CM 1606 Computational Mathematics

Propositional Logic

Week 5 | Ganesha Thondilege

ROBERT GORDON UNIVERSITY ABERDEEN
TEF Gold
INFORMATICS INSTITUTE OF TECHNOLOGY

Learning Outcomes

- Covers LO1 for Module
- On completion of this lecture, students are expected to be able to:
  - Identify valid formulas (Satisfiable and unsatisfiable)
  - Identify and represent an argument(Knowledge base) as a logical formula
  - Check the validity of an argument(KB) using truth values
  - Discuss the logical consequence and the validity of an argument

CONTENT

- Well formed formulae
- Satisfiable and unsatisfiable formulae
- Valid formulae
- Knowledge bases (Arguments)
- Validity and Invalidity of an argument
- Reasoning and logical consequence
- Proving validity by contradiction
- Tableau Technique

Well formed formulae

- Any primitive is a well-formed formulae(wff)
- If p and q are wff then any formulae in the form of

$$\sim p, p \wedge q, p \vee q, p \rightarrow q, p \leftrightarrow q$$ are wff.

$$\text{If A is a wff} \leftrightarrow \text{Enable for reasoning properly}$$

Satisfiable and unsatisfiable

- A compound **proposition** P is **satisfiable** if there is a truth assignment that satisfies P
- A compound **proposition** P is **unsatisfiable** (or a contradiction) if it is not **satisfiable**

Eg: i) $P \rightarrow (Q \rightarrow \neg P)$

ii) $P \wedge \neg Q$

iii) $(P \vee \neg P) \rightarrow Q$

iv) $(P \vee Q) \wedge (\neg P \wedge \neg Q)$

Valid formulae

- Formulae B is valid if there are only truth assignments that satisfies B
- Tautologies are valid formulas
- Contradictions are invalid
- $\models$ A notation is used to denote, A is valid

Eg:

i)P $\vee$ ¬P

ii) P → (Q → P)

Arguments

Consider the following

"If John drinks beer, he is at least 18 years old.

John does not drink beer. Therefore, John

is not yet 18 years old "

- An example for an argument

Arguments ctd.

- An argument is a sequence of propositions

$$A_1, A_2, \ldots\ldots\ldots A_n$$

- The last proposition is separated by the symbols $\therefore$ or $\vdash$ (Therefore)

$$A_1, A_2, \ldots\ldots\ldots, A_{n-1} \therefore A_n \quad \text{or}$$

$$A_1, A_2, \ldots\ldots\ldots, A_{n-1} \vdash A_n$$

- The last proposition is called as the conclusion

$$A_n \quad \text{is the conclusion}$$

- All propositions except the conclusion are called as premises (Assumptions or the knowledge base)

$$A_1, A_2, \ldots\ldots\ldots A_{n-1} \quad \text{are premises or the KB}$$

Arguments ctd.

- “If John drinks beer, he is at least 18 years old. John does not drink beer. Therefore, John is not yet 18 years old “

Let p - John drinks beer

q - John is at least 18 years old, then

$$p \rightarrow q, \sim p \text{ Therefore } \sim q$$

$$or$$

Conclusion

Premises or the KB

$$p \rightarrow q, \sim p \;\therefore\; \sim q$$

Validity of an argument

An argument is said to be valid,

if the conclusion is true whenever all the premises are true

An argument which is not valid (invalid)

is called a fallacy

Example

- p - John drinks beer

  q - John is at least 18 years old, then consider the same example

$$p \rightarrow q, \sim p \quad \therefore \quad \sim q$$

p	q	p → q	~p	~q
T	T	T	F	F
T	F	F	F	T
F	T	T	T	F
F	F	T	T	T

$\Rightarrow$ Invalid(Fallacy)

Example

- Consider the argument

$$p \to q, p \quad \therefore \quad q$$

p	q	p → q	p
T	T	T	T
T	F	F	T
F	T	T	F
F	F	T	F

$\Rightarrow$ Valid

Reasoning and Logical consequence

B is a logical consequence of a knowledgebase $A_1, A_2, A_3, \ldots\ldots, A_n$, if the logical formula $(A_1 \wedge (A_2 \wedge (A_3 \wedge \ldots\ldots A_n))) \to B$ is Valid (True always).

