Logic of Computer Science
Lecture 4: Predicate Logic – Quantifiers and Domains

Emmanuel Kwesi Tandoh

University of Mines and Technology (UMaT)

June 2025

UNIVERSITY OF MINES AND TECHNOLOGY (UMaT)

Knowledge | Truth | Excellence

Real-World Connections

Why Predicate Logic Matters

Before diving into symbols, let's see some real-world applications:

Database Queries

"Find all users who have never logged in."

$$\forall u \left(\text{User}(u) \land \neg \exists t \ \text{Login}(u, t)\right)$$

Program Verification

"Every sorted array has its elements in non-decreasing order."

$$\forall A \left(\text{Sorted}(A) \to \forall i < j \ A[i] \leq A[j]\right)$$

Real-World Connections Continued

Artificial Intelligence

"There exists a path from start to goal."

$$\exists p \; \mathrm{Path}(\mathrm{start}, p, \mathrm{goal})$$

Takeaway

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

Learning Objectives

By the end of this session, students should be able to:

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

- Translate English statements into formulas using $\forall, \exists$.
- Specify and clarify domains for variables.
- Negate quantified statements: push $\neg$ through $\forall \leftrightarrow \exists$.
- Use counterexamples and witnesses in proofs.

Syntax & Semantics Refresher

Variables

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

Predicates

Properties or relations on variables.

- CS(x): "x is a CS student."
- Takes(x, 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 $P(x)$."

Syntax & Semantics Refresher Continued

Domain Specification

Always clarify your universe: *"Over students"* means x ranges only over students.

Scope & Binding

Parentheses matter in logic:

$$\forall x\,(P(x) \to Q(x)) \quad \text{vs.} \quad (\forall x\, P(x)) \to Q$$

These are **not** the same.

Examples of Translation

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

Translating English $\to$ Predicate Logic

Tip: How to express "Exactly one"

$$\exists x\,[\,P(x) \land \forall y\,(P(y) \to y = x)\,]$$

This means: There exists a unique x such that $P(x)$ is true.

Negation of Quantified Statements

Quantifier Negation Rules

To push a negation through a quantifier, apply the following equivalences:

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

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

Example

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

Interpretation: "There is someone who does not like pizza."

Negation of Quantified Statements

Exercise

Negate: "There exists a solution to the equation $f(x) = 0$."
Try: Express $\neg(\exists x\, f(x) = 0)$ using universal quantification.

Refuting Universal Claims

Universal Claim: $\forall x\, P(x)$

To refute a universal statement, find a counterexample:

$$\text{Find } c \text{ such that } \neg P(c)$$

Example

Claim: "All primes are odd."

$$\forall p\,(\mathrm{Prime}(p) \to \mathrm{Odd}(p))$$

Counterexample: $p = 2$ is a prime but not odd.

$$\Rightarrow \neg(\forall p\,(\mathrm{Prime}(p) \to \mathrm{Odd}(p)))$$

Therefore, the universal statement is false.

Proving Existential Claims

Existential Claim: $\exists x\, P(x)$

To prove an existential statement, provide a witness:

$$\text{Exhibit } w \text{ such that } P(w)$$

Example

Claim: "There exists an even Fibonacci number greater than 10." Let's try a few Fibonacci numbers:

- $F_6 = 8 \rightarrow$ No
- $F_7 = 13 \rightarrow$ Not even
- $F_8 = 21 \rightarrow$ Not even

Try more or prove by construction that such a number exists.

In-Class Exercises: Translation & Counterexample

1. Translation & Negation

- (a) Statement: "There exists a student who has never missed a quiz."

In-Class Exercises: Translation & Counterexample

1. Translation & Negation

- (a) Statement: "There exists a student who has never missed a quiz."
  - ϕ: $\exists x\,(\text{Student}(x) \wedge \neg\,\text{MissedQuiz}(x))$

In-Class Exercises: Translation & Counterexample

1. Translation & Negation

- (a) Statement: "There exists a student who has never missed a quiz."
  - ϕ: $\exists x\,(\text{Student}(x) \wedge \neg\,\text{MissedQuiz}(x))$
  - $\neg\phi$: $\forall x\,(\neg\,\text{Student}(x) \vee \text{MissedQuiz}(x))$

In-Class Exercises: Translation & Counterexample

1. Translation & Negation

- (a) Statement: "There exists a student who has never missed a quiz."
  - ϕ: $\exists x\,(\mathrm{Student}(x) \land \neg\,\mathrm{MissedQuiz}(x))$
  - $\neg\phi$: $\forall x\,(\neg\,\mathrm{Student}(x) \lor \mathrm{MissedQuiz}(x))$
- (b) Statement: "All primes are odd."

