Unit-2 (Part-2)

Logic Concepts

Axiomatic system

- Axiomatic system is based on 3 axioms and one rule of deduction called Modes Ponen(MP).
- Proofs are difficult and need a guess in selection of appropriate axioms.
- Although minimal in structure, it is Powerful as Truth table and NDS approaches.
- In this system, only two logical operators ~(not) & →(implies) are used to from a formula.

<u>Note:</u>

- Other logical operators can be easily expressed in terms of ~ and → using equivalence laws.

For example:

$$A \wedge B \equiv \sim(\sim A \vee \sim B) \equiv \sim(A \rightarrow \sim B)$$

$$A \vee B \equiv \sim A \rightarrow B$$

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

Axiom & Rule

- Axiom 1: $\alpha \rightarrow (\alpha \rightarrow \beta)$
- 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 Modus Ponen Rule: Given that $\alpha \rightarrow \beta$ and α are hypotheses (assumed to be true), β is inferred (i.e true) as a consequent.

Axiomatic System- Example

1. Establish that $A \rightarrow C$ is a deductive consequence of $\{A \rightarrow B, B \rightarrow C\}$ i.e $\{A \rightarrow B, B \rightarrow C\} \mid\!- (A \rightarrow C)$.

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

Theorem

Deduction Theorem: Given that $\sum$ is a set of hypotheses and α and β are WFF. If β is proved from $\{\sum \cup \alpha\}$, then according to the deduction theorem, $(\alpha \rightarrow \beta)$ is proved from $\sum$. Alternatively, we can write $\{ \sum \cup \alpha \} |\!- \beta$ implies $\sum |\!-(\alpha \rightarrow \beta)$.

Converse of Deduction Theorem : The converse of the deduction theorem can be stated as : Given $\sum |\!- (\alpha \rightarrow \beta)$, then $\{ \sum \cup \alpha \} |\!- \beta$ is proved.

Useful tips

- If α is given, then we can easily prove $\beta \rightarrow \alpha$ for any well-formed formulae α and β.

- If $\alpha \rightarrow \beta$ is to be proved, then include α in the set of hypotheses $\sum$ and derive β from the set $\{ \sum \cup \alpha \}$. Then by using deduction theorem, we can conclude that $\alpha \rightarrow \beta$.

Axiomatic System Example 1:

Example 4.6 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 $\vdash \sim A \rightarrow (A \rightarrow B)$. The proof is shown in Table 4.10.

Table 4.10 Proof of $\{\sim A\} \vdash (A \rightarrow B)$

Description	Formula	Comments
Theorem	$\{\sim A\} \vdash (A \rightarrow B)$	*Prove*
Hypothesis 1	$\sim A$	1
Instance of Axiom 1	$\sim A \rightarrow (\sim B \rightarrow \sim A)$	2
MP (1, 2)	$(\sim B \rightarrow \sim A)$	3
Instance of Axiom 3	$(\sim B \rightarrow \sim A) \rightarrow (A \rightarrow B)$	4
MP (3, 4)	$(A \rightarrow B)$	*Proved*

Semantic Tableau System in Propositional Logic

- In NDS and axiomatic systems, forward chaining approach is used for constructing proofs and derivations.
- Axiomatic system, often requires a guess for the selection of appropriate axiom(s) to prove the theorem.
- Implementation and derivation of forward chaining approach is difficult.

- Two other approaches are:
  - ✔ Semantic tableau method
  - ✔ Resolution refutation method
- In the above two approaches, proofs follow backward chaining approach.

Semantic Tableau System in Propositional Logic

- Semantic tableau method, given set of rules will be applied systematically on a formula or a set of formulae in order to establish consistency or inconsistency.

- *Semantic tableau* is a binary tree that can be constructed by using semantic tableau rules with a formula as a root.

