Mathematical concepts for computer science

Mathematical concepts for computer science

Course Description

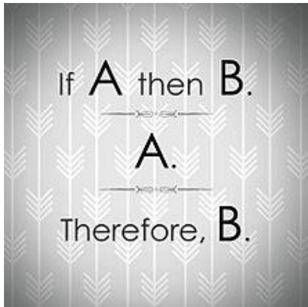
- This course introduces the **study of mathematical structures** that are fundamentally discrete in nature.

- It introduces **linear algebra, graph theory and probability**.

- The course is intended to cover the main aspects which are useful in **studying, describing and modeling** of objects and problems in the context of **computer algorithms and programming languages**.

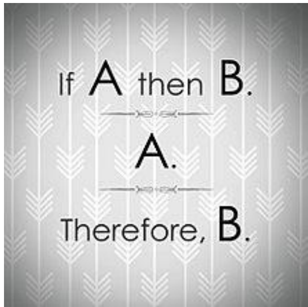
The Foundations: Logic and Proofs

- **Logic** is the systematic study of the **form** of **valid inference**, and the most general laws of **truth**.
- The **logical form** attempts to formalize a possibly ambiguous statement into a statement with a **precise**, **unambiguous** logical interpretation with respect to a formal system.



The Foundations: Logic and Proofs

- **Mathematical logic** is a subfield of mathematics exploring the applications of formal logic to mathematics.
- The **rules of logic** specify the meaning of mathematical statements.
- Eg: “For every positive integer n, the sum of the positive integers not exceeding n is n(n+1)/2”



The Foundations: Logic and Proofs

- **Logic** is the basis of all mathematical reasoning, and of all automated reasoning.

- **Applications**
  - Design of computing machines
  - Specification of the systems
  - Artificial intelligence
  - Computer programming etc..

The Foundations: Logic and Proofs

- What makes up a correct mathematical argument- Proof.
- Once we proved mathematical statement is true-Theorem.

The Foundations: Logic and Proofs

- A **Theorem** is a **major** result

- A **Corollary** is a theorem that follows on from **another theorem**

- A **Lemma** is a **small** result (less important than a theorem)

The Foundations: Logic and Proofs

Theorem

- If m and n are any two whole numbers and

$a = m^2 - n^2$

$b = 2mn$

$c = m^2 + n^2$

then $a^2 + b^2 = c^2$

Proof:

The Foundations: Logic and Proofs

Theorem

- If m and n are any two whole numbers and

$a = m^2 - n^2$

$b = 2mn$

$c = m^2 + n^2$

then $a^2 + b^2 = c^2$

Proof:

- $\mathbf{a^2 + b^2} = (m^2 - n^2)^2 + (2mn)^2$

$= m^4 - 2m^2n^2 + n^4 + 4m^2n^2$

$= m^4 + 2m^2n^2 + n^4$

$= (m^2 + n^2)^2 \mathbf{= c^2}$

The Foundations: Logic and Proofs

Corollary

a, b and c, as defined above, are a Pythagorean Triple

Proof:

From the Theorem $\mathbf{a^2 + b^2 = c^2}$,

so a, b and c are a Pythagorean Triple

The Foundations: Logic and Proofs

Lemma

If m = 2 and n = 1, then we get the Pythagorean triple 3, 4 and 5

Proof:

If m = 2 and n = 1, then

$a = 2^2 - 1^2 = 4 - 1 = \mathbf{3}$

$b = 2 \times 2 \times 1 = \mathbf{4}$

$c = 2^2 + 1^2 = 4 + 1 = \mathbf{5}$

Small result.

Propositions

A **proposition** is a declarative sentence (ie. A sentence that declares a fact) that is **either true or false** , but not both.

Declarative sentences are simply statements that relay information.

A **declarative sentence** states the facts or an opinion and lets the reader know something specific.

- Washington, D.C., is the capital of the United States of America. (True)
- Toronto is the capital of Canada. (False)
- 1 + 1 = 2. (True)
- 2 + 2 = 3. (False)

Propositions

Whether the following sentences are propositions?

- What time is it?
- Read this carefully.
- $x + 1 = 2$.
- $x + y = z$.

Propositions

Whether the following sentences are propositions?

Not declarative sentences

- What time is it?
- Read this carefully.

