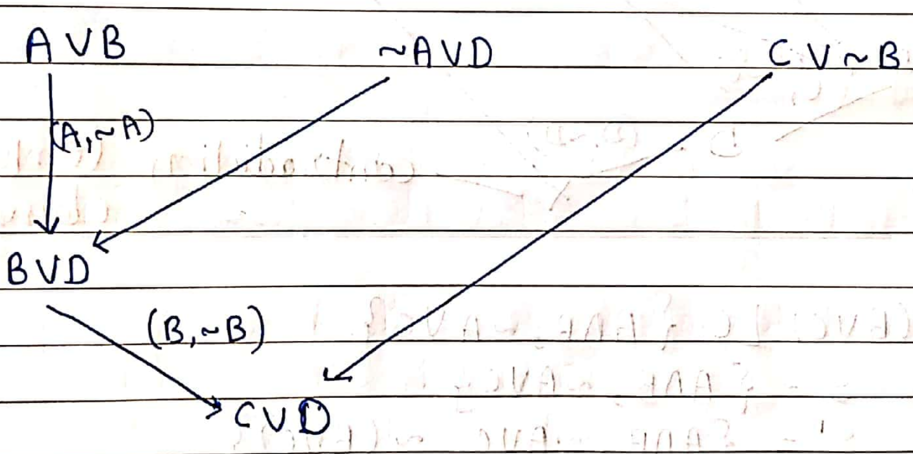
→ Two clauses can be resolved by eliminating complementary pair of litrals

→ If two clauses C_1 & C_2 contain a complementary pair of litrals (L_1, ~L), then these clauses may be resolved together by deleting L from C_1 & ~L from C_2 and constructing a new clause by the disjunction of remaining litrals, in C1 & C2, called "Resolvent"

C1 & C_2 cld parent calause of resolved clause.

The Resolution tree is an inverted Binary tree with the last node being a resolvent.

Q. Find the resolvent clause in the set {A∨B, ~A∨D, C∨~B}



Some Important Results:

→ If C is resolvent of C_1 & C_2 then C is called a logical consequence of the set of the clauses {C_1 & C_2}
This is known as Resolution Principle.

→ If a contradiction (or an empty clause) is derived from a set S of clauses using resolution then S is said Unsatisfiable

→ Derivation of contradiction for a set S by resolution method is called Resolution Refutation (RR) of S.

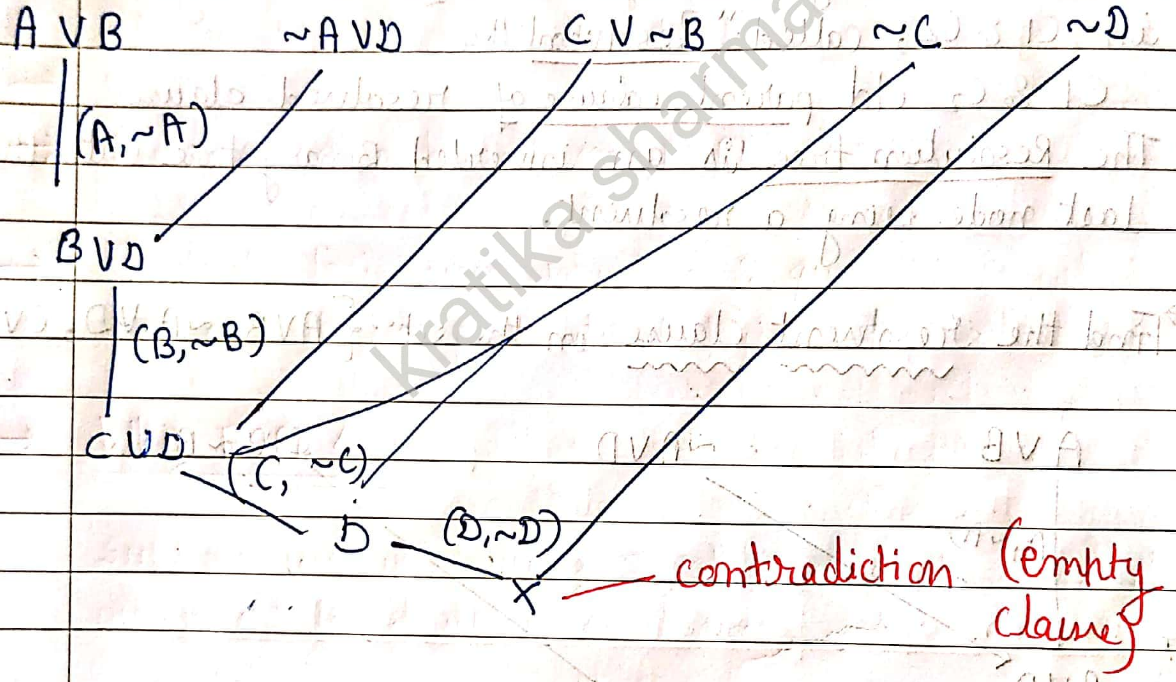
→ Alternatively, using RR concept, C is defined to be a logical consequence of S iff S' = S ∪ {~C} is unsatisfiable. Contradict deduced from S', assuming S is satisfiable.

