The Language of Predicate Logic

Brendan Shea, PhD

June 14, 2025

Why Logic Matters: From Ancient Greece to Modern AI

- As we've learned, logic is the **science of correct reasoning**, helping us distinguish good arguments from bad ones.
- Ancient Greek philosophers used logic to debate everything from ethics to the nature of reality.
- Today's computers run on logic—every app, game, and AI system uses logical rules at its core.
- Learning logic is like learning a superpower: you'll spot flawed arguments and think more clearly about complex problems.

Fun Fact

The same logical principles that Aristotle discovered 2,300 years ago now power Google's search algorithms!

Meet Aristotle: The First Logician

- **Aristotle** (384-322 BCE) was the first person to systematically study the rules of reasoning.
- He noticed that certain argument patterns always work, regardless of what they're about.
- His most famous pattern: "All humans are mortal. Socrates is human. Therefore, Socrates is mortal."
- This **syllogism** (a type of logical argument) has the same structure whether we're talking about humans, hobbits, or Hogwarts students!

Example

1. All wizards can do magic
2. Harry Potter is a wizard
3. Therefore, Harry Potter can do magic

Logic in Everyday Life: Arguments We Make Without Realizing It

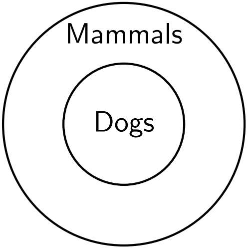
- We use logical reasoning dozens of times each day without even noticing it.
- When you conclude "I need an umbrella" from "It's raining" and "I don't want to get wet," you're using logic!
- **Deductive reasoning** helps us draw certain conclusions from known facts.
- However, everyday language can be ambiguous—that's why we need a more precise logical language.

Common Logical Patterns You Already Use

- If-then reasoning: "If I don't study, then I'll fail the test"
- Either-or reasoning: "Either the butler did it, or the gardener did"
- Elimination: "It can't be the butler, so it must be the gardener"

From "All Dogs Are Mammals" to Computer Programming

- The statement "All dogs are mammals" contains a **universal claim** about every member of a group.
- Computers need to handle such statements precisely—think of a database searching for "all customers who bought books."
- **Predicate logic** gives us tools to express relationships like "is a mammal," "bought," or "is friends with."
- Modern programming languages use these same logical structures in their code!



All dogs are inside the mammals circle

How Logic Helps Us Think More Clearly

- Logic acts like **grammar for reasoning**—it shows us the structure beneath our thoughts.
- Just as grammar helps us construct clear sentences, logic helps us construct clear arguments.
- When we formalize our reasoning, hidden assumptions become visible and errors become obvious.
- Logic training has been shown to improve performance in mathematics, computer science, and even creative writing!

Think About It

How many times have you felt someone's argument was wrong but couldn't explain exactly why? Logic gives you the tools to identify and explain logical flaws!

The Limits of Everyday Language: When Words Get Confusing

- Natural language is wonderfully expressive but often **ambiguous** (having multiple meanings).
- Consider: "I saw the person with the telescope"—who has the telescope?
- The word "some" can mean "at least one" or "not all"—which creates confusion in arguments.
- **Context-dependence** means the same sentence can mean different things in different situations.

Example

The sentence "Everyone loves someone" could mean:

1. Each person has someone they love (maybe different people)
2. There's one special person that everyone loves (like Keanu Reeves!)

In predicate logic, we can express each meaning precisely!

Enter Predicate Logic: A Clearer Way to Express Ideas

- **Predicate logic** is a formal language designed to eliminate ambiguity in reasoning.
- It breaks statements down into **individuals** (things we're talking about) and **predicates** (properties or relations).
- Unlike everyday language, each predicate logic statement has exactly one meaning.
- This precision makes it perfect for mathematics, computer science, philosophy, and any field requiring careful reasoning.

What's Coming Next

We'll learn to:

- Identify individuals and predicates in statements
- Use logical operators like "and," "or," and "not"
- Express "all" and "some" statements precisely
- Translate English arguments into logical form

The Two Main Ingredients: Things and Their Properties

- Predicate logic has two basic components: **individuals** and **predicates**.
- **Individuals** are the specific things we're talking about—like Hermione, Gandalf, or your pet goldfish.
- **Predicates** describe properties of individuals or relationships between them.
- Together, they form **atomic statements**—the simplest meaningful units in our logical language.

