Lecture-2
Compound Proposition

1.1 Propositional Logic

Truth Tables of Compound Propositions

- We can use connectives to build up complicated compound propositions involving any number of propositional variables, then use truth tables to determine the truth value of these compound propositions.
- Example: Construct the truth table of the compound proposition

$$(p \vee \neg q) \longrightarrow (p \wedge q).$$

The Truth Table of $(p \vee \neg q) \rightarrow (p \wedge q)$.

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

1.1 Propositional Logic

Precedence of Logical Operators

- We can use parentheses to specify the order in which logical operators in a compound proposition are to be applied.
- To reduce the number of parentheses, the precedence order is defined for logical operators.

Precedence of Logical Operators.	
Operator	Precedence
$\neg$	1
$\wedge$ $\vee$	2 3
$\rightarrow$ $\leftrightarrow$	4 5

E.g. $\neg p \wedge q = (\neg p) \wedge q$

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

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

Construct the truth table of following compound proposition

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

b) $\neg p \to (q \to r)$

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

d) $(p \to q) \wedge (\neg p \to r)$

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

f) $(\neg p \leftrightarrow \neg q) \leftrightarrow (q \leftrightarrow r)$

1.1 Propositional Logic

Translating English Sentences

- English (and every other human language) is often ambiguous. Translating sentences into compound statements removes the ambiguity.
- Example: How can this English sentence be translated into a logical expression?

  "You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old."

Solution: Let q, r, and s represent "You can ride the roller coaster," "You are under 4 feet tall," and "You are older than 16 years old." The sentence can be translated into:

$$(r \land \neg s) \rightarrow \neg q.$$

Let p and q be the propositions

p : I bought a lottery ticket this week.

q : I won the million dollar jackpot.

Express each of these propositions as an English sentence.

a) $\neg p$

b) $p \vee q$

c) $p \to q$

d) $p \wedge q$

e) $p \leftrightarrow q$

f) $\neg p \to \neg q$

g) $\neg p \wedge \neg q$

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

1.1 Propositional Logic

- Example: How can this English sentence be translated into a logical expression?

  "You can access the Internet from campus only if you are a computer science major or you are not a freshman."

Solution: Let a, c, and f represent "You can access the Internet from campus," "You are a computer science major," and "You are a freshman." The sentence can be translated into:

$$a \rightarrow (c \vee \neg f).$$

Let p and q be the propositions

p : You drive over 65 miles per hour.

q : You get a speeding ticket.

Write these propositions using p and q and logical connectives (including negations).

a) You do not drive over 65 miles per hour.

b) You drive over 65 miles per hour, but you do not get a speeding ticket.

c) You will get a speeding ticket if you drive over 65 miles per hour.

d) If you do not drive over 65 miles per hour, then you will not get a speeding ticket.

e) Driving over 65 miles per hour is sufficient for getting a speeding ticket.

f) You get a speeding ticket, but you do not drive over 65 miles per hour.

g) Whenever you get a speeding ticket, you are driving over 65 miles per hour.

1.1 Propositional Logic

Logic and Bit Operations

- Computers represent information using bits.
- A **bit** is a symbol with two possible values, 0 and 1.
- By convention, 1 represents T (true) and 0 represents F (false).
- A variable is called a Boolean variable if its value is either true or false.
- Bit operation – replace true by 1 and false by 0 in logical operations.

Table for the Bit Operators *OR*, *AND*, and *XOR*.

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

1.1 Propositional Logic

DEFINITION 7

A *bit string* is a sequence of zero or more bits. The *length* of this string is the number of bits in the string.

- Example: Find the bitwise *OR*, bitwise *AND*, and bitwise *XOR* of the bit string 01 1011 0110 and 11 0001 1101.

Solution:

01 1011 0110

11 0001 1101

----

11 1011 1111 bitwise *OR*
01 0001 0100 bitwise *AND*
10 1010 1011 bitwise *XOR*

1.1 Propositional Logic

- Other conditional statements:
  - Converse of $p \longrightarrow q$: $q \longrightarrow p$
  - Contrapositive of $p \longrightarrow q$: $\neg q \longrightarrow \neg p$
  - Inverse of $p \longrightarrow q$: $\neg p \longrightarrow \neg q$

What are the contrapositive, the converse, and the inverse of the conditional statement

"**The home team wins whenever it is raining?"**

Solution: Because "*q* whenever *p*" is one of the ways to express the conditional statement

$p \rightarrow q$, the original statement can be rewritten as

"If it is raining, then the home team wins."

Consequently**, the contrapositive** of this conditional statement is

"If the home team does not win, then it is not raining."

The converse is

"If the home team wins, then it is raining."

The inverse is

"If it is not raining, then the home team does not win."

Only the contrapositive is equivalent to the original statement.

Solve the question

State the converse, contrapositive, and inverse of each of these conditional statements.

a) If it snows today, I will ski tomorrow.
b) I come to class whenever there is going to be a quiz.
c) A positive integer is a prime only if it has no divisors other than 1 and itself.

Thank You