Q. Using Resolution refutation method following are logical Consequence (LC) of the given set:

$(C \vee D)$ LC $\{A \vee B, \sim A \vee D, C \vee \sim B\}$

① $S = \{A \vee B, \sim A \vee D, C \vee \sim B\}$

To prove the statement, first we will add negation of the LC, $\sim(C \vee D) = \sim C \wedge \sim D$ to the sets S to get $S' = \{A \vee B, \sim A \vee D, C \vee \sim B, \sim C, \sim D\}$

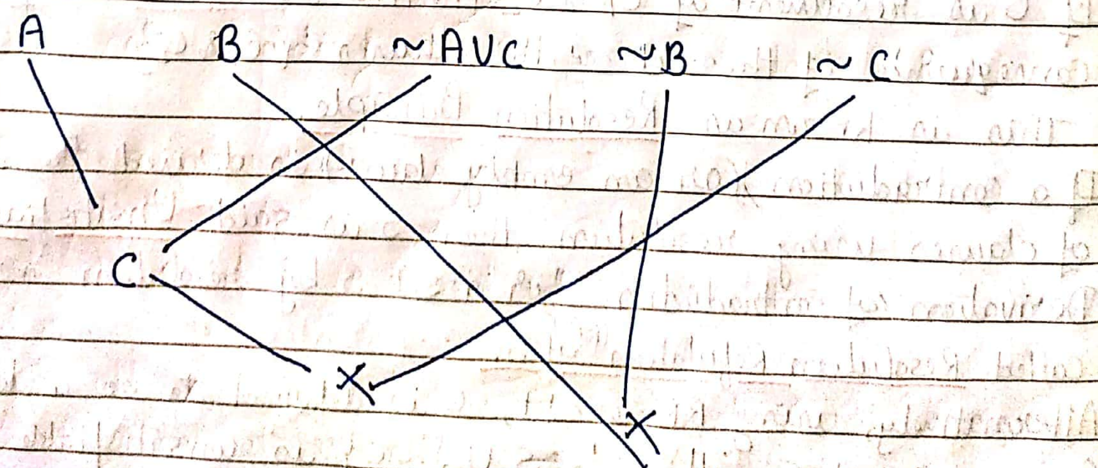


② $(B \vee C)$ LC $\{A \wedge B, \sim A \vee C\}$

$S = \{A \wedge B, \sim A \vee C\}$

$S' = \{A \wedge B, \sim A \vee C, \sim(B \vee C)\}$

$S' = \{A, B, \sim A \vee C, \sim B, \sim C\}$



classmate

Scanned by CamScanner

(3) $C \rightarrow A$ LC $\{(B \cap C) \rightarrow A, B\}$

Solution :

→ LC = $C \rightarrow A$

first remove $(\rightarrow)$ from LC.

$\sim C \cup A$

add negation

$\sim(\sim C \vee A) = \sim(\sim C) \cap \sim A$.

