Discrete Mathematics

CMP-200

Abdul Hameed

http://informationtechnology.pk/pucit

abdul.hameed@pucit.edu.pk

Lectures and Notes

- Lectures and presentations will be available online after class.
- There will usually be a low-tech component (whiteboard + marker) which you are responsible for.
- You can download lectures from

  http://informationtechnology.pk/

About Me

- MPhil in Information Technology
- MIT from PUCIT (Bronze Medal)
- Bachelor in Computer Science(Gold Medal)
- Topper in Punjab Public Service Commission Examination for three times
- Teaching experience of more than 19 years
- currently working as Assistant Professor and Head of the Computer Science department in Higher Education Department.

Projects

- Project In-charge at Govt. Degree College Chunian since 2002 to 2008
- Master trainer with Directorate of Staff Development (DSD) for computer IT teachers since 2010
- