Discrete Strutures

Dr. Syed Faisal Bukhari
Associate Professor
Department of Data Science, Faculty of Computing and Information Technology, University of the Punjab

Text book

Discrete Mathematics and Its Application, 7^{th} Edition

Kenneth H. Rosen

References

Chapter 1

1. Discrete Mathematics and Its Application, 6^{th} Editition

by

Kenneth H. Rose

2. Discrete Mathematics with Applications

by

Thomas Koshy

More keywords

- **Whenever** means **if**
- **But** means **and**
- p **but** q means p **and** q
- **neither** p **nor** q means $\neg$p **and** $\neg$q
- **either** p **or** q means p$\vee$q

More keywords

Some programming languages use statements of the form ***“r* unless *s*”** to mean that as long as *s* does not happen, then *r* will happen. More formally:

Definition:

If *r* and *s* are statements,

r **unless** *s* means if $\neg$ *s* then *r*.

How to remember **unless:** Replace **unless** with **if not**

More keywords

Example 1: Zarar won’t go to the party **unless** Abdullah goes to the party.

means

If Abdullah **doesn’t** go the party **then** Zarar won’t go to the party.

Example 2: Unless you pay us one million dollar, you'll never see your pet goldfish again.

If you don’t pay us one million dollar then you’ill never see your pet goldfish again.

Question: Are these system specifications consistent?

"The router can send packets to the edge system only if it supports the new address space. For the router to support the new address space it is necessary that the latest software release be installed. The router can send packets to the edge system if the latest software release is installed. The router does not support the new address space."

Different ways to express conditional statement (implication)

1. "if p, then q"
2. "if p, q "
3. "p is sufficient for q"
4. "q if p"
5. "q when p"
6. "a necessary condition for p is q"

Different ways to express conditional statement (implication)

7. ''q unless $\neg$p''

8. ''p implies q''

9. ''p only if q''

10''a sufficient condition for q is p''

11''q whenever p''

12''q is necessary for p''

13''q follows from p''

Common ways to express bi-implications

"p is necessary and sufficient for q "

"if p then q , and conversely"

"p iff q"

The last way of expressing the biconditional statement $p \leftrightarrow q$ uses the abbreviation **"iff"** for "if and only if."

Note that **$p \leftrightarrow q$** has exactly the same truth value **as $(p \rightarrow q) \wedge (q \rightarrow p)$**.

1. **The router can send packets to the edge system only if it supports the new address space.**

2. **For the router to support the new address space it is necessary that the latest software release be installed.**

3. **The router can send packets to the edge system if the latest software release is installed.**

4. **The router does not support the new address space.**

- ''if p, then q'' ≡ ''p only if q''
- ''if p, then q'' ≡ ''q is necessary for p'‘
- **p** is a necessary condition for **q** means "if not **p** then not **q**."
- ''if p, then q'' ≡ ''q if p''

Let

p: “The router can send packets to the edge system”

q: “It supports the new address space”

r: “The latest software release be installed”

The router can send packets to the edge system only if it supports the new address space.

"if p, then q" ≡ "p only if q"

1. $p \rightarrow q$

For the router to support the new address space it is necessary that the latest software release be installed.

"if p, then q" ≡ "q is necessary for p"

2. **$r \rightarrow q$**

p: “The router can send packets to the edge system”

q: “It supports the new address space”

r: “the latest software release be installed”

The router can send packets to the edge system if the latest software release is installed.

"if p, then q" ≡ "q if p"

3. $r \rightarrow p$

The router does not support the new address space.

4. $\neg q$

The system specifications

1. $p \to q$
2. $r \to q$
3. $r \to p$
4. $\neg q$

- $\neg q$ is true then q must be false using (4) .
- q is false then p must be false which makes (1) true.
- q is false then r must be false which makes (2) true.
- r is false then p is false which makes (3) true.

✓ **The system is consistent.**

Alternative method

Question: Are these system specifications consistent?

“The router can send packets to the edge system only if it supports the new address space. For the router to support the new address space it is necessary that the latest software release be installed. The router can send packets to the edge system if the latest software release is installed, The router does not support the new address space.”

- "if p, then q" ≡ "p only if q"
- **p** is a necessary condition for **q** means "if not **p** then not **q**."
- "if p, then q" ≡ "q if p"

Let

p: “The router can send packets to the edge system”

q: “It supports the new address space”

r: “The latest software release be installed”

The router can send packets to the edge system only if it supports the new address space.

