Predicate Logic Conti….

Professor Dr. A K M Akhtar Hossain
Dept. of CSE, RU.

Properties of Statements:

- **Satisfiable:** A statement is satisfiable if there is some interpretation for **which it is true**.
- **Contradiction:** A sentence is contradictory (unsatisfiable) if there is **no interpretation** for **which, it is true**.
- **Valid:** A sentence is valid **if it is true for every interpretation**.
- **Equivalence:** Two sentences are equivalent if they have the **same truth value** under every interpretations.

Conti…

- **Logical Consequences:** A sentence is a logical consequence of another if it is satisfied by all interpretations which satisfy the first.
- **A valid statement is satisfiable, and a contradictory statement is invalid, but the converse is not necessarily true.**

Example: On The above definitions:

- **P** is satisfiable but **not valid** since an interpretation **that assigns false to P assigns false to the sentence P**.
- **P V ~P** is **valid** since every interpretation results in a value of **true for (P V ~P).**
- **P & ~P** is a **contradiction** since every interpretation results in a value of **false for (P & ~P)**.

Conti…

- **P and ~(~P)** are **equivalent** since each has the **same truth values** under every interpretation.
- **P** is a **logical consequence** of (P & Q) since any interpretation for which **(P & Q) is true**, **P is also true**.

Theorem

- **Theorem 4.1: The sentence s is a logical consequence of s_1 , s_2 , ... , s_n if and only if s_1 & s_2 & s_3 . . . . . & $s_n \rightarrow$ s is valid.**
- **Proof:** Theorem 4.1 can be seen by first noting that if s is a logical consequence of s_1 , s_2 , ... , s_n , then for any interpretation ***I*** in which s_1 & s_2 & s_3 , ... &s_n $\rightarrow$ s is true.
- on the other hand, if s_1 & s_2 & s_3 , ... & s_n $\rightarrow$s is valid, then for any interpretation ***I*** if s_1 & s_2 & s_3 , ... &s_n is true, **s is also true**.

Theorem

- **Theorem 4.2:** The sentence s is a logical consequence of s_1 , s_2 , ... , s_n if and only if **s_1 & s_2 & s_3 , ... & s_n & ~s** is **inconsistent**.

- **Proof:** The proof of theorem 4.2 follows directly from theorem 4.1 since s is a logical consequence of s_1 , s_2 , ... , s_n if and only if s_1 & s_2 & s_3 , ... &s_n $\rightarrow$ s is valid, that is, if and only if ~(s_1 & s_2 & s_3 , ... &s_n $\rightarrow$ s) is inconsistent.

Conti .....

- But

$\sim(s_1 \,\&\, s_2 \,\&\, s_3 \,,\ \ldots\ldots\ \&s_n \rightarrow s\ ) = \sim(\sim(s_1 \,\&\, s_2 \,\&\, s_3 \,,\ \ldots\ldots\ \&s_n\ ) \vee s)$

[**By Conditional Elimination**]

$= \sim\sim(s_1 \,\&\, s_2 \,\&\, s_3 \,,\ \ldots\ldots\ \&s_n\ ) \,\&\, \sim s)$

[**By De Morgan's Law**]

$= s_1 \,\&\, s_2 \,\&\, s_3 \,,\ \ldots\ldots\ \&s_n \ \&\, \sim s$

- When s is a logical consequence of s_1 , s_2 , ... , s_n , the formula $s_1 \,\&\, s_2 \,\&\, s_3 \,,\ \ldots\ldots\ \&s_n \rightarrow s$ is called a theorem, with s is the conclusion.
- When **s** is a logical consequence of the set **S** = {s_1 , s_2 , ... , s_n } we will also set **S** logically implies or logically entails **S** , written $S \vdash s$.

Table 4.2 lists some of the important laws of PL (Some Equivalence Laws)

Name of Laws	Statements
Idempotency	P V P = P P & P = P
Associativity	(PVQ)VR = P V (Q V R) (P & Q) & R = P & (Q & R)
Commutativity	P V Q = Q V P P & Q = Q & P P ↔Q = Q↔P
Distributivity	P & (Q V R) = (P & Q)V (P &R) P V (Q & R) = (P V Q) & (P V R)
De Morgan's Laws	~(P V Q) = ~P & ~Q ~(P & Q) = ~P V ~Q
Conditional Elimination	P → Q = ~P V Q
Bi-conditional Elimination	P ↔Q = (p → Q) & (Q →P)

Example

- **Show that $P \rightarrow Q$ is equivalent to $\sim P \vee Q$ and that $P \leftrightarrow Q$ is equivalent to the expression $(P \rightarrow Q) \& (Q \rightarrow P)$.**

- The truth table 4.3 is given bellow.

TABLE 4.3 : Truth table for equitant sentences

P	Q	~P	(~PVQ)	(P→Q)	(Q→P)	(P→Q)& (Q→P)
T	T	F	T	T	T	T
T	F	F	F	F	T	F
F	T	T	T	T	F	F
F	F	T	T	T	T	T

Inference Rules

- **The inference rules of PL provide the means to perform logical proofs or deductions.**

- Few Such Rules are as follows:
  - **Modus ponens**
  - **Chain Rule**

Modus Ponens:

- From P and $P \rightarrow Q$ infer Q. This sometimes written as
- P
- $P \rightarrow Q$
- Q

Example For Modus Ponens:

- Given: (Joe is a father)
- And: (Joe is a father) $\rightarrow$ (Joe has a child)
- Conclude: (Joe has a child)

Chain Rule

- Form $P \rightarrow Q$ and $Q \rightarrow R$, infer $P \rightarrow R$.
- Or
- $P \rightarrow Q$
- $Q \rightarrow R$
- $P \rightarrow R$

Example for Chain Rule

- Given: (programmer likes LISP)→ (programmer hates COBOL)
- **and** : (programmer hates COBOL) → (programmer likes recursion)
- **Conclude**: (programmer likes LISP)→ (programmer likes recursion)

- LISP → **List** **P**rocessing
- COBOL → **Co**mmon **B**usiness **O**riented **L**anguage
- Prolog → **Programming in Log**ic

Assignment-3

1. Construct a **truth Table** for the expression **(A & (A V B))**.

2. Determine whether each of the following sentences is

(a) Satisfiable

(b) Contradictory

(c) Valid

Assignment-3 Conti…

- (i) S_1: (P & Q) V ˜(P & Q)
- (ii) S_2: (P V Q)→ (P & Q)
- (iii) S_3: (P V Q)→ R V ˜Q
- (iv) S_4: (P V Q)& (PV ˜Q) V P
- (v) S_5: P → Q→ ˜P
- (vi) S_6: P V Q & ˜P V ˜Q &P

- **Thanks**