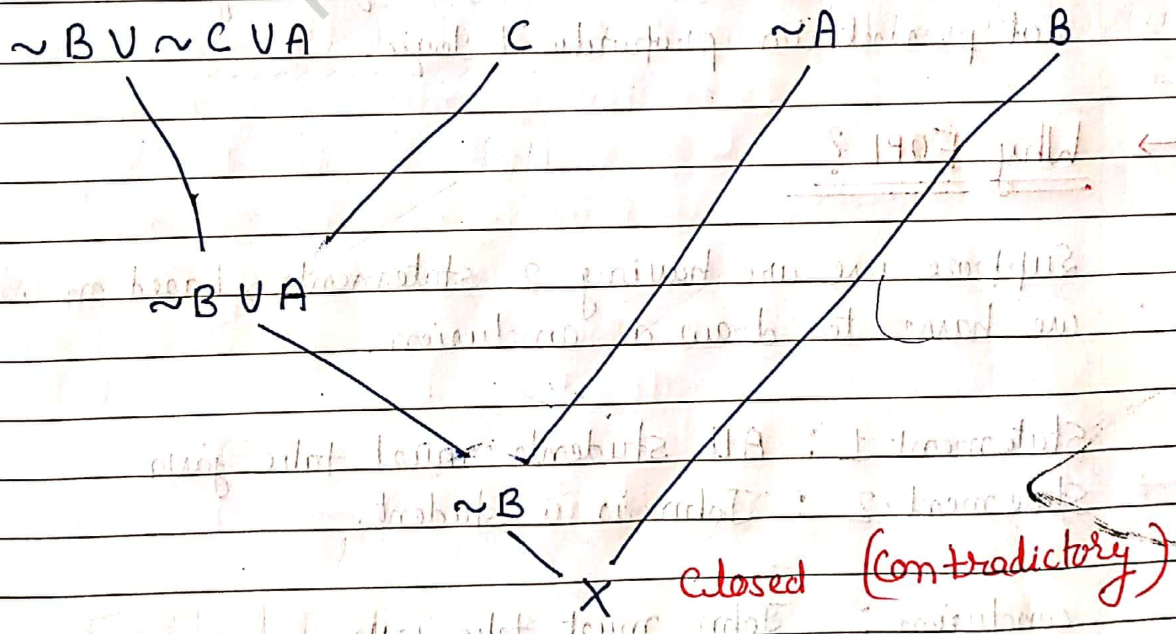
$C \cap \sim A$.

→ Remove $(\rightarrow)$ from the set of litral

$(B \cap C) \rightarrow A$

$\sim(B \cap C) \cup A$

$\sim B \cup \sim C \cup A$



(4) $(\sim U \cap S)$ LC $\{A \vee C, \sim C \rightarrow B, \sim B, A \rightarrow S, \sim U\}$

(5) $(A \vee \sim B)$ LC $\{A \vee C, \sim B \vee \sim C\}$

(6) $(A \vee C)$ LC $\{A, B \rightarrow C, B\}$

Predicate Logic

First Order Predicate logic (FOPL)

→ It was developed by logicians to extend the expressiveness of propositional logic.

→ It is generalization of propositional logic that permits reasoning about world entities (objects) as well as classes & subclasses of objects.

→ Prolog is also based on FOPL.

→ Predicate logic uses variables & quantifiers which is not present in propositional logic.

→ Why FOPL?

Suppose we are having 2 statements, based on which we have to draw a conclusion.

Statement 1 : All students must take java
Statement 2 : John is a student.

conclusion : John must take java.

but not according to Propositional logic (PL) Disadvantage

→ 0 is a natural number.
Natural (0)

→ for all x, if x is a natural number, then so is successor of x
For all x, natural (x) → natural (successor (x))

→ 2 is a natural number
Natural (2)

→ -1 is a natural number
Natural (-1)

DATE

First Order predicate calculus

FOPC classifies the different parts of such statements as follows:

1. Constants
2. Predicates
3. functions
4. Variable.
5. Connectives.
6. Quantifiers
7. punctuation symbols.

Predicates

→ These are names for functions that are true or false, like Boolean functions in a program.

→ Defined as a relation that binds 2 atoms together.

Eg: Amit likes sweets
likes (amit, sweets)

predicate. atoms

→ Predicate can take no. of arguments.

→ Its always possible to have a function as an argument.

