Database and its Applications

Data Models and Mathematical Foundations

Pooja T S
Computer Applications

Database and its Applications

Data Models and Mathematical Foundations

Principle of Inclusion and Exclusion; Predicate Logic

Pooja T S

Computer Applications

- **What is PIE?**
  - A counting method used when sets **overlap**.
  - Helps find $|A \cup B \cup C \dots|$ without double counting.
  - Works by alternately adding and subtracting overlaps.
- **Why needed?**
  - If sets are disjoint: $|A \cup B| = |A| + |B|$.
  - If not disjoint: adding directly counts intersections multiple times.

Two-Set PIE – Example

- Define sets:
  - A = students who like Cricket ($|A| = 40$).
  - B = students who like Football ($|B| = 30$).
  - $|A \cap B| = 10$ (students like both).
- Naive count: $40 + 30 = 70$ (overcounts 10 students).
- Correct using PIE:

$$|A \cup B| = |A| + |B| - |A \cap B| = 40 + 30 - 10 = 60$$

- Interpretation: 60 students like at least one sport.

Three-Set PIE – Formula

- General expression:

$$|A \cup B \cup C| = |A| + |B| + |C| - (|A \cap B| + |A \cap C| + |B \cap C|) + |A \cap B \cap C|$$

- Explanation:
  - Add single sets.
  - Subtract pairwise overlaps.
  - Add back the triple overlap.
  - Sets: M = failed Math ($|M| = 128$), P = failed Physics ($|P| = 87$), C = failed Chemistry ($|C| = 134$).

Three-Set PIE – Example (Failed Students)

- Given overlaps:
  - $|M \cap P| = 31$, $|M \cap C| = 54$, $|P \cap C| = 30$.
  - Total failed = 250.
- PIE calculation:

$$|M \cup P \cup C| = |M| + |P| + |C| - (|M \cap P| + |M \cap C| + |P \cap C|) + |M \cap P \cap C|$$

$$250 = (128 + 87 + 134) - (31 + 54 + 30) + x$$

$$250 = 349 - 115 + x = 234 + x$$

$$x = 16$$

- Result: 16 students failed in all three subjects.

General PIE – n Sets

- For n sets $A_1, A_2, \ldots, A_n$:

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{i=1}^{n} |A_i| - \sum_{1\le i<j\le n} |A_i \cap A_j| + \sum_{1\le i<j<k\le n} |A_i \cap A_j \cap A_k| - \cdots + (-1)^{n+1}\left|\bigcap_{i=1}^{n} A_i\right|$$

PIE – Pattern Recap

- Pattern of Inclusion–Exclusion:
  - Add all single sets.
  - Subtract pairwise intersections.
  - Add triple intersections.
  - Continue alternating signs.

PIE with Complements – "At Least One"

- Often easier to compute:

$$\left|\bigcup_{i=1}^{n} A_i\right| = |U| - \left|\bigcap_{i=1}^{n} A_i^c\right|$$

- Interpretation: Total elements − those in none.
- Example: 3-digit numbers with at least one digit 7.
  - Total = 900 (100–999).
  - No digit 7: $8 \times 9 \times 9 = 648$.
  - With at least one 7: $900 - 648 = 252$.

Applications of PIE

- PIE is useful for:
  - Survey analysis – who likes which options.
  - Exam analysis – students failing multiple subjects.
  - Probability – union of events.
  - Counting derangements (no fixed points).
  - Databases – duplicate removal, query overlap.

Summary – PIE and Predicate Logic

- Principle of Inclusion–Exclusion:
  - Prevents double counting.
  - Key in query optimization and set operations.

Activity

- Activity:
  - Let $|A| = 25$, $|B| = 20$, $|A \cap B| = 5$. Compute $|A \cup B|$.
  - Express the condition “All students in class A have registered for at least one course.” in predicate logic.
- Discussion: Connect your answers to SQL queries.