Neither true nor false

- $x + 1 = 2.$
- $x + y = z.$

Propositional variables

Propositional variables (or statement variables)

- Variables that represent propositions, just as letters are used to denote numerical variables.
- Conventional letters used for propositional variables are *p, q, r, s, ...*
- The **truth value** of a proposition is true, denoted by **T**, if it is a true proposition, and the truth value of a proposition is false, denoted by **F**, if it is a false proposition.

Propositional calculus or propositional logic

Propositional calculus

- The area of logic that deals with propositions is called the **propositional calculus or propositional logic.**

- It was first developed systematically by the Greek philosopher **Aristotle** more than 2300 years ago.

Compound propositions

Compound propositions

- New propositions, called **compound propositions, are** formed from existing propositions using logical operators.

$$\boldsymbol{p \land q}$$

Negation of the proposition

DEFINITION

Let p be a proposition. The negation of p, denoted by $\neg p$ (also denoted by not p), is the statement

"It is not the case that p."

The proposition $\neg p$ is read "not p."

The truth value of the negation of p, $\neg p$, is the opposite of the truth value of p.

Negation of the proposition

Find the negation of the proposition

“**Michael’s PC runs Linux**”

and express this in simple English.

Negation of the proposition

Find the negation of the proposition

“**Michael’s PC runs Linux**”

and express this in simple English.

Solution:

The negation is

“**It is not the case that Michael’s PC runs Linux**.”

This negation can be more simply expressed as

“**Michael’s PC does not run Linux**.”

Conjunction

DEFINITION

Let **p** and **q** be propositions. The **conjunction of p and q**, denoted by **p ∧ q**, is the proposition "**p and q**." The conjunction p ∧ q is **true when both p and q are true** and is **false otherwise**.

Find the conjunction of the propositions p and q where p is the proposition "**Rebecca's PC has more than 16 GB free hard disk space**" and q is the proposition "**The processor in Rebecca's PC runs faster than 1 GHz**."

Conjunction

Find the conjunction of the propositions p and q where p is the proposition "**Rebecca's PC has more than 16 GB free hard disk space**" and q is the proposition "**The processor in Rebecca's PC runs faster than 1 GHz**."

Rebecca's PC has more than 16 GB free hard disk space, and its processor runs faster than 1 GHz.

Disjunction

DEFINITION

Let **p** and **q** be propositions. The **disjunction of p and q**, denoted by **p V q**, is the proposition "**p or q**." The disjunction **p V q is false** when **both p and q are false** and is **true otherwise**.

"**Students who have taken calculus or computer science can take this class.**" **- inclusive or.**

"**Students who have taken calculus or computer science, but not both, can enroll in this class.**"- **exclusive or.**

Truth table

A **truth table** is a mathematical table used in logic—specifically in connection with Boolean algebra, boolean functions, and propositional calculus.

Truth table sets out the **functional values of logical expressions** on each of their functional arguments, that is, for each combination of values taken by their logical variables.

Truth table

TABLE 1 The Truth Table for the Negation of a Proposition.

p	$\neg p$
T	F
F	T

TABLE 2 The Truth Table for the Conjunction of Two Propositions.

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

TABLE 3 The Truth Table for the Disjunction of Two Propositions.

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

Exclusive or

DEFINITION

Let **p** and **q** be propositions. The **exclusive or of p and q**, denoted by **p ⊕ q**, is the **proposition that is true** when exactly one of p and q is true and is false otherwise.

TABLE 4 The Truth Table for the Exclusive Or of Two Propositions.

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Conditional Statements (implication)

DEFINITION

Let **p** and **q** be propositions. The conditional statement **p → q** is the proposition "**if p, then q**." The conditional statement **p → q is false** when **p is true and q is false**, and **true otherwise**.

In the conditional statement p → q, **p is called the hypothesis** (or antecedent or premise) and **q is called the conclusion** (or consequence).

"**If I am elected, then I will lower taxes**."

TABLE 5 The Truth Table for the Conditional Statement $p \to q$.

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

Conditional Statements (implication)

if 2 + 2 = 4 then ***x := x + 1,*** if *x = 0 before this statement is encountered?*

Conditional Statements (implication)

if 2 + 2 = 4 then *x := x + 1,* if *x = 0 before this statement is encountered?*

Solution:

Because 2 + 2 = 4 is true, the assignment statement x := x + 1 is executed. Hence, x has the value 0 + 1 = 1 after this statement is encountered.