Individual	Predicate	Statement
Frodo	is brave	Frodo is brave
Hogwarts	is magical	Hogwarts is magical
My cat	sleeps all day	My cat sleeps all day

Individual Names: Giving Things Specific Labels

- **Individual names** (also called constants) refer to specific, unique objects or beings.
- In logic, we often use descriptive names like "Sherlock" or "BakerStreet" instead of single letters.
- Each name refers to exactly one thing—"Harry" means just Harry Potter, not all people named Harry.
- Think of individual names as the logical equivalent of proper nouns in English.

Example

Individual names in our logical language:

- `Aragorn` — refers to the specific character from Lord of the Rings
- `Enterprise` — refers to the specific starship
- `MrDarcy` — refers to the character from Pride and Prejudice
- `Paris` — refers to the city in France

Predicates: Describing Properties and Relationships

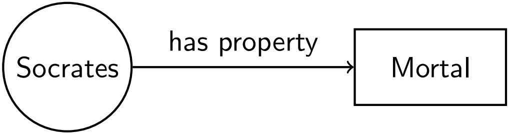
- **Predicates** express properties that individuals might have or relationships between individuals.
- A predicate is like a sentence with blanks: "___ is tall" or "___ loves ___".
- The number of blanks is called the **arity** of the predicate—one blank means it's unary, two means binary.
- We write predicates with descriptive names followed by the individuals in parentheses.

Predicate Examples

- `IsWizard(Harry)` — "Harry is a wizard" (unary predicate)
- `TeachesAt(Dumbledore, Hogwarts)` — "Dumbledore teaches at Hogwarts" (binary)
- `Introduced(Tolkien, Frodo, Readers)` — "Tolkien introduced Frodo to readers" (ternary)

"Socrates is Mortal": Our First Predicate Logic Statement

- Let's translate the famous statement "Socrates is mortal" into predicate logic.
- We need an individual name: `Socrates` (referring to the philosopher).
- We need a predicate: `IsMortal(___)` (expressing the property of being mortal).
- The complete statement: `IsMortal(Socrates)`—read as "Socrates has the property of being mortal."



`IsMortal(Socrates)`

One-Place Predicates: Properties of Single Things

- **One-place predicates** (also called unary predicates) describe properties of single individuals.
- They answer questions like "What is it?" or "What is it like?"
- In English, these often appear as "is [adjective]" or "is a [noun]" phrases.
- One-place predicates are the simplest type and perfect for beginning our logic journey!

Example

Common one-place predicates with fantasy characters:

- `IsElf(Legolas)` — "Legolas is an elf"
- `IsBrave(Katniss)` — "Katniss is brave"
- `CanFly(Buckbeak)` — "Buckbeak can fly"
- `IsEvil(Voldemort)` — "Voldemort is evil"

Two-Place Predicates: Relationships Between Things

- **Two-place predicates** (binary predicates) express relationships between two individuals.
- They answer questions like "How are these things related?" or "What does X do to Y?"
- The order matters: `Loves(Romeo, Juliet)` is different from `Loves(Juliet, Romeo)`!
- Most verbs in English that take objects become two-place predicates in logic.

Relationship Examples

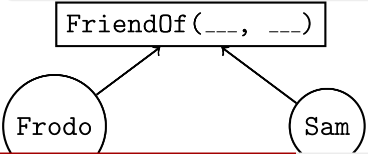
Predicate Logic	**English**
`Teaches(McGonagall, Hermione)`	McGonagall teaches Hermione
`DefeatedBy(Sauron, Frodo)`	Sauron was defeated by Frodo
`MarriedTo(Elizabeth, MrDarcy)`	Elizabeth is married to Mr. Darcy
`OwlOf(Hedwig, Harry)`	Hedwig is Harry's owl

Building Statements: Combining Names and Predicates

- To build a statement, we combine predicates with the right number of individual names.
- Think of predicates as functions that need the correct number of inputs to work.
- A predicate becomes a **proposition** (something true or false) when we fill in all its blanks.
- Getting the arity right is crucial—`Loves(Romeo)` is incomplete and meaningless!

Remember the Rules

- One-place predicate + one individual = complete statement
- Two-place predicate + two individuals = complete statement
- Wrong number of individuals = ERROR!



Practice: Identifying Individuals and Predicates in Sentences

- Let's practice breaking down English sentences into individuals and predicates.
- First, identify the specific things being discussed (individuals).
- Then, identify what's being said about them (predicates).
- Finally, determine the arity—how many individuals does the predicate connect?