In-Class Exercises: Translation & Counterexample

1. Translation & Negation

- (a) Statement: "There exists a student who has never missed a quiz."
  - ϕ: $\exists x\,(\text{Student}(x) \land \neg\,\text{MissedQuiz}(x))$
  - $\neg\phi$: $\forall x\,(\neg\,\text{Student}(x) \lor \text{MissedQuiz}(x))$
- (b) Statement: "All primes are odd."
  - ϕ: $\forall p\,(\text{Prime}(p) \to \text{Odd}(p))$

In-Class Exercises: Translation & Counterexample

1. Translation & Negation

- (a) Statement: "There exists a student who has never missed a quiz."
  - ϕ: $\exists x\,(\text{Student}(x) \wedge \neg\,\text{MissedQuiz}(x))$
  - $\neg\phi$: $\forall x\,(\neg\,\text{Student}(x) \vee \text{MissedQuiz}(x))$
- (b) Statement: "All primes are odd."
  - ϕ: $\forall p\,(\text{Prime}(p) \rightarrow \text{Odd}(p))$
  - $\neg\phi$: $\exists p\,(\text{Prime}(p) \wedge \neg\,\text{Odd}(p))$

In-Class Exercises: Translation & Counterexample

1. Translation & Negation

- (a) Statement: "There exists a student who has never missed a quiz."
  - ϕ: $\exists x\,(\text{Student}(x) \wedge \neg\,\text{MissedQuiz}(x))$
  - $\neg\phi$: $\forall x\,(\neg\,\text{Student}(x) \vee \text{MissedQuiz}(x))$
- (b) Statement: "All primes are odd."
  - ϕ: $\forall p\,(\text{Prime}(p) \to \text{Odd}(p))$
  - $\neg\phi$: $\exists p\,(\text{Prime}(p) \wedge \neg\,\text{Odd}(p))$

2. Counterexample Challenge

Statement: $\forall n \in \mathbb{N},\; n > 1 \to n$ is composite

In-Class Exercises: Translation & Counterexample

1. Translation & Negation

- (a) Statement: "There exists a student who has never missed a quiz."
  - ϕ: $\exists x\,(\text{Student}(x) \land \neg\,\text{MissedQuiz}(x))$
  - $\neg\phi$: $\forall x\,(\neg\,\text{Student}(x) \lor \text{MissedQuiz}(x))$
- (b) Statement: "All primes are odd."
  - ϕ: $\forall p\,(\text{Prime}(p) \to \text{Odd}(p))$
  - $\neg\phi$: $\exists p\,(\text{Prime}(p) \land \neg\,\text{Odd}(p))$

2. Counterexample Challenge

Statement: $\forall n \in \mathbb{N},\ n > 1 \to n$ is composite

Is it true? No.

In-Class Exercises: Translation & Counterexample

1. Translation & Negation

- (a) Statement: "There exists a student who has never missed a quiz."
  - ϕ: $\exists x\,(\text{Student}(x) \land \neg\,\text{MissedQuiz}(x))$
  - $\neg\phi$: $\forall x\,(\neg\,\text{Student}(x) \lor \text{MissedQuiz}(x))$
- (b) Statement: "All primes are odd."
  - ϕ: $\forall p\,(\text{Prime}(p) \to \text{Odd}(p))$
  - $\neg\phi$: $\exists p\,(\text{Prime}(p) \land \neg\,\text{Odd}(p))$

2. Counterexample Challenge

Statement: $\forall n \in \mathbb{N},\ n > 1 \to n$ is composite

Is it true? No.

Counterexample: $n = 2$

In-Class Exercises: Translation & Counterexample

1. Translation & Negation

- (a) Statement: “There exists a student who has never missed a quiz.”
  - ϕ: $\exists x\,(\text{Student}(x) \wedge \neg\,\text{MissedQuiz}(x))$
  - $\neg\phi$: $\forall x\,(\neg\,\text{Student}(x) \vee \text{MissedQuiz}(x))$
- (b) Statement: “All primes are odd.”
  - ϕ: $\forall p\,(\text{Prime}(p) \to \text{Odd}(p))$
  - $\neg\phi$: $\exists p\,(\text{Prime}(p) \wedge \neg\,\text{Odd}(p))$

2. Counterexample Challenge

Statement: $\forall n \in \mathbb{N},\ n > 1 \to n$ is composite

Is it true? No.

Counterexample: $n = 2$

$2 > 1$ but 2 is prime, not composite. $\Rightarrow$ Statement is **false**.