Eg: Ravi's father is Rani's father
father (father (Ravi), rani)

→ Constants. or (atoms) usually numbers or names.
they can not broken down into sub parts.

Ex. natural (0)
↳ constant

→ Variables
stands for quantities that are yet unspecified.
Eg: valuable (x)
variable

→ Function :
predicates - true or false and all others are function which represent non boolean value.

→ Quantifiers

Declares the scope or range of variables in a logical expression.

1. Universal Quantifier ($\forall$) for all, for each, f
2. Existential Quantifier ($\exists$) there exist.
   ↳ It is used to state that a predicate is true of at least one thing in the universe, indicated by variable x.
   (there exists a b, for some b, for atleast one b)

→ Precedence of Connectives

~ highest
&
V
→
↔ lowest.

What is prolog (logic programming)

→ It is declarative language, used to solve problems that involves objects and relationships b/w objects.

→ A prolog program can also be seen as a relational ~~bat~~ database containing rules as well as facts.

structure of logic programs:

- program consists of procedures.
- procedures consist of clauses.
- clauses are statements about what is true about the problem.
- Each clause is a fact or a rule.
- programs are executed by posing queries.
- Prolog Consist of:
  - Declaring some facts about objects and their relationship
  - Declaring some rules about.
  - Asking questions (goals) about.

Example:

Predicate

Procedure for elephant

Fact.

clauses

elephant (george)
elephant (Mary)
elephant (X) ← grey (X), mammal (X), hasTrunk (X)

Rule.

Queries
? elephant (george)
yes

Replies.
? elephant (jane)
no

→ **Why Rules?**

Rules is an extension of fact that with added conditions that have to be specified for it to be true.

Eg: John like all people who likes sweets.

Way 1: Write down separate facts as below:

likes (james, sweets)
likes (john, james)
likes (john, mary
⋮
so on.

Way 2:

likes (john, X) ← likes (X, sweets)

Head   if   Body

Eg: To write a rule, X is a sister of Y.
X is a female
X has mother M and father F
Y has the same M & F as X does.

sister_of (X, Y) ← female (X), parents (X, M, F),
parents (Y, M, F)

Eg: X is a grandfather of Y, if X is a father of Z and Z is a parent of Y.

grandfather (X,Y) ← father (X,Z) , parent (Z,Y)

X is a sibling of Y if they both have the same parent

Sibling (X,Y) ← parent (Z,X), parent (Z,Y).

Types of Queries:

1. Ground Query : goal contains constant (T or F)
2. Non Ground Query : goal should have at least one variable as an argument.

Eg. G.Q : Is raman a grandfather of manu?
?- grandfather (raman, manu)

NGQ : "Does there exist X such that X is a father of manu" (who is father of manu)

? father (X, manu)

Conjunctive queries.

?- father (raman, robert), father (robert, mike)

Inference Rules:

ways of deriving or proving new statements from a given set of statements.

Eg:
$a \to b, b \to c$
$a \to c$

Horn clauses: (inventor Alfred Horn)

a_1 and a_2 and a_3 - - - and $a_4 \to b$

a_i should be simple stmts with no connectives as well as no quantifiers.

b = head of the clause
a - - a_n = body of the clause.

a_i's may be zero, in this case horn clause:

$\to b$

means b is always true. (Facts)

Eg:
x is a grandparent of y if x is the parent of someone who is parent of y

Predicate calculus: grandparent $(X, Y) \leftarrow \forall X, \forall Y (\exists Z$ parent $(X, Z) \wedge$ parent $(Z, Y))$

Horn clause: grand parent $(X, Y) \leftarrow$ parent (X, Z) and parent (Z, Y).

DATE

for all x, if x is a mammal then x has 2 or 4 legs.

Resolution:

Resolution says that if we have 2 horn clauses, and we can match the head of the first HC with one of the statement in the body of second HC, then the first clause can be used to replace its head in second HC. by its body.

$a \leftarrow a_1 \ldots a_n$
$b \leftarrow b_1 \ldots b_m$

b_i matches a then

$b \leftarrow b_1 \ldots b_{i-1}, a_1 \ldots a_n, b_{i+1} \ldots b_m$.