Semantic Tableau System in Propositional Logic

Table. Semantic Tableau rules for α_j and β

Rule No.	Tableau tree	
Rule 1	*$\alpha \wedge \beta$ is true if both α and β are true* $\alpha \wedge \beta$ | α | β	A tableau for a formula $(\alpha \wedge \beta)$ is constructed by adding both α and β to the same path (branch)
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 a formula $\sim(\alpha \wedge \beta)$ is constructed by adding two new paths: one containing $\sim\alpha$ and the other containing $\sim\beta$
Rule 3	*$\alpha \vee \beta$ is true if either α or β is true* $\alpha \vee \beta$ α β	A tableau for a formula $(\alpha \vee \beta)$ is constructed by adding two new paths: one containing α and the other containing β
Rule 4	*$\sim(\alpha \vee \beta)$ is true if both $\sim\alpha$ and $\sim\beta$ are true* $\sim(\alpha \vee \beta)$ | $\sim\alpha$ | $\sim\beta$	A tableau for a formula $\sim(\alpha \vee \beta)$ is constructed by adding both $\sim\alpha$ and $\sim\beta$ to the same path
Rule 5	*$\sim(\sim\alpha)$ is true then α is true* $\sim(\sim\alpha)$ | α	A tableau for $\sim(\sim\alpha)$ is constructed by adding α on the same path

Table. Semantic Tableau rules for α_j and β (Cntd..)

	α	
Rule 6	$\alpha \rightarrow \beta$ *is true then* $\sim\alpha \vee \beta$ *is true* $\alpha \rightarrow \beta$ $\sim\alpha \quad\quad \beta$	A tableau for a formula $\alpha \rightarrow \beta$ is constructed by adding two new paths: one containing $\sim\alpha$ and the other containing β
Rule 7	$\sim(\alpha \rightarrow \beta)$ *true then* $\alpha \wedge \sim\beta$ *is true* $\sim(\alpha \rightarrow \beta)$ $\sim(\sim\alpha \vee \beta)$ α $\alpha \wedge \sim\beta$ $\sim\beta$	A tableau for a formula $\sim(\alpha \rightarrow \beta)$ is constructed by adding both α and $\sim\beta$ to the same path

(Contd.)

Table. Semantic Tableau rules for α_j and β (Cntd..)

Rule No.	Tableau tree	Explanation
Rule 8	$\alpha \leftrightarrow \beta$ *is true then* $(\alpha \wedge \beta) \vee (\sim\alpha \wedge \sim\beta)$ *is true* $\alpha \leftrightarrow \beta$ $\alpha \wedge \beta$ $\quad$ $\sim\alpha \wedge \sim\beta$	A tableau for a formula $\alpha \leftrightarrow \beta$ is constructed by adding two new paths: one containing $\alpha \wedge \beta$ and other $\sim\alpha \wedge \sim\beta$ which are further expanded
Rule 9	$\sim(\alpha \leftrightarrow \beta)$ is *true* then $(\alpha \wedge \sim\beta) \vee (\sim\alpha \wedge \beta)$ *is true* $\sim(\alpha \leftrightarrow \beta)$ $\alpha \wedge \sim\beta$ $\quad$ $\sim\alpha \wedge \beta$	A tableau for a formula $\sim(\alpha \leftrightarrow \beta)$ is constructed by adding two new paths: one containing $\alpha \wedge \sim\beta$ and the other $\sim\alpha \wedge \beta$ which are further expanded

Example 1:

Example 4.7 Construct a semantic tableau for a formula $(A \land \sim B) \land (\sim B \to C)$.

Solution The construction of the semantic tableau for the given formula $(A \land \sim B) \land (\sim B \to C)$ is shown in Table 4.12.

Table 4.12 Semantic Tableau for Example 4.7

Description	Formula	Line number
Tableau root	$(A \land \sim B) \land (\sim B \to C)$	1
Rule 1 (1)	$A \land \sim B$	2
	$\sim B \to C$	3
Rule 1 (2)	A	4
	$\sim B$	5
Rule 6 (3)	$\sim(\sim B)$ C	6
Rule 3 (6)	B √ (open)	
	× (closed) $\{B, \sim B\}$	