Example

Translate these sentences:

1. "Gandalf is wise"
   - Individual: `Gandalf`
   - Predicate: `IsWise(___)`
   - Translation: `IsWise(Gandalf)`
2. "Hermione helps Ron with homework"
   - Individuals: `Hermione`, `Ron`
   - Predicate: `HelpsWithHomework(___, ___)`
   - Translation: `HelpsWithHomework(Hermione, Ron)`

The Power of "And": Conjunction in Logic

- The **conjunction** operator (written as $\wedge$) connects two statements with "and."
- A conjunction is true only when **both** parts are true—it's very demanding!
- We read $P \wedge Q$ as "P and Q" where P and Q are any statements.
- Think of $\wedge$ as a strict gatekeeper that only lets truth through when everything checks out.

Conjunction Example

"Harry is a wizard AND Harry plays Quidditch"

- Let $P = \texttt{IsWizard(Harry)}$
- Let $Q = \texttt{PlaysQuidditch(Harry)}$
- Combined: $\texttt{IsWizard(Harry)} \wedge \texttt{PlaysQuidditch(Harry)}$
- This is true only if Harry is both a wizard and a Quidditch player!

The Inclusive "Or": Disjunction Explained

- The **disjunction** operator (written as $\vee$) connects statements with "or."
- Unlike everyday English, logical "or" is **inclusive**—it's true when at least one part is true.
- $P \vee Q$ is true when P is true, Q is true, or both are true!
- This matches how computers think: "Do you want fries or a drink?" can mean both!

Example

"Hermione is in the library OR Hermione is in class"

- $P = \texttt{InLibrary(Hermione)}$
- $Q = \texttt{InClass(Hermione)}$
- Combined: $\texttt{InLibrary(Hermione)} \vee \texttt{InClass(Hermione)}$
- This is false only if Hermione is neither in the library nor in class

The Art of Denial: Understanding Negation

- The **negation** operator (written as $\neg$) expresses "not" or "it is not the case that."
- Negation simply flips the truth value: if P is true, then $\neg P$ is false, and vice versa.
- We read $\neg P$ as "not P" or "it is not the case that P."
- Negation is the logical equivalent of pressing the "opposite" button!

Negation in Action

Original	Negated	Meaning
`IsEvil(Voldemort)`	$\neg$`IsEvil(Voldemort)`	Voldemort is not evil
`CanFly(Ron)`	$\neg$`CanFly(Ron)`	Ron cannot fly
$P \wedge Q$	$\neg(P \wedge Q)$	It's not the case that both P and Q

"If...Then": Conditional Statements

- The **conditional** operator (written as $\rightarrow$) expresses "if...then" relationships.
- $P \rightarrow Q$ means "if P is true, then Q must be true."
- Surprisingly, a conditional is only false when P is true but Q is false!
- This captures promises: "If you study, then you'll pass" is only broken if you study but still fail.

Example

"If Dobby gets a sock, then Dobby is free"

- $P = \texttt{GetsSock(Dobby)}$
- $Q = \texttt{IsFree(Dobby)}$
- Conditional: $\texttt{GetsSock(Dobby)} \rightarrow \texttt{IsFree(Dobby)}$
- This is false only if Dobby gets a sock but isn't freed!

When is $P \rightarrow Q$ true?

P true, Q true: ✓

P true, Q false: ×

P false, Q true: ✓

P false, Q false: ✓

"If and Only If": Biconditional Connections

- The **biconditional** operator (written as $\leftrightarrow$) means "if and only if."
- $P \leftrightarrow Q$ is true when P and Q have the **same truth value**—both true or both false.
- This expresses a two-way relationship: P implies Q, AND Q implies P.
- Think of it as logical equality: "P happens exactly when Q happens."

Biconditional Example

"You're a Gryffindor if and only if the Sorting Hat placed you in Gryffindor"

- $P = \texttt{IsGryffindor(You)}$
- $Q = \texttt{SortedIntoGryffindor(You)}$
- Biconditional: $\texttt{IsGryffindor(You)} \leftrightarrow \texttt{SortedIntoGryffindor(You)}$
- True when both happen or neither happens!

Truth Tables: Mapping All Possibilities

- **Truth tables** show all possible combinations of truth values for statements.
- They help us understand exactly when complex statements are true or false.
- Each row represents one possible scenario, and we calculate the result for each.
- Truth tables are like instruction manuals for logical operators!

