

The Foundations: Logic and Proofs

Chapter 1, Part II: Predicate Logic

With Question/Answer Animations

Recap: Propositional Logic

Summary

- The Language of Propositions
 - Connectives
 - Truth Values
 - Truth Tables
- Applications
 - Translating English Sentences
 - System Specifications
 - Logic Puzzles
 - Logic Circuits
- Logical Equivalences
 - Important Equivalences
 - Showing Equivalence
 - Satisfiability

Key Logical Equivalences

- Identity Laws(恒等律): $p \wedge T \equiv p$, $p \vee F \equiv p$
- Domination Laws(支配律): $p \vee T \equiv T$, $p \wedge F \equiv F$
- Idempotent Laws(幂等律): $p \vee p \equiv p$, $p \wedge p \equiv p$
- Double Negation Law(双重否定律): $\neg(\neg p) \equiv p$
- Negation Laws(否定律): $p \vee \neg p \equiv T$, $p \wedge \neg p \equiv F$

Key Logical Equivalences (*cont*)

- **Commutative Laws:** $p \vee q \equiv q \vee p$, $p \wedge q \equiv q \wedge p$
(交换律)
- **Associative Laws:** $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
(结合律)
 $(p \vee q) \vee r \equiv p \vee (q \vee r)$
- **Distributive Laws:** $(p \vee (q \wedge r)) \equiv (p \vee q) \wedge (p \vee r)$
(分配律)
 $(p \wedge (q \vee r)) \equiv (p \wedge q) \vee (p \wedge r)$
- **Absorption Laws:** $p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$
(吸收律)

Disjunctive Normal Form (*optional*)

- A propositional formula is in *disjunctive normal form* if it consists of a disjunction of $(1, \dots, n)$ disjuncts where each disjunct consists of a conjunction of $(1, \dots, m)$ atomic formulas or the negation of an atomic formula.
 - Yes $(p \wedge \neg q) \vee (\neg p \vee q)$
 - No $p \wedge (p \vee q)$
- Disjunctive Normal Form is important for the circuit design methods discussed in Chapter 12.

disjunctive Normal Form 析取范式

Disjunctive Normal Form (optional)

Example: Show that every compound proposition can be put in disjunctive normal form.

Solution: Construct the truth table for the proposition. Then an equivalent proposition is the disjunction with n disjuncts (where n is the number of rows for which the formula evaluates to **T**). Each disjunct has m conjuncts where m is the number of distinct propositional variables. Each conjunct includes the positive form of the propositional variable if the variable is assigned **T** in that row and the negated form if the variable is assigned **F** in that row. This proposition is in disjunctive normal form.

Disjunctive Normal Form (optional)

Example: Find the Disjunctive Normal Form (DNF) of

$$(p \vee q) \rightarrow \neg r$$

Solution: This proposition is true when r is false or when both p and q are false.

$$(\neg p \wedge \neg q) \vee \neg r$$

More Logical Equivalences

TABLE 7 Logical Equivalences
Involving Conditional Statements.

$$p \rightarrow q \equiv \neg p \vee q$$

$$p \rightarrow q \equiv \neg q \rightarrow \neg p$$

$$p \vee q \equiv \neg p \rightarrow q$$

$$p \wedge q \equiv \neg(p \rightarrow \neg q)$$

$$\neg(p \rightarrow q) \equiv p \wedge \neg q$$

$$(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$$

$$(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$$

$$(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$$

$$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$$

TABLE 8 Logical
Equivalences Involving
Biconditional Statements.

$$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$$

$$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$$

$$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$$

$$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$$

Conjunctive Normal Form (optional)

- A compound proposition is in *Conjunctive Normal Form* (CNF) if it is a conjunction of disjunctions.
- Every proposition can be put in an equivalent CNF.
- Conjunctive Normal Form (CNF) can be obtained by eliminating implications, moving negation inwards and using the distributive and associative laws.
- Important in resolution theorem proving used in artificial Intelligence (AI).
- A compound proposition can be put in conjunctive normal form through repeated application of the logical equivalences covered earlier.

