ITP20002-01 Discrete Mathematics

Logic and Proofs

31 August 2018

4 September 2018

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) $p \rightarrow q$ for propositions p and q is the proposition that is false when p is true and q is false, and $p \rightarrow q$ 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 $p \rightarrow q$ is $q \rightarrow p$.
- The inverse of $p \rightarrow q$ is $\neg p \rightarrow \neg q$.
- The contrapositive of $p \rightarrow q$ is $\neg q \rightarrow \neg p$.

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 $p \rightarrow q \wedge q \rightarrow p$ is valid

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

Example

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

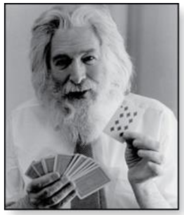
p	q	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

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	

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

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

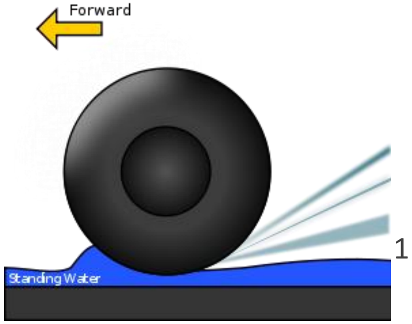
$$\bigwedge_{r=0}^{2} \bigwedge_{s=0}^{2} \bigwedge_{n=1}^{9} \bigvee_{i=1}^{3} \bigvee_{j=1}^{3} p(3r + i, 3s + j, n)$$

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

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., $\exists!\, x\ (\, x - 1 = 0)$