3. Statement

"Every binary string starts with '0'."

In-Class Exercises: Domain Variations

3. Statement

"Every binary string starts with '0'."

(a) Domain: All binary strings

In-Class Exercises: Domain Variations

3. Statement

"Every binary string starts with '0'."

(a) Domain: All binary strings

- Consider the string `"1"` — does it start with '0'?

In-Class Exercises: Domain Variations

3. Statement

"Every binary string starts with '0'."

(a) Domain: All binary strings

- Consider the string `"1"` — does it start with '0'? No.
- Counterexample: `"10"`, `"1"` statement is **false**.

In-Class Exercises: Domain Variations

3. Statement

"Every binary string starts with '0'."

(a) Domain: All binary strings

- Consider the string `"1"` — does it start with '0'? No.
- Counterexample: `"10"`, `"1"` statement is **false**.

(b) Domain: Binary strings of even length

In-Class Exercises: Domain Variations

3. Statement

"Every binary string starts with '0'."

(a) Domain: All binary strings

- Consider the string `"1"` — does it start with '0'? No.
- Counterexample: `"10"`, `"1"` statement is **false**.

(b) Domain: Binary strings of even length

- Consider `"10"`, `"11"` — both are even length.
- Do they start with '0'?

In-Class Exercises: Domain Variations

3. Statement

"Every binary string starts with '0'."

(a) Domain: All binary strings

- Consider the string `"1"` — does it start with '0'? No.
- Counterexample: `"10"`, `"1"` statement is **false**.

(b) Domain: Binary strings of even length

- Consider `"10"`, `"11"` — both are even length.
- Do they start with '0'? Not necessarily. Still **false**.

3. Statement

"Every binary string starts with '0'."

(a) Domain: All binary strings

- Consider the string `"1"` — does it start with '0'? No.
- Counterexample: `"10"`, `"1"` statement is **false**.

(b) Domain: Binary strings of even length

- Consider `"10"`, `"11"` — both are even length.
- Do they start with '0'? Not necessarily. Still **false**.

Discussion Prompt:

How does changing the domain affect the truth value of statements?

Summary & Key Takeaways

What You Should Remember

- **Quantifiers** ($\forall$, $\exists$) let us express "*all*" vs. "*some*" precisely.

Summary & Key Takeaways

What You Should Remember

- **Quantifiers** ($\forall$, $\exists$) let us express "*all*" vs. "*some*" precisely.
- **Domains** and **scope** are critical—never assume a hidden universe.

Summary & Key Takeaways

What You Should Remember

- **Quantifiers** ($\forall$, $\exists$) let us express "*all*" vs. "*some*" precisely.
- **Domains** and **scope** are critical—never assume a hidden universe.
- **Negation Rules:** Flip quantifiers and push the negation in:
  $$\neg\forall x\, P(x) \equiv \exists x\, \neg P(x), \qquad \neg\exists x\, P(x) \equiv \forall x\, \neg P(x)$$

Summary & Key Takeaways

What You Should Remember

- **Quantifiers** ($\forall$, $\exists$) let us express "*all*" vs. "*some*" precisely.
- **Domains** and **scope** are critical—never assume a hidden universe.
- **Negation Rules:** Flip quantifiers and push the negation in:
  $\neg\forall x\, P(x) \;\equiv\; \exists x\, \neg P(x), \qquad \neg\exists x\, P(x) \;\equiv\; \forall x\, \neg P(x)$
- **Counterexamples** disprove universal claims.
  Witnesses validate existential claims.

Summary & Key Takeaways

What You Should Remember

- **Quantifiers** ($\forall$, $\exists$) let us express "*all*" vs. "*some*" precisely.
- **Domains** and **scope** are critical—never assume a hidden universe.
- **Negation Rules:** Flip quantifiers and push the negation in:
  $\neg\forall x\, P(x) \;\equiv\; \exists x\, \neg P(x), \qquad \neg\exists x\, P(x) \;\equiv\; \forall x\, \neg P(x)$
- **Counterexamples** disprove universal claims.
  Witnesses validate existential claims.

Final Tip

Approach logic like debugging code—check definitions, test boundaries, and validate truth.

References I

- M. Huth and M. Ryan, *Logic in Computer Science: Modelling and Reasoning about Systems*, Cambridge University Press, 2004.
- D. Gries and F. B. Schneider, *A Logical Approach to Discrete Math*, Springer-Verlag, 1993.
- K. H. Rosen, *Discrete Mathematics and Its Applications*, 7th ed., McGraw-Hill, 2011.
- P. Halmos, *Naive Set Theory*, Van Nostrand, 1960.

Thank You!