Satisfiability and Unsatisfiability

- A path is said to be contradictory or closed (finished) whenever complementary atoms appear on the same path of a semantic tableau. This denotes inconsistency.
- If all paths of a tableau for a given formula α are found to be closed, it is called a contradictory tableau. This indicates that there is no interpretation or model that satisfies α.
- A formula α is said to be satisfiable if a tableau with root α is not a contradictory tableau, that is, it has atleast one open path. We can obtain a model or an interpretation for the formula α and is evaluated to be true by assigning T(true) to all atomic formulae appearing on the open path of semantic tableau of α.
- A formula α is said to be unsatisfiable if a tableau with root α is a contradictory tableau.
- If we obtain a contradictory tableau with root ~ α, we say that the formula α is tableau provable. Alternatively, a formula α is said to be tableau provable (denoted by |- α) if a tableau with root ~ α is a contradictory tableau.

Satisfiability and unsatisfiability

- A set of formulae $S = \{\alpha_1, \alpha_2, \ldots \alpha_n\}$ is said to unsatisfiable if a tableau with root $\{\alpha_1 \wedge \alpha_2, \ldots \wedge \alpha_n\}$ is a contradictory tableau.
- A set of formulae $S=\{\alpha_1, \alpha_2, \ldots \alpha_n\}$ is said to satisfiable if the formulae in a set are simultaneously true, that is if a tableau for $\alpha_1 \wedge \alpha_2, \ldots \wedge \alpha_n$ has atleast one open (non-contradictory) path.
- Let S be a set of formulae. The formulae α is said to be tableau provable from S (denoted by $S \vdash \alpha$) if there is a contradictory tableau from S with $\sim \alpha$ as a root.
- A formula α is said to be a logical consequence of a set S if and only if α is tableau provable from S.

Example 1:

Example 4.9 Show that $\alpha : (A \land B) \land (B \rightarrow \sim A)$ is unsatisfiable using the tableau method.

Solution It can be proven that $\alpha : (A \land B) \land (B \rightarrow \sim A)$ is unsatisfiable as shown in Table 4.13.

Table 4.13 Tableau Method for Example 4.9

Description	Formula	Line number
Tableau root	$(A \land B) \land (B \rightarrow \sim A)$	1
Rule 1 (1)	$A \land B$	2
	$B \rightarrow \sim A$	3
Rule 1 (2)	A	4
	B	5
Rule 6 (3)	$\sim B$ $\sim A$	
	$\times \{B, \sim B\}$ $\times \{A, \sim A\}$	

Example 2:

Example 4.11 Show that a set $S = \{\sim(A \vee B), (B \to C), (A \vee C)\}$ is consistent.

Solution The set S can be shown to be consistent (Table 4.15) as follows:

Table 4.15 Tableau Method for Example 4.11

Description	Formula	Line number
Tableau root	$\sim(A \vee B) \wedge (B \to C) \wedge (A \vee C)$	1
Rule 1 (1)	$\sim(A \vee B)$	2
	$(B \to C)$	3
	$(A \vee C)$	4
Rule 4 (2)	$\sim A$	
	$\sim B$	
Rule 3 (4)	A C	
Rule 6 (3)	$\times \{A, \sim A\}$ $\sim B$ C	
	$\surd$ $\surd$	

Example 3:

Example 4.12 Show that B is a logical consequence of $S = \{A \rightarrow B, A\}$.

Solution Let us include $\sim B$ as a root with S in the tableau tree.

Table 4.16 Tableau Method for Example 4.12

Description	Formula	Line number
Tableau root	$\sim B$	1
Premise 1	$A \rightarrow B$	2
Premise 2	A	
Rule 6 (2)	$\sim A \qquad B$	3
	$\times \{A, \sim A\} \qquad \times \{B, \sim B\}$	

We see from Table 4.16 that B is tableau provable from S, that is, $\sim B$ as root gives contradictory tableau; thus B is a logical consequence of S.

Resolution Refutation in Propositional logic

- It is most favoured method for developing computer-based systems that can be used to prove theorems automatically.

- This is another method that can be used in propositional logic to prove a formula or derive a goal from a given set of clauses by contradiction.

- The term clause is used to denote a special formula containing the Boolean operators ~ and V.

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

Resolution Refutation in Propositional logic

- It uses single inference rule, known as resolution based on modus ponen inference rule.

