Database and its Applications

Data Models and Mathematical Foundations

Pooja T S
Computer Applications

Database and its Applications

Data Models and Mathematical Foundations

Logical Equivalence, Implication, Quantifiers

Pooja T S

Computer Applications

Boolean Expressions – Introduction

- A **Boolean expression** is a logical statement that can have only two values:
  - **True (T)**
  - **False (F)**
- Propositions like ”2+2=4” or ”It is raining” can be represented as Boolean expressions.
- Logical connectives (AND, OR, NOT, etc.) combine these expressions to form compound statements.
- Truth tables are used to evaluate Boolean expressions under all possible truth values of their variables.

What is a Truth Table?

- A **truth table** is a tabular method used in logic to list all possible truth values of propositions and their combinations.
- Purpose:
  - Shows how compound statements behave under all conditions.
  - Helps determine whether a statement is always true, always false, or sometimes true.
- Key points:
  - Each proposition is either True (T) or False (F).
  - For *n* propositions, table has 2^n rows.
  - Columns represent simple or compound statements.

Logical Connectives

- Connectives combine propositions into compound statements.
- Common connectives:
  - Negation: $\neg P \rightarrow$ "not P"
  - Conjunction: $P \wedge Q \rightarrow$ "P and Q"
  - Disjunction: $P \vee Q \rightarrow$ "P or Q"
  - Implication: $P \Rightarrow Q \rightarrow$ "If P then Q"
  - Biconditional: $P \Leftrightarrow Q \rightarrow$ "P if and only if Q"

Steps to Build a Truth Table

- Step 1: List the simple propositions (P, Q, R).
- Step 2: Write all possible combinations of truth values.
- Step 3: Add columns for compound expressions.
- Step 4: Fill values row by row using logic rules.
- Number of rows = 2^n for n propositions.

Truth Tables – Basics

- Used to formally define logical connectives.
- Example: Conjunction ($P \wedge Q$)

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

Class Exercise – Truth Tables

- Task: Construct the truth tables for the following connectives:
  - Negation ($\neg P$)
  - Disjunction ($P \vee Q$)
  - Implication ($P \Rightarrow Q$)
  - Biconditional ($P \Leftrightarrow Q$)
- Instructions:
  - Use two variables P and Q.
  - List all possible truth value combinations.
  - Fill in the results for each connective.
- Discussion:
  - Compare your tables with a partner.
  - Be prepared to explain why each row is true or false.

Logical Equivalence – Definition

- Two formulas are **logically equivalent** if they have the same truth values under all interpretations.
- Notation: $P \equiv Q$
- Example Laws:
  - Commutativity: $P \vee Q \equiv Q \vee P$
  - Associativity: $(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$
  - De Morgan's: $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$

Logical Equivalence – Example Proof

- Verify: $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
- Truth Table:

P	Q	$\neg(P \vee Q)$	$\neg P$	$\neg P \wedge \neg Q$
T	T	F	F	F
T	F	F	F	F
F	T	F	T	F
F	F	T	T	T

- Columns match $\rightarrow$ formulas are equivalent.

Logical Implication – Definition

- $P \Rightarrow Q$ means: If P is true, then Q must also be true.
- Only false when P is true and Q is false.
- Equivalent form:

$$P \Rightarrow Q \equiv \neg P \vee Q$$

- Database example:
  - If a student is enrolled in DBMS, then that student must exist in Student table.

Logical Implication – Truth Table

P	Q	$P \Rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

- Note: Implication holds whenever P is false.

Quantifiers – Basics

- Predicate logic adds variables and quantifiers.
- Universal Quantifier ($\forall$):
  - $\forall x\, P(x) \rightarrow$ "For all x, $P(x)$ holds"
- Existential Quantifier ($\exists$):
  - $\exists x\, P(x) \rightarrow$ "There exists some x such that $P(x)$ holds"

Quantifiers – Negation Laws

- Negating quantifiers changes their form:
  - $\neg(\forall x\ P(x)) \equiv \exists x\ \neg P(x)$
  - $\neg(\exists x\ P(x)) \equiv \forall x\ \neg P(x)$
- Example:
  - "Not all students passed DBMS" $\equiv$ "There exists a student who did not pass DBMS."

Quantifiers – Examples

- Example 1 (Universal):

$$\forall x \; Student(x) \rightarrow Enrolled(x, DBMS)$$

Meaning: Every student is enrolled in DBMS.

- Example 2 (Existential):

$$\exists x \; Enrolled(x, OS)$$

Meaning: At least one student is enrolled in OS.

- Example 3 (Combined):

$$\forall c \; \exists p \; Teaches(p, c)$$

Meaning: Every course has at least one professor.

Applications in Databases

- Logical Equivalence:
  - Used for query rewriting and optimization.
- Logical Implication:
  - Used in expressing dependencies and constraints.
- Quantifiers:
  - Map to SQL constructs:
  - $\exists \rightarrow$ `EXISTS`
  - $\forall \rightarrow$ expressed via `NOT EXISTS`

Summary

- Logical Equivalence:
  - Two formulas always have same truth values.
- Logical Implication:
  - If P is true, then Q must also be true.
- Quantifiers:
  - $\forall$: property holds for all elements.
  - $\exists$: property holds for at least one element.
  - Negation rules switch quantifiers.
- All three are foundational for database queries and constraints.

Thank You

Pooja T S
Assistant Professor
Department of Computer Applications
poojats@pes.edu
080-26721983 Extn: 233