Department Of Computer Science, CUI Lahore Campus

CSC102 - Discrete Structures

By

Mahwish Waqas

Lecture Outline

- Applications of Propositional Logic
- Logic and Bit Operations
- Logical Equivalence

Applications of Propositional Logic

- Translating English sentences (Formalization)
- System Specifications
- Boolean Searches
- Logic circuits

Translating English Sentences

- Steps to convert an English sentence to a statement in propositional logic
  - Identify atomic propositions and represent using propositional variables.
  - Determine appropriate logical connectives

Translating English Sentences

- "I have neither given nor received help on this exam"

  Let p = I have given help on this exam

  q = I have received help on this exam

- Rephrase: It is not the case that either I have given or received help on this exams

Translating English Sentences

- "I have neither given nor received help on this exam"

  Let p = I have given help on this exam

  q = I have received help on this exam

  $\neg p \land \neg q$

- Rephrase: It is not the case that either I have given or received help on this exams

  $\neg(p \lor q)$

Translating English Sentences

- “If I go to Harry’s or to the country, I will not go shopping.”

Translating English Sentences

- “If I go to Harry’s or to the country, I will not go shopping.”

  - Let p = I go to Harry’s
  - q = I go to the country.
  - r = I will go shopping.

Translating English Sentences

- "If I go to Harry's or to the country, I will not go shopping."
  - Let *p* = I go to Harry's
  - q = I go to the country.
  - *r* = I will go shopping.
- If *p* or *q* then not *r*

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

Translating English Sentences

- Let p = It is below freezing
  q = It is snowing

a) It is below freezing and it is snowing
b) It is below freezing but not snowing
c) It is not below freezing and it is not snowing
d) It is either snowing or below freezing (or both)
e) If it is below freezing, it is also snowing
f) It is either below freezing or it is snowing (not both), but it is not snowing if it is below freezing
g) That it is below freezing is necessary and sufficient for it to be snowing

Translating English Sentences

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

Translating English Sentences

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

- Let a = You can access the Internet from campus
  c = You are a computer science major
  and f = You are a freshman” respectively

Translating English Sentences

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

- Let a = You can access the Internet from campus
  c = You are a computer science major
  and f = You are a freshman” respectively

- a only if c or not f

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

Exercise

- Let *p* and *q* be the propositions "The election is decided" and "The votes have been counted," respectively. Express each of these compound propositions as an English sentence.

1. $\neg p$
2. $p \vee q$
3. $\neg p \wedge q$
4. $q \rightarrow p$
5. $\neg q \rightarrow \neg p$
6. $\neg p \rightarrow \neg q$
7. $p \leftrightarrow q$
8. $\neg q \vee (p \wedge q)$

System Specifications

- System and Software engineers take requirements in English and express them in a precise specification language based on logic.

- The automated reply cannot be sent when the file system is full

System Specifications

- System and Software engineers take requirements in English and express them in a precise specification language based on logic.

- The automated reply cannot be sent when the file system is full

  p = The automated reply can be sent

  q = The system is full

  $$q \longrightarrow \neg p$$

Consistency

- System specifications should be **consistent**, They should not contain conflicting requirements that could be used to derive a contradiction.

- When specifications are not consistent, there would be no way to develop a system that satisfies all specifications.

- A list of propositions is **consistent** if it is possible to assign truth values to the proposition variables so that each proposition is true.

Determine whether these system specifications are **consistent**:

1. The diagnostic message is stored in the buffer or it is retransmitted.
2. The diagnostic message is not stored in the buffer.
3. If the diagnostic message is stored in the buffer, then it is retransmitted.

- Determine whether these system specifications are consistent:

1. The diagnostic message is stored in the buffer or it is retransmitted.
2. The diagnostic message is not stored in the buffer.
3. If the diagnostic message is stored in the buffer, then it is retransmitted.

- p = The diagnostic message is stored in the buffer
- q = The diagnostic message is retransmitted
- 1. $p \vee q$ 2. $\neg p$ 3. $p \rightarrow q$

1. $\boldsymbol{p \vee q}$ 2. $\boldsymbol{\neg p}$ 3. $\boldsymbol{p \rightarrow q}$

Reasoning

- An assignment of truth values that makes all three specifications true must have *p* false to make $\neg p$ true.
- Because we want $p \vee q$ to be true but p must be false, *q* must be true.
- Because $p \rightarrow q$ is true when p is false and q is true
- we conclude that these specifications are **consistent**
- Let us do it with **truth table** now

- Is it remain consistent if the specification

“The diagnostic message is not retransmitted” is added?

p: The diagnostic message is stored in the buffer

q: The diagnostic message is retransmitted

1. $\boldsymbol{p} \vee \boldsymbol{q}$ 2. $\neg \boldsymbol{p}$ 3. $\boldsymbol{p} \rightarrow \boldsymbol{q}$

- Is it remain consistent if the specification

“The diagnostic message is not retransmitted” is added?

p: The diagnostic message is stored in the buffer

q: The diagnostic message is retransmitted

1. $p \vee q$ 2. $\neg p$ 3. $p \rightarrow q$
4. $\neg q$

Inconsistent

Logic and Bit Operations

- Computer represents information using bits. A bit is a symbol with two possible values, namely, 0 (zero) and 1 (one).
- 1 represents T (true) and 0 represents (false).
- A variable is called a Boolean Variable if its value is either true or false.
- A Boolean Variable can be represented by a bit.
- A bit string is a sequence of zero or more bits. Length of the string is the number of bits.
    - 1 0101 0011 is bit string of length 9.