Example

Truth table for $P \wedge Q$ (Hermione is smart AND brave):

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

Only true when Hermione is both smart (P) and brave (Q)!

Building Complex Statements with Multiple Operators

- We can combine multiple operators to express complex ideas precisely.
- Use parentheses to show which operations happen first—just like in math!
- Without parentheses, we follow precedence: $\neg$ first, then $\wedge$, then $\vee$, then $\rightarrow$ and $\leftrightarrow$.
- Complex statements let us capture the full richness of logical relationships.

Complex Statement Example

"If Harry defeats Voldemort, then either Harry survives or he becomes a legend"

- Let D = `Defeats(Harry, Voldemort)`
- Let S = `Survives(Harry)`
- Let L = `BecomesLegend(Harry)`
- Translation: $D \rightarrow (S \vee L)$

This captures that defeating Voldemort leads to at least one positive outcome!

Practice: Creating Truth Tables Together

- Let's build a truth table for: $(P \wedge Q) \to R$
- This means: "If P and Q are both true, then R is true"
- Example: "If it's raining and I forgot my umbrella, then I'll get wet"
- We need to check all 8 possible combinations of P, Q, and R!

P	Q	R	$P \wedge Q$	$(P \wedge Q) \to R$
T	T	T	T	T
T	T	F	T	F
T	F	T	F	T
T	F	F	F	T
F	T	T	F	T
F	T	F	F	T
F	F	T	F	T
F	F	F	F	T

Beyond Individual Statements: Talking About Groups

- So far, we've only made claims about specific individuals like Harry or Hermione.
- But what about statements like "All wizards can do magic" or "Some students like math"?
- **Quantifiers** let us make claims about entire groups without naming each member.
- The two main quantifiers are **universal** (all) and **existential** (some).

Why Quantifiers Matter

Without quantifiers, we'd need to write:

- `CanDoMagic(Harry)` $\wedge$ `CanDoMagic(Hermione)` $\wedge$ `CanDoMagic(Ron)` $\wedge$ ...

With quantifiers, we can simply say:

- "For all x, if x is a wizard, then x can do magic"

"All" Statements: Universal Quantification

- The **universal quantifier** (written as $\forall$) means "for all" or "for every."
- $\forall x$ reads as "for all x" where x is a variable that can represent any individual.
- Universal statements make claims about **every** member of a group—no exceptions!
- They're often combined with conditionals to express rules or laws.

Example

"All hobbits are short"

- Translation: $\forall x\ (\texttt{IsHobbit}(x) \rightarrow \texttt{IsShort}(x))$
- Read as: "For all x, if x is a hobbit, then x is short"
- This claims that being a hobbit guarantees being short
- To prove it false, we'd only need to find one tall hobbit!

"Some" Statements: Existential Quantification

- The **existential quantifier** (written as $\exists$) means "there exists" or "for some."
- $\exists x$ reads as "there exists an x" or "for some x."
- Existential statements claim that **at least one** thing has a certain property.
- They're weaker than universal statements—we only need one example to make them true!

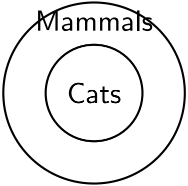
Existential Example

"Some students love logic class"

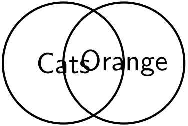
- Translation: $\exists x\ (\texttt{IsStudent}(x) \land \texttt{LovesLogic}(x))$
- Read as: "There exists an x such that x is a student and x loves logic"
- Note: We use $\land$ (and) here, not $\rightarrow$ (if-then)!
- This is true if we can find even one logic-loving student

The Difference Between "All Cats Are Mammals" and "Some Cats Are Orange"

- Universal and existential statements have very different logical structures.
- "All cats are mammals": $\forall x\ (\texttt{IsCat}(x) \to \texttt{IsMammal}(x))$
- "Some cats are orange": $\exists x\ (\texttt{IsCat}(x) \wedge \texttt{IsOrange}(x))$
- Notice: Universal uses $\to$, but existential uses $\wedge$!



All cats inside mammals



Some overlap exists

Combining Quantifiers with Predicates

- Quantifiers can be combined with any predicates and logical operators we've learned.
- The **scope** of a quantifier is the part of the formula it applies to—usually shown with parentheses.
- Variables can be **bound** (controlled by a quantifier) or **free** (not controlled).
- Multiple quantifiers can be used in the same statement for complex relationships!