Conjunctive Normal Form (optional)

Example: Put the following into CNF:

$$\neg(p \rightarrow q) \vee (r \rightarrow p)$$

Solution:

1. Eliminate implication signs:

$$\neg(\neg p \vee q) \vee (\neg r \vee p)$$

2. Move negation inwards; eliminate double negation:

$$(p \wedge \neg q) \vee (\neg r \vee p)$$

3. Convert to CNF using associative/distributive laws

$$(p \vee \neg r \vee p) \wedge (\neg q \vee \neg r \vee p)$$

Propositional Satisfiability

- A compound proposition is *satisfiable* if there is an assignment of truth values to its variables that make it true. When no such assignments exist, the compound proposition is *unsatisfiable*.
- A compound proposition is unsatisfiable if and only if its negation is a tautology.

satisfiable 可满足的
unsatisfiable 不可满足的

Questions on Propositional Satisfiability

Example: Determine the satisfiability of the following compound propositions:

$$(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$$

Solution: Satisfiable. Assign **T** to p , q , and r .

$$(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$$

Solution: Satisfiable. Assign **T** to p and **F** to q .

$$(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$$

Solution: Not satisfiable. Check each possible assignment of truth values to the propositional variables and none will make the proposition true.

Notation

$\bigvee_{j=1}^n p_j$ is used for $p_1 \vee p_2 \vee \dots \vee p_n$

$\bigwedge_{j=1}^n p_j$ is used for $p_1 \wedge p_2 \wedge \dots \wedge p_n$

Needed for the next example.

Sudoku

- A **Sudoku puzzle** is represented by a 9×9 grid made up of nine 3×3 subgrids, known as **blocks**. Some of the 81 cells of the puzzle are assigned one of the numbers 1, 2, ..., 9.
- The puzzle is solved by assigning numbers to each blank cell so that every row, column and block contains each of the nine possible numbers.
- Example

	2	9				4		
			5			1		
	4							
				4	2			
6							7	
5								
7			3					5
	1			9				
							6	

Encoding as a Satisfiability Problem

- Let $p(i,j,n)$ denote the proposition that is true when the number n is in the cell in the i th row and the j th column.
- There are $9 \times 9 \times 9 = 729$ such propositions.
- In the sample puzzle $p(5,1,6)$ is true, but $p(5,j,6)$ is false for $j = 2,3,\dots,9$

Encoding (cont)

- For each cell with a given value, assert $p(i,j,n)$, when the cell in row i and column j has the given value.
- Assert that every row contains every number.

$$\bigwedge_{i=1}^9 \bigwedge_{n=1}^9 \bigvee_{j=1}^9 p(i, j, n)$$

- Assert that every column contains every number.

$$\bigwedge_{j=1}^9 \bigwedge_{n=1}^9 \bigvee_{i=1}^9 p(i, j, n)$$

Encoding (cont)

- Assert that each of the 3×3 blocks contain every number.

$$\bigwedge_{r=0}^2 \bigwedge_{s=0}^2 \bigwedge_{n=1}^9 \bigwedge_{i=1}^3 \bigvee_{j=1}^3 p(3r + i, 3s + j, n)$$

(this is tricky - ideas from chapter 4 help)

- Assert that no cell contains more than one number. Take the conjunction over all values of n, n', i , and j , where each variable ranges from 1 to 9 and $n \neq n'$, of

$$p(i, j, n) \rightarrow \neg p(i, j, n')$$

Solving Satisfiability Problems

- To solve a Sudoku puzzle, we need to find an assignment of truth values to the 729 variables of the form $p(i,j,n)$ that makes the conjunction of the assertions true. Those variables that are assigned T yield a solution to the puzzle.
- A truth table can always be used to determine the satisfiability of a compound proposition. But this is too complex even for modern computers for large problems.
- There has been much work on developing efficient methods for solving satisfiability problems as many practical problems can be translated into satisfiability problems.

