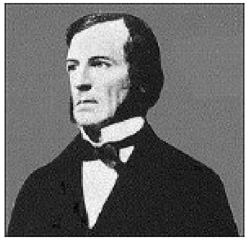
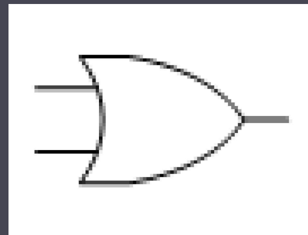


Introduction to Mathematics for Computing

Logic I

Logic I



- Introduction
- Mathematical Statements
- Logic Operations and Truth Tables
- Summary

Introduction

- Syllogism
  - Let's consider some statements

All NCI students are human

All humans are mortal

Can we reach a conclusion regarding the mortality of NCI students?

Aristotle (384BC – 322BC)

Aristotle defines the **syllogism** as, "...a discourse in which certain (specific) things having been supposed, something different from the things supposed results of necessity because these things are so.“ https://en.wikipedia.org/wiki/Syllogism

Introduction

- Syllogism

  - We are given two statements which we refer to as ***premises***

All NCI students are human — *Premise*

All humans are mortal — *Premise*

All NCI students are mortal — *Conclusion*

Introduction

- Syllogism
  - Consider the following premises

All elephants are yellow	*Premise*
All yellow objects are capable of flying	*Premise*
All elephants are capable of flying	*Conclusion*

Introduction

- Syllogism
  - Consider the following premises

Some elephants are grey — *Premise*

Some grey objects are capable of flying — *Premise*

~~Some elephants are capable of flying~~ — *Conclusion*

Mathematical Statements

- Propositional Logic

  - A lot of mathematics involves formulating statements about mathematical objects (numbers, lines, sets, relations functions etc.) and then determining whether or not those statements are **true** or **false**.

  - So it is important that we have some idea of what constitutes a mathematical statement …

We will begin by considering the idea of a **proposition**

Mathematical Statements

- Propositional Logic

  - We can think of a **proposition** as being a statement which is either true or false (but not both)

  - Propositions are the fundamental objects in propositional logic

Definition: A ***proposition*** is a sentence that is either true or false but not both. The ***truth value*** is the value of the proposition (i.e., whichever of these is the case – either true or false).

Mathematical Statements

- Propositional Logic

  - Consider the following:

    i) $4 + 5 = 9$

    ii) $8 + 5 = 56$

    iii) Today is Monday

    iv) Dublin is the capital city of Norway

    v) 29 is a prime number

    vi) Every integer greater than 2 may be written as the sum of two prime numbers

    vii) n is a prime number

    viii) π is a strange number

See Wikipedia entry on the Goldbach Conjecture

Logical Operations and Truth Tables

- Logical Operations
  - We will typically encounter mathematical statements that are built up out of simpler statements
  - The simpler statements are joined together using ***logical operators***
  - The truth or falsehood of a mathematical statement will be dependent on the truth or falsehood of the component statements and the way in which the component statements are joined together using the logical operators
  - **Note:** logical operators are also referred to as ***logical connectives***

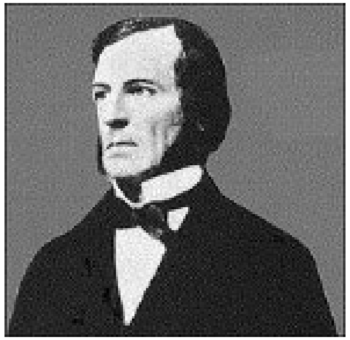
Logical Operations and Truth Tables

- Logical Operators
  - We will consider 3 logical operators

AND

OR

NOT



Boole, G., 1854, *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities.*

https://en.wikipedia.org/wiki/Boole

Logical Operations and Truth Tables

- An *Algebra of Logic*
  - In mathematics we typically use the letters x, y, z to denote variables that can be replaced by real numbers. These variables can be combined using the operators $+, -, \times, \div, =$.
  - In logic, we call a sentence a proposition if it is unambiguously true or false. A propositional variable is simply a variable name that stands for a proposition. The letters p, q, r are typically used to denote propositional variables.

Definition: Any propositional variable alone is a ***formal proposition***.

Logical Operations and Truth Tables

- An *Algebra of Logic*

  - Example:

Let p denote the statement "Today is Monday".
Let q denote the statement "It is raining".

Both p and q are formal propositions.

They can be combined using *logical connectives* to form compound statements.

We write propositions as follows:

p : Today is Monday
q : It is raining

Logical Operations and Truth Tables

- AND
  - We use the **and** logical connective when we want to assert that two statements are both true
  - Example:

We assert that	*The value of* π *is between* 3 *and* 4
or in symbols	$3 < \pi < 4$
then this means that	$\pi > 3$ **and** $\pi < 4$

This is called the ***conjunction*** of the two statements "$\pi > 3$" and "$\pi < 4$"

Logical Operations and Truth Tables

- AND

  - We use the symbol $\wedge$ for the **and** logical connective

Definition: Given formal propositions $\boldsymbol{p}$ and $\boldsymbol{q}$, the compound statement $\boldsymbol{p} \wedge \boldsymbol{q}$ is a formal proposition known as the ***conjunction*** and is read as "***p* and *q***".

$$\wedge$$

Logical Operations and Truth Tables

- Truth Table for AND
  - Given any two propositions $\boldsymbol{p}$ and $\boldsymbol{q}$ each has two possible truth values
  - Taken together there are then 4 possible combinations of truth values for $\boldsymbol{p}$ and $\boldsymbol{q}$
  - For each specific combination of truth values for $\boldsymbol{p}$ and $\boldsymbol{q}$ we can specify the truth value associated with $\boldsymbol{p} \wedge \boldsymbol{q}$
  - We can create a ***truth table*** specifying the 4 combinations of truth values for $\boldsymbol{p}$ and $\boldsymbol{q}$ and the associated truth value for $\boldsymbol{p} \wedge \boldsymbol{q}$

$$\boxed{\wedge}$$

Logical Operations and Truth Tables

- Truth Table for AND

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Truth Table for Conjunction

$\wedge$

Logical Operations and Truth Tables

- OR

  - We use the **or** logical connective when we want to assert either of two statements are true or both statements are true. This is called the ***inclusive*** use of 'or'.

