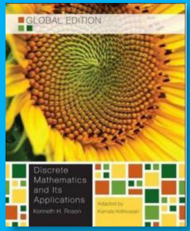
ITP20002-01 Discrete Mathematics

Logic and Proofs

31 August 2018
4 September 2018
13 September 2018



- Taken from the instructor's resource of *Discrete Mathematics and Its Applications*, 7/e
- Edited by Shin Hong hongshin@handong.edu

Chapter 1. Logic and Proofs

- Propositional logic (1.1, 1.2)
- Logical equivalence and satisfiability (1.3)
- Predicate logic (1.4, 1.5)
- Inference (1.6, 1.7)
- Proof basics (1.8, 1.9)

Logic

- Logic is a way to state arguments and to reason with arguments, clearly and correctly

- A logic system has the syntactic and the semantic aspects
  - syntax: symbolic structure of arguments
  - semantics: a relation between symbolic structures and meaning

Proposition

- A proposition is a declarative sentence that is either true or false
  - 1 + 1 = 2
  - *Vancouver is the capital of Canada*
  - ~~1 + 2 + 3~~
  - ~~x + 1 = 2~~

- The negation of *p* for a proposition *p,* denoted as $\neg p$, is the proposition that is true only when *p* is false.

- A compound proposition is formed from existing propositions using logical operators
  - logical operators: negation, disjunction, conjunction, exclusive-or, implication, etc.
  - propositional variable: a variable that represents a proposition

Conditional Statement

- A conditional statement (or implication) $p \to q$ for propositions p and q is the proposition that is false when p is true and q is false, and $p \to \ q$ is true otherwise
  - if you do not take midterm, then you get F
  - if you are in the Handong campus, you are in Pohang
  - if Juan has a smartphone, then 2 + 3 = 5
  - (2 + 3 = 4) → (1 + 2 = 4)

- The converse of $p \to q$ is $q \to p$.
- The inverse of $p \to q$ is $\neg p \to \neg q$.
- The contrapositive of $p \to q$ is $\neg q \to \neg p$.

Propositional Satisfiability

- A compound proposition p is **satisfiable** if there is an assignment of truth values to the propositional variables that makes p true
  - Such assignment is called as a solution

- A compound proposition p is **unsatisfiable** if p is not satisfiable
  - A unsatisfiable proposition is called as contradiction

- A compound proposition p is **valid** if p is true for all assignments
  - A valid proposition is called as tautology
  - E.g., if x = y, then x = y
  - E.g., *I just want to live while I am alive - Bon Jovi*

Propositional Equivalence

- Two compound propositions p and q are logically equivalent when $p \to q \land q \to p$ is valid

- How to show two propositions are logically equivalent?
  - construct truth table
  - use knowledge of equivalent propositions

Example

- De Morgan's law: $\neg(p \wedge q) \equiv \neg p \vee \neg q$, $\neg(p \vee q) \equiv \neg p \wedge \neg q$

p	q	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

Equivalence	*Name*
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws

Equivalence	*Name*
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

$$
\begin{aligned}
\neg(p \rightarrow q) &\equiv \neg(\neg p \vee q) && \text{by Example 3} \\
&\equiv \neg(\neg p) \wedge \neg q && \text{by the second De Morgan law} \\
&\equiv p \wedge \neg q && \text{by the double negation law}
\end{aligned}
$$

Logic Puzzles

Raymond
Smullyan
(Born 1919)

- An island has two kinds of inhabitants, *knights*, who always tell the truth, and *kna ves*, who always lie.
- You go to the island and meet A and B.
    - A says "B is a knight."
    - B says "The two of us are of opposite types."

Example: What are the types of A and B?

Solution: Let p and q be the statements that A is a knight and B is a knight, respe ctively. So, then $\neg p$ represents the proposition that A is a knave and $\neg q$ that B is a knave.

- If A is a knight, then p is true. Since knights tell the truth, q must also be true. Then $(p \wedge \neg q) \vee (\neg p \wedge q)$ would have to be true, but it is not. So, A is not a knight and ther efore $\neg p$ must be true.
- If A is a knave, then B must not be a knight since knaves always lie. So, then both $\neg p$ and $\neg q$ hold since both are knaves.

Sudoku Puzzle as Satisfiability Problem

- A Sudoku puzzle is represented as a 9x9 grid with nine 3x3 subgrids called subgrids
  - each cell has a number in 1 to 9
- The puzzle is solved by assigning a number to each cell so that every row, every column, and every of a block contains each of the 9 numbers.
- Modeling
  - $p(i, j, n)$ holds when row i and column j has n

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

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

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

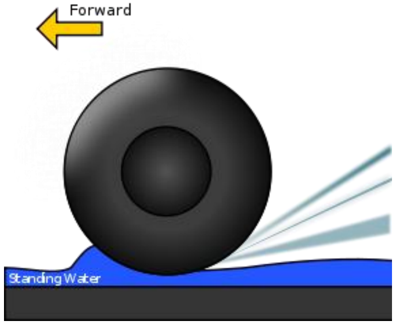
$$\bigwedge_{r=0}^{2} \bigwedge_{s=0}^{2} \bigwedge_{n=1}^{9} \bigvee_{i=1}^{3} \bigvee_{j=1}^{3} p(3r + i, 3s + j, n)$$

$$\bigwedge_{i=1}^{9} \bigwedge_{j=1}^{9} \bigwedge_{n=1}^{9} \bigwedge_{m=1}^{9} (p(i, j, n) \land n \neq m) \rightarrow \neg p(i, j, m)$$