Summary

- Predicate Logic (First-Order Logic (FOL), Predicate Calculus)
 - The Language of Quantifiers
 - Logical Equivalences
 - Nested Quantifiers
 - Translation from Predicate Logic to English
 - Translation from English to Predicate Logic

Predicate Logic 谓词逻辑
First-Order Logic 一阶逻辑
Predicate Calculus 谓词演算

Predicates and Quantifiers

Section 1.4

Section Summary

- Predicates
- Variables
- Quantifiers
 - Universal Quantifier
 - Existential Quantifier
- Negating Quantifiers
 - De Morgan's Laws for Quantifiers
- Translating English to Logic
- Logic Programming (*optional*)

Propositional Logic Not Enough

- If we have:
 - “All men are mortal.”
 - “Socrates is a man.”
- Does it follow that “Socrates is mortal?”
- Can’t be represented in propositional logic. Need a language that talks about objects, their properties, and their relations.
- Later we’ll see how to draw inferences.

Introducing Predicate Logic

- Predicate logic uses the following new features:
 - Variables: x, y, z
 - Predicates: $P(x), M(x)$
 - **Quantifiers** (*to be covered in a few slides*):
- **Propositional functions** are a generalization of propositions.
 - They contain variables and a predicate, e.g., $P(x)$
 - Variables can be replaced by elements from their **domain**.

Quantifiers 量词
Propositional functions 命题函数
domain 域

Propositional Functions

- Propositional functions become propositions (and have truth values) when their variables are each replaced by a value from the *domain* (or *bound* by a quantifier, as we will see later).
- The statement $P(x)$ is said to be the value of the propositional function P at x .
- For example, let $P(x)$ denote “ $x > 0$ ” and the domain be the integers. Then:
 - $P(-3)$ is false.
 - $P(0)$ is false.
 - $P(3)$ is true.
- Often the domain is denoted by U . So in this example U is the integers.

Examples of Propositional Functions

- Let “ $x + y = z$ ” be denoted by $R(x, y, z)$ and U (for all three variables) be the integers. Find these truth values:

$R(2, -1, 5)$

Solution: F

$R(3, 4, 7)$

Solution: T

$R(x, 3, z)$

Solution: Not a Proposition

- Now let “ $x - y = z$ ” be denoted by $Q(x, y, z)$, with U as the integers. Find these truth values:

$Q(2, -1, 3)$

Solution: T

$Q(3, 4, 7)$

Solution: F

$Q(x, 3, z)$

Solution: Not a Proposition

Compound Expressions

- Connectives from propositional logic carry over to predicate logic.
- If $P(x)$ denotes “ $x > 0$,” find these truth values:
 - $P(3) \vee P(-1)$ **Solution:** T
 - $P(3) \wedge P(-1)$ **Solution:** F
 - $P(3) \rightarrow P(-1)$ **Solution:** F
 - $P(3) \rightarrow \neg P(-1)$ **Solution:** T
- Expressions with variables are not propositions and therefore do not have truth values. For example,
 - $P(3) \wedge P(y)$
 - $P(x) \rightarrow P(y)$
- When used with quantifiers (to be introduced next), these expressions (propositional functions) become propositions.



Charles Peirce (1839-1914)

Quantifiers

- We need *quantifiers* to express the meaning of English words including *all* and *some*:
 - “All men are Mortal.”
 - “Some cats do not have fur.”
- The two most important quantifiers are:
 - *Universal Quantifier*, “For all,” symbol: \forall
 - *Existential Quantifier*, “There exists,” symbol: \exists
- We write as in $\forall x P(x)$ and $\exists x P(x)$.
- $\forall x P(x)$ asserts $P(x)$ is true for every x in the *domain*.
- $\exists x P(x)$ asserts $P(x)$ is true for some x in the *domain*.
- The quantifiers are said to bind the variable x in these expressions.

<i>Universal Quantifier</i>	全称量词
<i>Existential Quantifier</i>	存在量词

Universal Quantifier

- $\forall x P(x)$ is read as “For all x , $P(x)$ ” or “For every x , $P(x)$ ”

Examples:

- 1) If $P(x)$ denotes “ $x > 0$ ” and U is the integers, then $\forall x P(x)$ is false.
- 2) If $P(x)$ denotes “ $x > 0$ ” and U is the positive integers, then $\forall x P(x)$ is true.
- 3) If $P(x)$ denotes “ x is even” and U is the integers, then $\forall x P(x)$ is false.

Existential Quantifier

- $\exists x P(x)$ is read as “For some x , $P(x)$ ”, or as “There is an x such that $P(x)$,” or “For at least one x , $P(x)$.”