CONVERSE, CONTRAPOSITIVE, AND INVERSE

- The proposition **q → p** is called the **converse** of **p → q**.
- The **contrapositive** of **p → q** is the proposition **¬q →¬p**.
- The proposition **¬p →¬q** is called the **inverse** of **p → q**.

- Write the truth table of converse, contrapositive and inverse

CONVERSE, CONTRAPOSITIVE, AND INVERSE

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

Only the **contrapositive** always has the **same truth value** as **$p \rightarrow q$**.

CONVERSE, CONTRAPOSITIVE, AND INVERSE

When **two compound propositions** always have the **same truth value** we call them **equivalent,** so that a conditional statement and its contrapositive are equivalent.

- The **converse and the inverse of a conditional statement** are also equivalent

CONVERSE, CONTRAPOSITIVE, AND INVERSE

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

“**The home team wins whenever it is raining**?”

$p \rightarrow q$:- “If it is raining, then the home team wins.”

CONVERSE, CONTRAPOSITIVE, AND INVERSE

$p \rightarrow q$:- "If it is raining, then the home team wins."

- **Contrapositive** (**$\neg q \rightarrow \neg p$**) -"If the home team does not win, then it is not raining."
- **Converse** (**$p \rightarrow q$**)-"If the home team wins, then it is raining."
- **Inverse** (**$\neg p \rightarrow \neg q$**)-"If it is not raining, then the home team does not win."

BICONDITIONALS

Let **p** and **q** be propositions. The **biconditional** statement **p ↔ q** is the proposition "**p if and only if q**." The biconditional statement **p ↔ q is true** when **p and q** have the **same truth values**, and is **false otherwise**.

Biconditional statements are also called **bi-implications**.

TABLE 6 The Truth Table for the Biconditional $p \leftrightarrow q$.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

BICONDITIONALS

Let **p** be the statement "**You can take the flight**," and let **q** be the statement "**You buy a ticket**." Then **p ↔ q** is the statement "**You can take the flight if and only if you buy a ticket**."

TABLE 6 The Truth Table for the Biconditional $p \leftrightarrow q$.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Truth Tables of Compound Propositions

Construct the truth table of the compound proposition

$(p \lor \neg q) \rightarrow (p \land q)$.

Truth Tables of Compound Propositions

Construct the truth table of the compound proposition

$(p \lor \neg q) \rightarrow (p \land q)$.

TABLE 7 The Truth Table of $(p \lor \neg q) \rightarrow (p \land q)$.

p	q	$\neg q$	$p \lor \neg q$	$p \land q$	$(p \lor \neg q) \rightarrow (p \land q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F

Precedence of Logical Operators

Priority in importance

$(p \land q \lor r)$ means ?

$p \lor q \rightarrow r$ means ?

TABLE 8
Precedence of Logical Operators.

Operator	*Precedence*
$\neg$	1
$\land$ $\lor$	2 3
$\rightarrow$ $\leftrightarrow$	4 5

Precedence of Logical Operators

Priority in importance

$(p \land q \lor r)$ means $(p \land q) \lor r$

$p \lor q \rightarrow r$ means $(p \lor q) \rightarrow r$

TABLE 8
Precedence of Logical Operators.

Operator	*Precedence*
$\neg$	1
$\land$ $\lor$	2 3
$\rightarrow$ $\leftrightarrow$	4 5

Logic and Bit Operations

- A **bit** is a symbol with two possible values, namely**, 0 (zero) and 1 (one)**.
- A variable is called a **Boolean** variable if its value is either true or false**.**
- Boolean variable can be represented using a **bit.**
- Computer **bit operations** correspond to the **logical connectives**.

Truth Value	*Bit*
T	1
F	0

TABLE 9 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**.
  - **1 01010011** is a bit string of length **nine**
- Find the bitwise **OR**, bitwise **AND**, and bitwise **XOR** of the bit strings **01 1011 0110** and **11 0001 1101**.

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**.
  - **1 01010011** is a bit string of length **nine**
- Find the bitwise **OR**, bitwise **AND**, and bitwise **XOR** of the bit strings **01 1011 0110** and **11 0001 1101**.

01 1011 0110	
11 0001 1101	
11 1011 1111	bitwise *OR*
01 0001 0100	bitwise *AND*
10 1010 1011	bitwise *XOR*

Reference

- **Rosen, Kenneth H., and Kamala Krithivasan. Discrete mathematics and its applications: with combinatorics and graph theory. Tata McGraw-Hill Education, 2016.**