Thank You

Pooja T S

Assistant Professor

Department of Computer Applications

poojats@pes.edu

080-26721983 Extn: 233

Database and its Applications

Data Models and Mathematical Foundations

Principle of Inclusion and Exclusion; Predicate Logic

Pooja T S

Computer Applications

Introduction to Logic

- Logic provides a formal framework to reason about statements.
- In databases:
  - Queries are expressed as logical conditions.
  - Constraints use logical formulas.
  - Optimization relies on logical equivalence.

Predicate Logic – Introduction

- Predicate logic extends propositional logic by including variables and quantifiers.
- A **predicate** is a statement involving variables that becomes true or false when values are assigned.
- Example:
  - $Enrolled(x, mca101)$ – true if student x is enrolled in course mca101.
- Importance:
  - Foundation of query languages (SQL uses predicates in WHERE clauses).
  - Provides formalism for expressing database constraints.
- Example Predicate:
  - $Enrolled(x, mca101)$ – “Student x is enrolled in mca101.”

Predicate Logic – Quantifiers

- Universal Quantifier ($\forall$):
  - $\forall x \, Student(x) \rightarrow Enrolled(x, DBMS)$
  - Meaning: All students are enrolled in DBMS.
- Existential Quantifier ($\exists$):
  - $\exists x \, Enrolled(x, OS)$
  - Meaning: There exists a student enrolled in OS.
- Mapping to SQL:
  - $\exists \rightarrow$ `EXISTS` clause
  - $\forall \rightarrow$ checked via `NOT EXISTS` negation

Predicate Logic – Applications in DBMS

- Query Expression:
  - “Find all students enrolled in both DBMS and OS.”
  - Logic: $Enrolled(x, DBMS) \wedge Enrolled(x, OS)$
- Integrity Constraints:
  - “Every course must have at least one instructor.”
  - Logic: $\forall c\ \exists p\ Teaches(p, c)$
- Helps in proving equivalence of SQL queries.

Introduction to Propositions

- Motivation:
  - Logic is the foundation of mathematics and computer science.
  - We need a way to reason formally using true/false values.
- A **proposition** is a declarative statement that is either true or false, but not both.
- Examples:
  - "2 + 3 = 5" → True.
  - "Bangalore is the capital of India" → False.

What is Not a Proposition?

- Not all sentences are propositions.
- Examples:
  - Questions: ”What time is it?” (no truth value).
  - Commands: ”Close the door!” (no truth value).
  - Expressions: ”$x + 2 = 5$” (depends on x).

Compound Statements – Connectives

- Compound statements are formed by combining simple propositions using **logical connectives**.
- Common connectives:
  - Negation ($\neg p$): "not p"
  - Conjunction ($p \wedge q$): "p and q"
  - Disjunction ($p \vee q$): "p or q"
  - Implication ($p \rightarrow q$): "if p, then q"
  - Biconditional ($p \leftrightarrow q$): "p if and only if q"

Negation – Truth Table

- Negation reverses the truth value.

p	$\neg p$
T	F
F	T

Conjunction and Disjunction

- Conjunction ($p \wedge q$): True only if both p and q are true.
- Disjunction ($p \vee q$): True if at least one of p or q is true.

p	q	$p \wedge q$	$p \vee q$
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

Implication and Biconditional

- Implication ($p \rightarrow q$): False only when p is true and q is false.
- Biconditional ($p \leftrightarrow q$): True when p and q have the same truth value.

p	q	$p \rightarrow q$	$p \leftrightarrow q$
T	T	T	T
T	F	F	F
F	T	T	F
F	F	T	T

Examples of Compound Propositions

- Example 1: p: ”It is raining”, q: ”I will carry an umbrella”.
  - $p \wedge q$: ”It is raining and I will carry an umbrella.”
  - $p \rightarrow q$: ”If it is raining, then I will carry an umbrella.”
- Example 2:
  - p: ”2 is even”
  - q: ”3 is prime”
  - $p \vee q$: ”2 is even or 3 is prime.” → True.

Summary

- A proposition is a statement with a truth value (True/False).
- Non-propositions include questions, commands, open statements.
- Compound propositions are formed using logical connectives.
- Truth tables define the meaning of connectives.
- Propositions form the basis of reasoning in mathematics, CS, and AI.

Thank You

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