Examples:

1. If $P(x)$ denotes “ $x > 0$ ” and U is the integers, then $\exists x P(x)$ is true. It is also true if U is the positive integers.
2. If $P(x)$ denotes “ $x < 0$ ” and U is the positive integers, then $\exists x P(x)$ is false.
3. If $P(x)$ denotes “ x is even” and U is the integers, then $\exists x P(x)$ is true.

Thinking about Quantifiers

- When the domain of discourse is finite, we can think of quantification as looping through the elements of the domain.
- To evaluate $\forall x P(x)$ loop through all x in the domain.
 - If at every step $P(x)$ is true, then $\forall x P(x)$ is true.
 - If at a step $P(x)$ is false, then $\forall x P(x)$ is false and the loop terminates.
- To evaluate $\exists x P(x)$ loop through all x in the domain.
 - If at some step, $P(x)$ is true, then $\exists x P(x)$ is true and the loop terminates.
 - If the loop ends without finding an x for which $P(x)$ is true, then $\exists x P(x)$ is false.
- Even if the domains are infinite, we can still think of the quantifiers this fashion, but the loops will not terminate in some cases.

Properties of Quantifiers

- The truth value of $\exists x P(x)$ and $\forall x P(x)$ depend on both the propositional function $P(x)$ and on the domain U .
- **Examples:**
 1. If U is the positive integers and $P(x)$ is the statement “ $x < 2$ ”, then $\exists x P(x)$ is true, but $\forall x P(x)$ is false.
 2. If U is the negative integers and $P(x)$ is the statement “ $x < 2$ ”, then both $\exists x P(x)$ and $\forall x P(x)$ are true.
 3. If U consists of 3, 4, and 5, and $P(x)$ is the statement “ $x > 2$ ”, then both $\exists x P(x)$ and $\forall x P(x)$ are true. But if $P(x)$ is the statement “ $x < 2$ ”, then both $\exists x P(x)$ and $\forall x P(x)$ are false.

Precedence of Quantifiers

- The quantifiers \forall and \exists have higher precedence than all the logical operators.
- For example, $\forall x P(x) \vee Q(x)$ means $(\forall x P(x)) \vee Q(x)$
- $\forall x (P(x) \vee Q(x))$ means something different.
- Unfortunately, often people write $\forall x P(x) \vee Q(x)$ when they mean $\forall x (P(x) \vee Q(x))$.

Translating from English to Logic

Example 1: Translate the following sentence into predicate logic: “Every student in this class has taken a course in Java.”

Solution:

First decide on the domain U .

Solution 1: If U is all students in this class, define a propositional function $J(x)$ denoting “ x has taken a course in Java” and translate as $\forall x J(x)$.

Solution 2: But if U is all people, also define a propositional function $S(x)$ denoting “ x is a student in this class” and translate as $\forall x (S(x) \rightarrow J(x))$.

$\forall x (S(x) \wedge J(x))$ is not correct. What does it mean?

Translating from English to Logic

Example 2: Translate the following sentence into predicate logic: “Some student in this class has taken a course in Java.”

Solution:

First decide on the domain U .

Solution 1: If U is all students in this class, translate as

$$\exists x J(x)$$

Solution 2: But if U is all people, then translate as

$$\exists x (S(x) \wedge J(x))$$

$\exists x (S(x) \rightarrow J(x))$ is not correct. What does it mean?

Returning to the Socrates Example

- Introduce the propositional functions $Man(x)$ denoting “ x is a man” and $Mortal(x)$ denoting “ x is mortal.” Specify the domain as all people.
- The two premises are: $\forall x(Man(x) \rightarrow Mortal(x))$
 $Man(Socrates)$
- The conclusion is: $Mortal(Socrates)$
- Later we will show how to prove that the conclusion follows from the premises.