Logic and Bit Operations

- We can then do operations on these bit strings.
  - Each column is its own bit operation

Operations	Operator	Bit String1	Bit String 2	Result
Bitwise XOR	$\oplus$	0101 1010	1011 0100	1110 1110
Bitwise OR	$\vee$	0101 1010	1011 0100	1111 1110
Bitwise AND	$\wedge$	0101 1010	1011 0100	00010000

Propositional Equivalence

- An important type of step used in a mathematical argument is the replacement of a statement with another statement with the same truth value

- Propositional Equivalence is extensively used in the construction of mathematical arguments.

Tautology and Contradiction

- A compound proposition that is always true, no matter what the truth values of the propositional variables that occur in it, is called a *tautology. A* compound proposition that is always false is called a *contradiction.*

p	$\neg p$	$p \lor \neg p$	$p \land \neg p$
T	F	T	F
F	T	T	F

- Show that $(p \land q) \rightarrow p$ is a tautology.

Logical Equivalence

- Compound propositions that have the same truth values in all possible cases are called **logically equivalent**.

- The compound propositions p and q are called *logically equivalent* if $p \leftrightarrow q$ is a tautology.

- The notation $p \equiv q$ denotes that p and q are logically equivalent.

Logical Equivalence

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

p	q	$\neg p$	$\neg p \lor q$
T	T	F	T
T	F	F	F
F	T	T	T
F	F	T	T

Logical Equivalence

- Converse

  The proposition $q \rightarrow p$ is **converse** of $p \rightarrow q$.

- Contrapositive

  The **contrapositive** of $p \rightarrow q$ is the proposition $\neg q \rightarrow \neg p$.

- Inverse

  The proposition $\neg p \rightarrow \neg q$ is called the **inverse** of $p \rightarrow q$.

Logical Equivalence

				Implication	Inverse	Converse	Contrapositive
p	**q**	**¬p**	**¬q**	**p → q**	**¬p → ¬q**	**q → p**	**¬q → ¬p**
T	T	F	F	T	T	T	T
T	F	F	T	F	T	T	F
F	T	T	F	T	F	F	T
F	F	T	T	T	T	T	T

Logical Equivalence

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
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distribution Laws

Logical Equivalence

Equivalence	Name
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	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

Logical Equivalence

- Distributive: $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

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

Logical Equivalence involving Implication

Logical Equivalence involving Implication
$p \rightarrow q \equiv \neg p \vee q$
$p \rightarrow q \equiv \neg q \rightarrow \neg p$
$p \vee q \equiv \neg p \rightarrow q$
$p \wedge q \equiv \neg(p \rightarrow \neg q)$
$\neg(p \rightarrow q) \equiv p \wedge \neg q$
$(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$
$(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$
$(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$
$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$

Logical Equivalence involving Bi-conditional

Logical Equivalence involving Bi-conditional
$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Proof using Logical Equivalence

- Show that $\neg(p \wedge \neg q) \vee q \equiv \neg p \vee q$ is logically equivalent.

$\neg(p \wedge \neg q) \vee q$	
$\equiv (\neg p \vee \neg\neg q) \vee q$	DeMorgan's
$\equiv (\neg p \vee q) \vee q$	Double negation
$\equiv \neg p \vee (q \vee q)$	Associative
$\equiv \neg p \vee q$	Idempotent

-

Proof using Logical Equivalence

Show that $(p \wedge q) \rightarrow q$ is a Tautology.

Proof:

$(p \wedge q) \rightarrow q$	
$\equiv \neg(p \wedge q) \vee q$	Implication
$\equiv (\neg p \vee \neg q) \vee q$	De Morgan
$\equiv \neg p \vee (\neg q \vee q)$	Associative
$\equiv \neg p \vee T$	Negation
$\equiv T$	Dominations

Proof using Logical Equivalence

- Show that $[p \wedge (p \rightarrow q)] \rightarrow q$ is a tautology.

$[p \wedge (p \rightarrow q)] \rightarrow q$	
$\equiv [p \wedge (\neg p \vee q)] \rightarrow q$	Substitution for $\rightarrow$
$\equiv [(p \wedge \neg p) \vee (p \wedge q)] \rightarrow q$	Distributive
$\equiv [\,F \vee (p \wedge q)] \rightarrow q$	Negation
$\equiv (p \wedge q) \rightarrow q$	Identity
$\equiv \neg(p \wedge q) \vee q$	Substitution for $\rightarrow$
$\equiv (\neg p \vee \neg q) \vee q$	DeMorgan's
$\equiv \neg p \vee (\neg q \vee q\,)$	Associative
$\equiv \neg p \vee T$	Negation
$\equiv T$	Domination

Proof using Logical Equivalence

Show that $\neg(p \vee (\neg p \wedge q)) \equiv \neg p \wedge \neg q$ is logically equivalent.

$L.H.S = \neg(p \vee (\neg p \wedge q))$

$\equiv \neg p \wedge \neg(\neg p \wedge q)$ DeMorgan's Law

$\equiv \neg p \wedge (\neg(\neg p) \vee \neg q)$ DeMorgan's Law

$\equiv \neg p \wedge (p \vee \neg q)$ Double Negation Law

$\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q)$ Distributive Law

$\equiv F \vee (\neg p \wedge \neg q)$ Negation Law

$\equiv (\neg p \wedge \neg q) \vee F$ Commutative Law

$\equiv \neg p \wedge \neg q$ Identity Law

$= R.H.S$

Chapter Reading

- ***Chapter 1***, Kenneth H. Rosen, Discrete Mathematics and Its Applications, Section 1.2,1.3

Chapter Exercise (For Practice)

- Section 1.2: Question # 1, 2, 3, 4, 7, 8, 9, 10, 11 ,12
- Section 1.3: Question # 1, 6, 7, 8, 9, 10, 11, 12