- It is more efficient compared to NDS and Axiomatic system.

- During resolution, two clauses need to be identified for the application of resolution rule:
  - one with a positive atom (P) and
  - the other with a negative atom (~P)

Conversion of Formula into Set of Clauses

- In propositional logic, there are two normal forms:
  - ✔ Disjunctive Normal Form (DNF)
  - ✔ Conjunctive Normal Form (CNF)

- A formula is said to be in normal form, if it is constrcuted using only natural connectivities {~, ^, V}.

- Disjunctive Normal Form : The formula is represented as disjunction of conjunction, that is, in the form $(L_{11} \wedge \ldots\ldots\ldots \wedge L_{1m}) \vee \ldots\ldots\ldots \vee (L_{p1} \wedge \ldots\ldots\ldots \wedge L_{pk})$

- Conjunctive Normal Form : The formula is represented as conjunction of disjunction, that is, in the form $(L_{11} \vee \ldots\ldots\ldots \vee L_{1m}) \wedge \ldots\ldots\ldots \wedge (L_{p1} \vee \ldots\ldots\ldots \vee L_{pk})$.

Conversion of Formula to its CNF

- Eliminating double negation signs by using

  $\sim(\sim A) \equiv A$

- Use De Morgan's Laws to push ~(negation) immediately before the atomic formula.

  $\sim(A \wedge B) \equiv \sim A \vee \sim B$

  $\sim(A \vee B) \equiv \sim A \wedge \sim B$

- Use distributive law to get CNF

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

- Eliminate $\rightarrow$ and $\leftrightarrow$ by using the following equivalence laws

  $A \rightarrow B \equiv \sim A \vee B$

  $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$

Example for Conversion of a formula to its CNF:

The method of conversion will become clearer with the help of the following examples:

Example 4.14 Convert the formula $(\sim A \rightarrow B) \wedge (C \wedge \sim A)$ into its equivalent CNF representation.

Solution The given formula $(\sim A \rightarrow B) \wedge (C \wedge \sim A)$ can be transformed into its CNF representation in the following manner:

$$
\begin{aligned}
(\sim A \rightarrow B) \wedge (C \wedge \sim A) &\equiv (\sim(\sim A) \vee B) \wedge (C \wedge \sim A) && \{\text{as } \sim A \rightarrow B \equiv (\sim A) \vee B\} \\
&\equiv (A \vee B) \wedge (C \wedge \sim A) && \{\text{as } \sim(\sim A) \equiv A\} \\
&\equiv \underset{1}{(A \vee B)} \wedge \underset{2}{C} \wedge \underset{3}{\sim A}
\end{aligned}
$$

The set of clauses in this case is written as $\{(A \vee B), C, \sim A\}$

Resolution of Clauses

- Two clauses C_1 and C_2, can be resolved by eliminating complementary pair of literals if any from both.
- A new clause is constructed by disjunction of the remaining literals in both the clauses.
- The clause C_1 and C_2 are called parent clause and the new clauses are called resolvent of C_1 and C_2.
- The resolution tree is an inverted binary tree with the last node being a resolvent, which is generated as a part of the resolution process.

Resolution of Clauses

Example 4.15 Find resolvent of the clauses in the set $\{A \vee B, \sim A \vee D, C \vee \sim B\}$.

Solution The method of resolution is shown in Fig. 4.1.