Equivalences in Predicate Logic

- Statements involving predicates and quantifiers are *logically equivalent* if and only if they have the same truth value
 - for every predicate substituted into these statements and
 - for every domain of discourse used for the variables in the expressions.
- The notation $S \equiv T$ indicates that S and T are logically equivalent.
- **Example:** $\forall x \neg \neg S(x) \equiv \forall x S(x)$

Thinking about Quantifiers as Conjunctions and Disjunctions

- If the domain is finite, a universally quantified proposition is equivalent to a conjunction of propositions without quantifiers and an existentially quantified proposition is equivalent to a disjunction of propositions without quantifiers.
- If U consists of the integers 1,2, and 3:

$$\forall x P(x) \equiv P(1) \wedge P(2) \wedge P(3)$$

$$\exists x P(x) \equiv P(1) \vee P(2) \vee P(3)$$

- Even if the domains are infinite, you can still think of the quantifiers in this fashion, but the equivalent expressions without quantifiers will be infinitely long.

Negating Quantified Expressions

- Consider $\forall x J(x)$

“Every student in your class has taken a course in Java.”

Here $J(x)$ is “x has taken a course in Java” and the domain is students in your class.

- Negating the original statement gives “It is not the case that every student in your class has taken Java.” This implies that “There is a student in your class who has not taken Java.”

Symbolically $\neg \forall x J(x)$ and $\exists x \neg J(x)$ are equivalent

Negating Quantified Expressions (continued)

- Now Consider $\exists x J(x)$

“There is a student in this class who has taken a course in Java.”

Where $J(x)$ is “x has taken a course in Java.”

- Negating the original statement gives “It is not the case that there is a student in this class who has taken Java.” This implies that “Every student in this class has not taken Java”

Symbolically $\neg \exists x J(x)$ and $\forall x \neg J(x)$ are equivalent

De Morgan's Laws for Quantifiers

- The rules for negating quantifiers are:

TABLE 2 De Morgan's Laws for Quantifiers.			
Negation	Equivalent Statement	When Is Negation True?	When False?
$\neg \exists x P(x)$	$\forall x \neg P(x)$	For every x , $P(x)$ is false.	There is an x for which $P(x)$ is true.
$\neg \forall x P(x)$	$\exists x \neg P(x)$	There is an x for which $P(x)$ is false.	$P(x)$ is true for every x .

- The reasoning in the table shows that:

$$\neg \forall x P(x) \equiv \exists x \neg P(x)$$

$$\neg \exists x P(x) \equiv \forall x \neg P(x)$$

- These are important. You will use these.

Translation from English to Logic

Examples:

1. “Some student in this class has visited Mexico.”

Solution: Let $M(x)$ denote “ x has visited Mexico” and $S(x)$ denote “ x is a student in this class,” and U be all people.

$$\exists x (S(x) \wedge M(x))$$

2. “Every student in this class has visited Canada or Mexico.”

Solution: Add $C(x)$ denoting “ x has visited Canada.”

$$\forall x (S(x) \rightarrow (M(x) \vee C(x)))$$

Some Fun with Translating from English into Logical Expressions

- $U = \{\text{fleegles, snurds, thingamabobs}\}$

$F(x)$: x is a fleegle

$S(x)$: x is a snurd

$T(x)$: x is a thingamabob

Translate “Everything is a fleegle”

Solution: $\forall x F(x)$

Translation (cont)

- $U = \{\text{fleegles, snurds, thingamabobs}\}$

$F(x)$: x is a fleegle

$S(x)$: x is a snurd

$T(x)$: x is a thingamabob

“Nothing is a snurd.”

Solution: $\neg \exists x S(x)$ What is this equivalent to?

Solution: $\forall x \neg S(x)$

Translation (cont)

- $U = \{\text{fleegles, snurds, thingamabobs}\}$

$F(x)$: x is a fleegle

$S(x)$: x is a snurd

$T(x)$: x is a thingamabob

“All fleegles are snurds.”

Solution: $\forall x (F(x) \rightarrow S(x))$

Translation (cont)

- $U = \{\text{fleegles, snurds, thingamabobs}\}$

$F(x)$: x is a fleegle

$S(x)$: x is a snurd

$T(x)$: x is a thingamabob

“Some fleegles are thingamabobs.”

Solution: $\exists x (F(x) \wedge T(x))$

Translation (cont)

- $U = \{\text{fleegles, snurds, thingamabobs}\}$

$F(x)$: x is a fleegle

$S(x)$: x is a snurd

$T(x)$: x is a thingamabob

“No snurd is a thingamabob.”

Solution: $\neg \exists x (S(x) \wedge T(x))$ What is this equivalent to?