If so, the KB is logically consequence to B

Denoted as

$$A_1, A_2, A_3, \ldots A_n \models B$$

Example

$$p \to q, p \quad \therefore \quad q$$

p	q	p → q	p	(p → q) ∧ p	((p → q) ∧ p)→q
T	T	T	T	T	T
T	F	F	T	F	T
F	T	T	F	F	T
F	F	T	F	F	T

Examples

1) Identify the invalid arguments out of the following arguments

$$i)\, p \wedge q \therefore q$$

$$ii)\, p \vee \sim q,\, p \therefore \sim q$$

$$iii)\, p \rightarrow q, q \therefore \sim p$$

$$iv)\, p \rightarrow q,\, p \therefore q$$

$$v)\, p \rightarrow q, \sim q \therefore p$$

Proving validity by contradiction

Ex: Prove that $((P \land Q) \rightarrow P)$ is valid without constructing the full truth table

1. Assume that - $((P \land Q) \rightarrow P)$ is invalid.

2. This may happen when $(P \land Q)$ is True and P is False only.

3. Again $(P \land Q)$ can be Ture if both P and Q are True only. But we have P as False according to the 2nd step.(Finally we have three results, True P, True Q, and False P)

4. Since this is a contradiction, Assumption in 1st is wrong. So $((P \land Q) \rightarrow P)$ is valid

Tableau Technique

- Notation

  For a well-formed formulae A

  T[A] – A is True

  F[A] – A is False

α Formulae

The unique property of each logical connective is considered.

α	α_1 and α_2
$T[A \wedge B]$	$T[A]$ and $T[B]$
$F[A \vee B]$	$F[A]$ and $\mathrm{F}[B]$
$F[A \rightarrow B]$	$T[A]$ and $\mathrm{F}[B]$
$T[\sim A]$	$F[A]$
$F[\sim A]$	$T[A]$

β Formulae

Not unique.

'Or', Branching

β	β_1 or β_2
$T[A \vee B]$	$T[A]$ or $T[B]$
$F[A \wedge B]$	$F[A]$ or $F[B]$
$T[A \rightarrow B]$	$F[A]$ or $T[B]$

Construction of a Tableau

- **Definition** (Configuration in a tableau) Sets of signed formulae are called *configurations*.
- Rule is applied to a signed formula in the configuration above the horizontal line and the rule’s conclusion is a configuration(s) below the horizontal line.
- Using α rules - unique conditions - transform to new configuration
- Using β rules - branching conditions - transform to new configurations reflecting branches

Algorithm

Let's consider building tableau for the formulae A

Step 1: Label the initial node as $F[A]$ (e.g.: assume that A is false).

Step 2: The α and β expansion rules are applied to the formulae within labels of nodes of the graph.

Step 3.1: If an expansion rule applies to α-formula in a label of a node n_i then create a new node, n_{i+1}, the successor of n_i, and put both conclusions of the rule into the label of n_{i+1}.

Algorithm

- Step 3.2: If an expansion rule applies to β-formula in a label of a node n_i then create two nodes $n_{i.1}$ and $n_{i.2}$, the children of n_i, and put the conclusions, β1 and β2 (of the rule being applied) into $n_{i.1}$ and $n_{i.2}$, respectively.
- Step 4: Apply 3.1 and 3.2 until no expansion rule to a configuration – label of a node - is applicable; such a configuration is called completed.
- Step 5: If you get a contradiction for the label of a node as T[B] and F[B], for some formula B. Such a configuration is called closed.

Reduced Graph

- Step 6:

Delnode.1 Delete every node if is labelled by a closed configuration

Delnode.2 If all the successors of a node have been deleted then delete this node.

  - Reduced graph G' is empty if the initial node of the original graph is deleted.

- Step 7: A tableau is called closed if its reduced graph is empty.

Validity

- Statement 1. For any formula G, a tableau is closed, if and only if, G is unsatisfiable.
- Statement 2 [correctness of tableau for a formulae] A tableau constructed for the assumption F[A] is closed if, and only if A is valid.

Example

Construct the tableau and check the validity

- $(P \land Q) \rightarrow P$
- $((P \rightarrow Q) \rightarrow P) \rightarrow P$
- $((P \rightarrow Q) \land P) \rightarrow Q$
- $(P \lor Q) \land (\neg Q \land \neg P)$