  - Example:

Consider the statement $a = 0$ ***or*** $b = 0$

Then the statement is true if $a = 0$ (regardless of the value of b) and is also true if $b = 0$ (regardless of the value of a).

The statement is also true if both $a = 0$ and $b = 0$ are true.
This is called the ***disjunction*** of the two statements "$a = 0$" and "$b = 0$"

Logical Operations and Truth Tables

- OR

  - We use the symbol ∨ for the **or** logical connective

Definition: Given formal propositions $\boldsymbol{p}$ and $\boldsymbol{q}$, the compound statement $\boldsymbol{p} \vee \boldsymbol{q}$ is a formal proposition known as the ***disjunction*** and is read as "***p* or *q***".

$$\vee$$

Logical Operations and Truth Tables

- Truth Table for OR
  - We can create a ***truth table*** specifying the 4 combinations of truth values for $\boldsymbol{p}$ and $\boldsymbol{q}$ and the associated truth value for $\boldsymbol{p} \vee \boldsymbol{q}$

$\boxed{\vee}$

Logical Operations and Truth Tables

- Truth Table for OR

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Truth Table for Disjunction

$\vee$

Logical Operations and Truth Tables

- NOT
  - The **not** logical connective is used when we wish to ***negate*** a proposition.
  - The negation of a proposition is true when the original proposition is false and it is false when the original proposition is true.
  - Example:

Consider the statement $a \neq 0$

Then the statement is true if $a = 0$ is false.

This is called the ***negation*** of the statement "$a = 0$"

Logical Operations and Truth Tables

- NOT
  - We use either the symbol $\neg$ or the over-bar symbol $\overline{}$ for the **not** logical connective

Definition: Given the formal proposition $\boldsymbol{p}$, the compound statement $\overline{\boldsymbol{p}}$ is a formal proposition known as the ***negation*** of $\boldsymbol{p}$ and is read as "***not p***".

$\neg$ $\overline{}$

Logical Operations and Truth Tables

- Truth Table for NOT

  - We can create a ***truth table*** specifying the truth values for $\boldsymbol{p}$ and the associated truth value for $\overline{\boldsymbol{p}}$

$\neg$ $\overline{\ }$

Logical Operations and Truth Tables

- Truth Table for NOT

p	$\overline{p}$
T	F
F	T

Truth Table for Negation

$\neg$

$\overline{}$

Summary

- Logic
- Mathematical Statements
- Logical Connectives
- Truth Tables

Bonus Example

- The following example is an application of propositional logic (via Boolean Satisfiability) to a real-world problem.

- This will NOT be on the exam.

Boolean Satisfiability (SAT)

- Video: Boolean Satisfiability Problem

- ALL variables are binary – TRUE or FALSE

$$x_i \in \{True, False\}$$

- Three operators – NOT (negation $\neg$) , AND (or conjunction $\wedge$) , OR (or disjunction $\vee$)

$$\neg x_1$$
$$x_1 \wedge x_2$$
$$x_1 \vee x_2$$

- A clause is a combination of variables with operators:

$$(\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2)$$

We need an assignment to the variables that satisfies this clause:

SAT Example – Class Scheduling

We need to schedule three classes…:

Maths (M), English (E), and Science (S)

…into two available time slots:

Morning (AM) and Afternoon (PM).

Define 3 Boolean variables:

- M: Maths is scheduled in the morning (**true**) or afternoon (**false**)
- E: English is scheduled in the morning (**true**) or afternoon (**false**)
- S: Science is scheduled in the morning (**true**) or afternoon (**false**)

SAT Example – Class Scheduling

Constraints:

1. Only two classes can be in the morning, (i.e. at least one class must be in the afternoon):

$$\neg M \vee \neg E \vee \neg S$$

2. Maths and English cannot be at the same time.

$$(M \vee E) \wedge (\neg M \vee \neg E)$$

3. Science must be in the morning.

$$S$$

Solving using a SAT solver or algorithm gives the possible solution:

- $M = True$ (morning)
- $E = False$ (afternoon)
- $S = True$ (morning)

Class	Morning	Afternoon
Maths	X	
English		X
Science	X	

- This assignment satisfies all constraints.

SAT Applications

1. Cryptography:
   - Used to analyze and break cryptographic algorithms by encoding them as SAT problems.
   - Helps in detecting weaknesses in hash functions and ciphers.

2. Artificial Intelligence & Planning
   - Solves planning problems by encoding action sequences as logic constraints.
   - Used in automated reasoning and knowledge representation.

3. Hardware & Software Verification
   - Checks circuit correctness by verifying logic gates and states.
   - Finds software bugs through symbolic execution and constraint checking.

4. Scheduling & Timetabling
   - Assigns tasks to time slots/resources under constraints (e.g., exams, jobs).
   - Ensures feasible, conflict-free schedules using logical constraints.