Solution: $\forall x (\neg S(x) \vee \neg T(x))$

Translation (cont)

- $U = \{\text{fleegles, snurds, thingamabobs}\}$

$F(x)$: x is a fleegle

$S(x)$: x is a snurd

$T(x)$: x is a thingamabob

“If any fleegle is a snurd then it is also a thingamabob.”

Solution: $\forall x ((F(x) \wedge S(x)) \rightarrow T(x))$

System Specification Example

- Predicate logic is used for specifying properties that systems must satisfy.
- For example, translate into predicate logic:
 - “Every mail message larger than one megabyte will be compressed.”
 - “If a user is active, at least one network link will be available.”
- Decide on predicates and domains (left implicit here) for the variables:
 - Let $L(m, y)$ be “Mail message m is larger than y megabytes.”
 - Let $C(m)$ denote “Mail message m will be compressed.”
 - Let $A(u)$ represent “User u is active.”
 - Let $S(n, x)$ represent “Network link n is state x .”

- Now we have:
$$\forall m (L(m, 1) \rightarrow C(m))$$
$$\exists u A(u) \rightarrow \exists n S(n, available)$$



Charles Lutwidge Dodgson
(AKA Lewis Carroll)
(1832-1898)

Lewis Carroll Example

- The first two are called *premises* and the third is called the *conclusion*.
 1. “All lions are fierce.”
 2. “Some lions do not drink coffee.”
 3. “Some fierce creatures do not drink coffee.”
- Here is one way to translate these statements to predicate logic. Let $P(x)$, $Q(x)$, and $R(x)$ be the propositional functions “ x is a lion,” “ x is fierce,” and “ x drinks coffee,” respectively.
 1. $\forall x (P(x) \rightarrow Q(x))$
 2. $\exists x (P(x) \wedge \neg R(x))$
 3. $\exists x (Q(x) \wedge \neg R(x))$
- Later we will see how to prove that the conclusion follows from the premises.

Some Predicate Calculus

Definitions (*optional*)

- An assertion involving predicates and quantifiers is *valid* if it is true
 - for all domains
 - every propositional function substituted for the predicates in the assertion.

Example: $\forall x \neg S(x) \leftrightarrow \neg \exists x S(x)$

- An assertion involving predicates is *satisfiable* if it is true
 - for some domains
 - some propositional functions that can be substituted for the predicates in the assertion.

Otherwise it is *unsatisfiable*.

Example: $\forall x (F(x) \leftrightarrow T(x))$ not valid but satisfiable

Example: $\forall x (F(x) \wedge \neg F(x))$ unsatisfiable

valid 有效的
satisfiable 可满足的
unsatisfiable 不可满足的

More Predicate Calculus

Definitions (*optional*)

- The *scope* of a quantifier is the part of an assertion in which variables are bound by the quantifier.

Example: $\forall x(F(x) \vee S(x))$ x has wide scope

Example: $\forall x(F(x)) \vee \forall y(S(y))$ x has narrow scope

Nested Quantifiers

Section 1.4



Section Summary

- Nested Quantifiers
- Order of Quantifiers
- Translating from Nested Quantifiers into English
- Translating Mathematical Statements into Statements involving Nested Quantifiers.
- Translated English Sentences into Logical Expressions.
- Negating Nested Quantifiers.

Nested Quantifiers

- Nested quantifiers are often necessary to express the meaning of sentences in English as well as important concepts in computer science and mathematics.

Example: “Every real number has an inverse” is

$$\forall x \exists y (x + y = 0)$$

where the domains of x and y are the real numbers.

- We can also think of nested propositional functions:

$\forall x \exists y (x + y = 0)$ can be viewed as $\forall x Q(x)$ where $Q(x)$ is $\exists y P(x, y)$ where $P(x, y)$ is $(x + y = 0)$

Thinking of Nested Quantification

- Nested Loops

- To see if $\forall x \forall y P(x,y)$ is true, loop through the values of x :
 - At each step, loop through the values for y .
 - If for some pair of x and y , $P(x,y)$ is false, then $\forall x \forall y P(x,y)$ is false and both the outer and inner loop terminate.

$\forall x \forall y P(x,y)$ is true if the outer loop ends after stepping through each x .