Example

"Every student has a favorite teacher"

- Translation: $\forall x\ (\texttt{IsStudent}(x) \rightarrow \exists y\ (\texttt{IsTeacher}(y) \wedge\ \texttt{IsFavoriteOf}(y, x)))$
- This says: For every student x, there exists a teacher y who is x's favorite
- Notice how we use two different variables (x and y) for the two quantifiers!

Common Mistakes: When "All" and "Some" Get Tricky

- **Mistake 1**: Using $\wedge$ with universal quantifiers instead of $\rightarrow$.
- **Mistake 2**: Using $\rightarrow$ with existential quantifiers instead of $\wedge$.
- **Mistake 3**: Forgetting that "some" means "at least one"—it could be all!
- **Mistake 4**: Confusing the order of multiple quantifiers—order matters!

Watch Out!

Wrong	**Right**
$\forall x\ (\texttt{Dog}(x) \wedge \texttt{Barks}(x))$ Says everything is a dog that barks!	$\forall x\ (\texttt{Dog}(x) \rightarrow \texttt{Barks}(x))$ Says all dogs bark
$\exists x\ (\texttt{Bird}(x) \rightarrow \texttt{CanFly}(x))$ True even if no birds exist!	$\exists x\ (\texttt{Bird}(x) \wedge \texttt{CanFly}(x))$ Says some bird can fly

Translating Between English and Logic: Quantifier Practice

- Translating quantified statements requires careful attention to English phrasing.
- Words like "every," "all," "each" usually signal universal quantification ($\forall$).
- Words like "some," "there is," "at least one" signal existential quantification ($\exists$).
- Watch for hidden quantifiers—"Dogs bark" really means "All dogs bark"!

Translation Guide

English Pattern	Logical Form
All A's are B	$\forall x\ (A(x) \rightarrow B(x))$
Some A's are B	$\exists x\ (A(x) \wedge B(x))$
No A's are B	$\forall x\ (A(x) \rightarrow \neg B(x))$
Not all A's are B	$\neg\forall x\ (A(x) \rightarrow B(x))$
Only A's are B	$\forall x\ (B(x) \rightarrow A(x))$

Real-World Applications: How Scientists Use Quantifiers

- Scientists use quantified statements to express laws, theories, and hypotheses precisely.
- Medical research: "All patients with condition X respond to treatment Y" can be tested.
- Computer science: Database queries use quantifiers to search for specific patterns.
- Mathematics: Theorems are often universal statements that must be proven for all cases.

Example

Scientific statements using quantifiers:

- Physics: $\forall x\ (\texttt{IsMass}(x) \rightarrow \texttt{AttractsOtherMass}(x))$
- Biology: $\exists x\ (\texttt{IsOrganism}(x) \land \texttt{LivesInExtremeHeat}(x))$
- Chemistry: $\forall x\ (\texttt{IsNobleGas}(x) \rightarrow \texttt{IsStable}(x))$

These precise statements can be tested and verified through experiments!

From English to Logic: A Step-by-Step Method

- **Step 1**: Identify the main claim—what is being said about what?
- **Step 2**: Determine if it's about all, some, or specific individuals.
- **Step 3**: Identify predicates and their arity (how many blanks to fill).
- **Step 4**: Choose appropriate logical connectives and build the formula.

The Translation Process

1. Read the sentence carefully
2. Circle quantifier words (all, some, every, no)
3. Underline the properties or relationships
4. Write the logical formula piece by piece
5. Check: Does your translation capture the original meaning?

Example 1: "Every Student Who Studies Passes the Test"

- Let's break down this sentence step by step using our method.
- Main claim: There's a connection between studying and passing for all students.
- Quantifier: "Every" indicates universal quantification ($\forall$).
- Predicates needed: `IsStudent`, `Studies`, and `PassesTest`.

Example

Building the translation:

1. Start with: $\forall x$ (...)
2. Add student condition: $\forall x\ (\texttt{IsStudent}(x) \rightarrow ...)$
3. Add studying condition: $\forall x\ ((\texttt{IsStudent}(x) \wedge \texttt{Studies}(x)) \rightarrow ...)$
4. Add conclusion: $\forall x\ ((\texttt{IsStudent}(x) \wedge \texttt{Studies}(x)) \rightarrow \texttt{PassesTest}(x))$

"For all x, if x is a student and x studies, then x passes the test"

Example 2: "Some Musicians Play Multiple Instruments"

- This sentence claims that at least one musician has a special property.
- Quantifier: "Some" indicates existential quantification ($\exists$).
- The tricky part: "multiple instruments" needs careful handling.
- We need: `IsMusician` and `PlaysInstrument` predicates.