"if p, then q" ≡ "p only if q"

1. $p \rightarrow q$

For the router to support the new address space it is necessary that the latest software release be installed.

- "if p, then q" ≡ **p** is a necessary condition for **q** means "if not **p** then not **q**."

2. $q \rightarrow r$

p: “The router can send packets to the edge system”

q: “It supports the new address space”

r: “the latest software release be installed”

The router can send packets to the edge system if the latest software release is installed.

"if p, then q" ≡ "q if p"

3. $r \rightarrow p$

The router does not support the new address space.

4. $\neg q$

The system specifications

1. $p \to q$
2. $q \to r$
3. $r \to p$
4. $\neg q$

- $\neg q$ is true the q must be false using (4) .
- q is **false** then p must be **false** to make (1) true.
- p is **false** then r must be **false** using (3).
- r is false then q is false using (1) which makes (2) true.
- ✓ The system is consistent.

Logic and Bit Operations

- Computers represent information using **bits**. A bit is a symbol with two possible values, namely, **0 (zero)** and **1 (one). 1** represents **T (true), 0** represents **F (false).**

- A **variable** is called a **Boolean variable** if its value is **either true or false**. Consequently, **a Boolean variable** can be represented using a **bit**.

- Computer bit **operations** correspond to the **logical connectives**.

- We will also use the notation **OR, AND,** and **XOR** for the operators $\vee$, $\wedge$, and $\oplus$

The Bit Operators OR, AND and XOR

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

Bit string

- 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

1 0 1 0 1 00 1 1 is a bit string of **length nine**.

The Bit Operators OR, AND and XOR

Example Find the **bitwise OR, bitwise AND,** and **bitwise XOR** of the bit strings **0 1 1 0 1 1 0 1 1 0** and **1 1 000 1 1 1 0 1** .

0 1 1 0 1 1 0 1 1 0 and **1 1 000 1 1 1 0 1**

0 1 1 0 1 1 0 1 1 0
1 1 0 0 0 1 1 1 0 1

1 1 1 0 1 1 1 1 1 1 **bitwise OR**

0 1 0 0 0 1 0 1 0 0 **bitwise AND**

1 0 1 0 1 0 1 0 1 1 **bitwise XOR**

Precedence of Logical Operators

Precedence of Logical Operators	
Operator	**Precedence**
¬	1
∧	2
∨	3
→	4
↔	5

Propositional Equivalences

- **Tautology**: A compound proposition that is **always true**, no matter **what the truth values of the propositions** that occur in it, is called a **tautology**.

- **Contradiction**: A compound proposition that is **always false** is called a **contradiction**.

- **Contingency:** A compound proposition that is **neither a tautology nor a contradiction** is called a **contingency**.

Propositional Equivalences

Examples of a Tautology and a Contradiction.			
p	¬p	p V ¬p	p Λ ¬p
T	F	T	F
F	T	T	F

Logical Equivalences

- **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**.

- **Remark:** The symbol **$\equiv$** is not a **logical connective** and $p \equiv q$ is not a compound proposition but rather is the statement that **$p \leftrightarrow q$** is a **tautology**. The symbol **$\Leftrightarrow$** is sometimes used instead of **$\equiv$** to denote logical equivalence

Example: Show that **$\neg(p \vee q)$** and **$\neg p \wedge \neg q$** are **logically equivalent.**

¬(p V q) and ¬p Λ ¬q are logically equivalent.

Truth Tables for ¬(p Λ q) and ¬p V ¬q						
p	q	¬p	¬q	¬p Λ¬q	p V q	¬(p V q)
T	T	F	F	F	T	F
T	F	F	T	F	T	F
F	T	T	F	F	T	F
F	F	T	T	T	F	T

Example: Show that $p \vee (q \wedge r)$ and $(p \vee q) \wedge (p \vee r)$ are **logically equivalent**.

p V (q Λ r) and (p V q) Λ (p V r) are logically equivalent

p	q	r	q Λ r	p V (q Λ r)	p V q	p V r	(p V q) Λ (p V 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

Example: Show that $p \rightarrow q$ and $\neg p \vee q$ are logically equivalent.

p→q and ¬p v q are logically equivalent

Truth Table of p→ q and ¬p v q				
p	q	**¬p**	¬p v q	p→ q
T	T	**F**	T	T
T	F	**F**	F	F
F	T	**T**	T	T
F	F	**T**	T	T

Suggested Readings

1.3 Propositional Equivalences