Application

- Logic-based languages (formal languages) are powerful tools for specifying and analyzing software requirements rigorously

- E.g., Lufthansa A320 Airbus accident at Warsaw in 1993 (adopted)
  - Specification
    - Turn on reverse thrust when the airplane is running on runway for landing
  - System Design
    - Define REVERSE_THRUST as ON iff MODE = LANDING and ALTITUDE = 0
    - Define MODE as LANDING iff VELOCITY > 0 and LANDING_GEAR_ANG > 0



Predicate Logic

- A **predicate** is a propositional function over variables
  - once values are assigned to the predicate variables, a predicate becomes a proposition and has a truth value
  - E.g., let $Q(x, y)$ denote $x = y + 3$.
    $Q(4, 1)$ is true and $Q(2, 3)$ is false.

- A **quantification** expresses the extent to which a predicate is true over a range of elements such as "all", "some", "many", "none" represented as a variable.
  - Domain (universe) is the set of all values on which a property is asserted
  - A variable is bound if a quantifier is used on the variable, or free otherwise.
  - Structure

<Quantifier> <Variable w/ domain condition> (<Predicate>)

Quantification

- The universal quantification of $P(x)$, denoted as $\forall x. P(x)$, is the statement that $P(x)$ holds for all values of x in the domain.
  - $\forall$ is called the universal quantifier
  - E.g., $\forall x \in \mathbb{R}.\ \ x^2 \geq 0$

- The existential quantification of $P(x)$, denoted as $\exists x. P(x)$, is the statement that $P(x)$ holds for a value of x in the domain.
  - $\exists$ is called the existential quantifier
  - E.g., $\exists x \in \mathbb{R}.\ \ x^2 = 1$

- The uniqueness quantifier $\exists!$ is to state there is only one value in the domain such that a predicate holds.
  - E.g., $\exists! x\ (x - 1 = 0)$

Propositional Functions

- Propositional functions (i.e., predicate) 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

  $P(-1) \rightarrow P(3)$ **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.

Quantifiers

Charles Peirce (1839-1914)

- 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 <u>every</u> x in the *domain*.
- $\exists x\, P(x)$ asserts $P(x)$ is true for <u>some</u> x in the *domain*.
- The quantifiers are said to bind the variable x in these expressions.

Universal and Existential Quantifiers

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

- $\exists x\, P(x)$ 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)."

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.

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:

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

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

- 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.

Uniqueness Quantifier (*optional*)

- $\exists! x\, P(x)$ means that $P(x)$ is true for <u>one and only one</u> x in the universe of discourse.
- This is commonly expressed in English in the following equivalent ways:
  - "There is a unique x such that $P(x)$."
  - "There is one and only one x such that $P(x)$"
- Examples:
  1. If $P(x)$ denotes "$x + 1 = 0$" and U is the integers, then $\exists! x\, P(x)$ is true.
  2. But if $P(x)$ denotes "$x > 0$," then $\exists! x\, P(x)$ is false.
- The uniqueness quantifier is not really needed as the restriction that there is a unique x such that $P(x)$ can be expressed as:

$$\exists x\, (P(x) \land \forall y\, (P(y) \rightarrow y = x))$$

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 $\forall x\, (S(x) \rightarrow J(x))$.

$\forall x\, (S(x) \land J(x))$ 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 1: But if *U* is all people, then translate as $\exists x\, (S(x) \land J(x))$

$\exists x\, (S(x) \rightarrow J(x))$ 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**: $\forall x \neg\neg S(x) \equiv \forall x\, S(x)$

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 Java course" 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 calculus."

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

Negating Quantified Expressions

- 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.

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 $(x + y = 0)$

Nested Quantifiers

- Example
  - Can we express the uniqueness quantifier ∃! with universal and existential quantifier? How?

- Nested quantifiers have one quantifier within the scope of another quantifier
  - E.g., $\forall x(\exists y(x + y = 0))$
  - E.g., *every man has exactly one wife*
  - E.g., every real number except zero has a multiplication inverse

- Depending on quantification orders, the statements containing the same predicate may have different truth values
  - E.g., $\forall x\big(\exists y(x + y = 0)\big)$, $\exists y\big(\forall x(x + y = 0)\big)$ for that x and y are integers

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: True

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

   Answer: False

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

   Answer: True

Translating Math. 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 Nested Quantifiers into English

- Every student in your school has a computer or has a friend who has a computer

- C(x) is “*x* has a computer,” and
  F(*x*,*y*) is “*x* and *y* are friends”

- $\forall x \ (C(x) \lor \exists y \ (C(y) \land F(x, y)))$