Translation Options

Option 1 (Simple): Create a predicate for the property

- $\exists x\ (\texttt{IsMusician}(x) \wedge \texttt{PlaysMultipleInstruments}(x))$

Option 2 (Detailed): Express "multiple" explicitly

- $\exists x\ (\texttt{IsMusician}(x) \wedge \exists y \exists z\ (\texttt{IsInstrument}(y) \wedge \texttt{IsInstrument}(z) \wedge y \neq z \wedge \texttt{Plays}(x, y) \wedge \texttt{Plays}(x, z)))$

Both capture the meaning, but Option 1 is clearer for beginners!

Example 3: "No Reptiles Are Warm-Blooded"

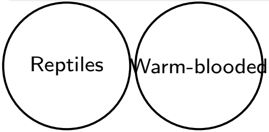
- "No" statements are universal claims about what is NOT true.
- This says: "For all things, if it's a reptile, then it's not warm-blooded."
- Predicates: `IsReptile` and `IsWarmBlooded`.
- Remember: "No A's are B" means $\forall x\ (A(x) \rightarrow \neg B(x))$.

Example

Translation: $\forall x\ (\texttt{IsReptile}(x) \rightarrow \neg\texttt{IsWarmBlooded}(x))$

Alternative expressions of the same idea:

- $\neg\exists x\ (\texttt{IsReptile}(x) \wedge \texttt{IsWarmBlooded}(x))$
- "There does not exist a reptile that is warm-blooded"
- Both formulas say the same thing—no warm-blooded reptiles exist!



Common Translation Pitfalls and How to Avoid Them

- **Pitfall 1**: Translating "only" incorrectly—"Only wizards can apparate" is NOT the same as "All wizards can apparate"!
- **Pitfall 2**: Missing hidden universal quantifiers—"Birds fly" usually means "All birds fly."
- **Pitfall 3**: Misplacing negation—"Not all heroes wear capes" is different from "All heroes don't wear capes."
- **Pitfall 4**: Wrong connectives with quantifiers—remember the $\rightarrow/\wedge$ rule!

Quick Reference for Tricky Phrases

English	**Logic**
Only A's are B	$\forall x\ (B(x) \rightarrow A(x))$
All A's are not B	$\forall x\ (A(x) \rightarrow \neg B(x))$
Not all A's are B	$\neg\forall x\ (A(x) \rightarrow B(x))$
A's are always B	$\forall x\ (A(x) \rightarrow B(x))$

Your Turn: Practice Translations

- Do your best to translate these statements into predicate logic.
- Remember to identify quantifiers, predicates, and use our step-by-step method.
- Different approaches might both be correct!

Practice Problems

1. "Some superheroes can't fly" (Think: Thor vs. Spider-Man)
2. "Every vampire fears garlic and sunlight"
3. "Only jedis can use the Force"
4. "Not every wizard attended Hogwarts"
5. "If someone is a Time Lord, then they have two hearts"
6. "There's a hobbit who has never left the Shire"

What We've Learned: The Power of Precise Thinking

- We've built a **formal language** that eliminates ambiguity in logical reasoning.
- You can now translate complex English statements into precise logical formulas.
- You understand how **truth tables** show us exactly when statements are true or false.
- You've mastered the building blocks: individuals, predicates, operators, and quantifiers!

Your New Logical Toolkit

- **Predicates**: Express properties and relationships clearly
- **Logical operators**: $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$
- **Quantifiers**: $\forall$ (all) and $\exists$ (some)
- **Translation skills**: Convert everyday arguments into logical form

Next Steps: Where Predicate Logic Can Take You

- **Mathematics**: Predicate logic is the foundation for mathematical proofs and set theory.
- **Computer Science**: Programming, databases, and AI all use predicate logic daily.
- **Philosophy**: Analyze complex arguments about ethics, knowledge, and existence.
- **Critical Thinking**: Spot logical fallacies in advertisements, politics, and everyday discussions.

Example

Future Applications You'll Encounter:

- Writing SQL database queries using logical operators
- Creating "if-then" rules in programming languages
- Understanding mathematical proofs in advanced classes
- Building logical arguments in debate or essay writing