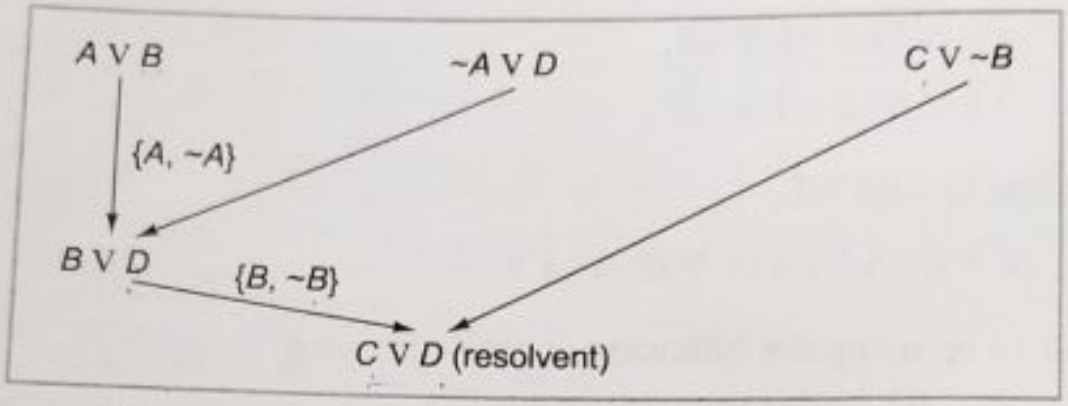

Figure 4.1 Resolution of clauses of Example 4.15

We can clearly see that $C \vee D$ is a resolvent of the set $\{A \vee B, \sim A \vee D, C \vee \sim B\}$.

Resolution of Clauses

- If C is a resolvent of two clauses C_1 and 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 of clauses using resolution, then S is said to be satisfiable. Derivation of contradiction for a set S by resolution method is called a resolution refutation S.
- A clause C is said to be a logical consequence of S if C is derived from S.
- Alternatively, using the resolution refutation concept, a clause C is defined to be a logical consequence of S if and only if the set S' = S U {~C} is unsatisfiable, that is a contradiction(or an empty clause) is deduced from the set S', assuming that initially the set S is satisfied.

Resolution of Clauses

Example 4.16 Using resolution refutation principle show that $C \vee D$ is a logical consequence of $S = \{A \vee B, \sim A \vee D, C \vee \sim B\}$.

$\sim(C \vee D) \rightarrow \sim C \wedge \sim D$

Solution To prove the statement, first we will add negation of the logical consequence, that is, $\sim(C \vee D) \equiv \sim C \wedge \sim D$ to the set S to get $S' = \{A \vee B, \sim A \vee D, C \vee \sim B, \sim C, \sim D\}$. Now, we can show that S' is unsatisfiable by deriving contradiction using the resolution principle (Fig. 4.2).

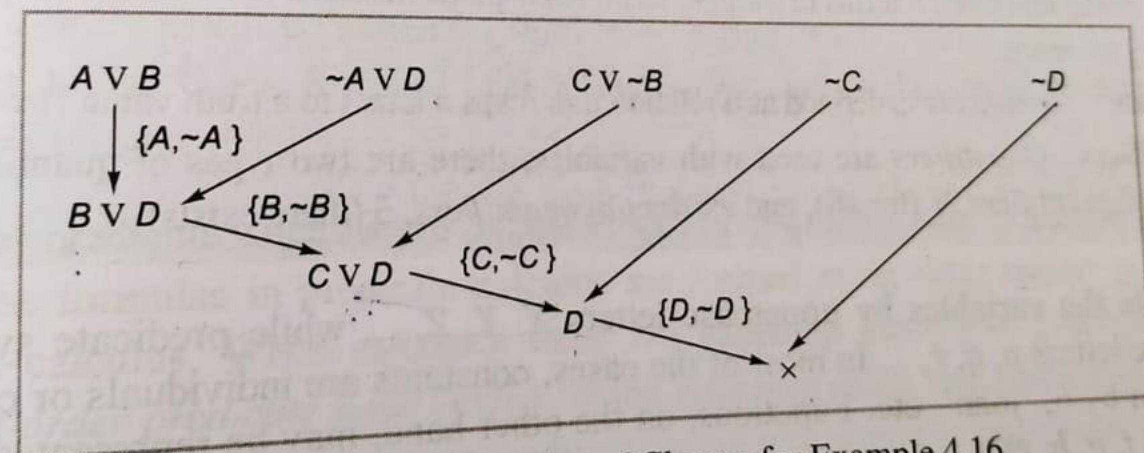

Figure 4.2 Resolution of Clauses for Example 4.16

Since we get contradiction from S', we can conclude that $(C \vee D)$ is a logical consequence of $S = \{A \vee B, \sim A \vee D, C \vee \sim B\}$.

Queries ?