- To see if $\forall x \exists y P(x,y)$ is true, loop through the values of x :
 - At each step, loop through the values for y .
 - The inner loop ends when a pair x and y is found such that $P(x,y)$ is true.
 - If no y is found such that $P(x,y)$ is true the outer loop terminates as $\forall x \exists y P(x,y)$ has been shown to be false.

$\forall x \exists y P(x,y)$ is true if the outer loop ends after stepping through each x .

- If the domains of the variables are infinite, then this process can not actually be carried out.

Order of Quantifiers

Examples:

1. Let $P(x,y)$ be the statement “ $x + y = y + x$.” Assume that U is the real numbers. Then $\forall x \forall y P(x,y)$ and $\forall y \forall x P(x,y)$ have the same truth value.
2. Let $Q(x,y)$ be the statement “ $x + y = 0$.” Assume that U is the real numbers. Then $\forall x \exists y Q(x,y)$ is true, but $\exists y \forall x Q(x,y)$ is false.

Questions on Order of Quantifiers

Example 1: Let U be the real numbers,

Define $P(x,y) : x \cdot y = 0$

What is the truth value of the following:

1. $\forall x \forall y P(x,y)$

Answer: False

2. $\forall x \exists y P(x,y)$

Answer: True

3. $\exists x \forall y P(x,y)$

Answer: True

4. $\exists x \exists y P(x,y)$

Answer: True

Questions on Order of Quantifiers

Example 2: Let U be the real numbers,

Define $P(x,y) : x / y = 1$

What is the truth value of the following:

1. $\forall x \forall y P(x,y)$

Answer: False

2. $\forall x \exists y P(x,y)$

Answer: False

3. $\exists x \forall y P(x,y)$

Answer: False

4. $\exists x \exists y P(x,y)$

Answer: True

Quantifications of Two Variables

Statement	When True?	When False
$\forall x \forall y P(x, y)$ $\forall y \forall x P(x, y)$	$P(x, y)$ is true for every pair x, y .	There is a pair x, y for which $P(x, y)$ is false.
$\forall x \exists y P(x, y)$	For every x there is a y for which $P(x, y)$ is true.	There is an x such that $P(x, y)$ is false for every y .
$\exists x \forall y P(x, y)$	There is an x for which $P(x, y)$ is true for every y .	For every x there is a y for which $P(x, y)$ is false.
$\exists x \exists y P(x, y)$ $\exists y \exists x P(x, y)$	There is a pair x, y for which $P(x, y)$ is true.	$P(x, y)$ is false for every pair x, y

Translating Nested Quantifiers into English

Example 1: Translate the statement

$$\forall x (C(x) \vee \exists y (C(y) \wedge F(x, y)))$$

where $C(x)$ is “ x has a computer,” and $F(x, y)$ is “ x and y are friends,” and the domain for both x and y consists of all students in your school.

Solution: Every student in your school has a computer or has a friend who has a computer.

Example 2: Translate the statement

$$\exists x \forall y \forall z ((F(x, y) \wedge F(x, z) \wedge (y \neq z)) \rightarrow \neg F(y, z))$$

Solution: There is a student none of whose friends are also friends with each other.

Translating Mathematical Statements into Predicate Logic

Example : Translate “The sum of two positive integers is always positive” into a logical expression.

Solution:

1. Rewrite the statement to make the implied quantifiers and domains explicit:

“For every two integers, if these integers are both positive, then the sum of these integers is positive.”

2. Introduce the variables x and y , and specify the domain, to obtain:

“For all positive integers x and y , $x + y$ is positive.”

3. The result is:

$$\forall x \forall y ((x > 0) \wedge (y > 0) \rightarrow (x + y > 0))$$

where the domain of both variables consists of all integers

Translating English into Logical Expressions Example

Example: Use quantifiers to express the statement
“There is a woman who has taken a flight on every
airline in the world.”

Solution:

1. Let $P(w,f)$ be “ w has taken f ” and $Q(f,a)$ be “ f is a flight on a .”
2. The domain of w is all women, the domain of f is all flights, and the domain of a is all airlines.
3. Then the statement can be expressed as:

$$\exists w \forall a \exists f (P(w,f) \wedge Q(f,a))$$