CE 474 - LOGIC OF COMPUTER SCIENCE

Lecture Note 4

1. Motivation & Real-World Connections

Before diving into symbols, let's see why predicate logic matters:

- **Database Queries**: "Find all users who have never logged in."
  $\forall u \, (\text{User}(u) \, \wedge \, \neg\exists t \, \text{Login}(u, t))$
- **Program Verification**: "Every sorted array has its elements in non-decreasing order."
  $\forall A \, (\text{Sorted}(A) \rightarrow \forall \, i < j \; A[i] \leq A[j])$
- **Artificial Intelligence**: "There exists a path from start to goal."
  $\exists p \, \text{Path}(start, p, goal)$

Takeaway: Predicate logic gives us the language to express "for all" and "there exists" precisely—crucial when specs must be unambiguous.

2. Learning Objectives

By the end of this lecture you will be able to:

1. Translate English statements into predicate-logic formulas using $\forall$ and $\exists$.
2. Specify domains and understand how they shape meaning.
3. Negate quantified statements correctly.
4. Use counterexamples to refute universal claims and proofs to demonstrate existential ones.

3. Syntax & Semantics Refresher

Variables

Placeholders ranging over a domain D. For example, $x, y \in D$ where $D = \{\text{all students}\}$.

Predicates

Properties or relations on variables. Examples:

- $\text{CS}(x)$: "x is a CS student."
- $\text{Takes}(x, \text{DM})$: "$x$ takes Discrete Math."

Quantifiers

- Universal: $\forall x\, P(x)$ means "for all x in the domain, $P(x)$ holds."
- Existential: $\exists x\, P(x)$ means "there exists at least one x in the domain such that $P(x)$."

Domain Specification

Clarify your universe. "Over students" means x ranges only over students.

Scope & Binding

Parentheses matter. Examples:

- $\forall x\, (P(x) \to Q(x))$
- $(\forall x\, P(x)) \to Q$

These are not the same!

4. Translating English → Predicate Logic

English Statement	Formal Translation
Every CS student has taken Discrete Math.	$\forall x\, [\text{CS}(x) \to \text{TookDM}(x)]$
Some algorithms terminate.	$\exists a\ \text{Terminates}(a)$
No student skipped the final exam.	$\neg\exists x\, [\text{Student}(x) \land \text{SkippedFE}(x)]$
Exactly one professor is absent today.	$\exists! p\ \text{Absent}(p)$

Tip: "Exactly one" can be rendered as $\exists x\, [P(x) \land \forall y\, (P(y) \to y = x)]$.

5. Negation of Quantified Statements

To push a negation through a quantifier:

$$\neg(\forall x\, P(x)) \iff \exists x\, \neg P(x), \quad \neg(\exists x\, P(x)) \iff \forall x\, \neg P(x)$$

Example

$$\neg(\forall x\, \text{Likes}(x, \text{Pizza})) \equiv \exists x\, \neg\text{Likes}(x, \text{Pizza})$$

"There is someone who does not like pizza."

Exercise

Negate: "There exists a solution to the equation $f(x) = 0$."

6. Counterexamples & Proofs

Refuting Universal Claims

To refute $\forall x\, P(x)$, find c such that $\neg P(c)$.

Example: "All primes are odd."

Counterexample: 2 is prime but not odd $\Rightarrow \neg(\forall p\, \text{Prime}(p) \to \text{Odd}(p))$.

Proving Existential Claims

To prove $\exists x\, P(x)$, exhibit w such that $P(w)$.

Example: Prove "There exists an even Fibonacci number greater than 10."

Try: $F_6 = 8$, $F_7 = 13$, $F_8 = 21$, etc.

7. In-Class Exercises

1. **Translation & Negation**

    - "There exists a student who has never missed a quiz."
    - "All primes are odd."

    For each: write ϕ, then $\neg\phi$.

2. **Counterexample Challenge**
    Statement: $\forall n \in N,\ n > 1 \to n$ is composite.
    Is it true? If not, give a counterexample.

3. **Domain Variations**
    "Every binary string starts with '0'."
    (a) Domain = all binary strings.
    (b) Domain = strings of even length.
    Discuss how the truth changes.

8. Summary & Key Takeaways

- Quantifiers ($\forall$, $\exists$) let us express "all" vs. "some" precisely.
- Domains and scope are critical—never assume a hidden universe.
- Negation rules flip quantifiers: $\forall \leftrightarrow \exists$ and push in $\neg$.
- Counterexamples disprove universals; witnesses prove existentials.

Further Reading

- Huth & Ryan, *Logic in Computer Science.*
- Harrison, *Mathematical Logic with Applications.*
- Khan Academy interactive logic tools.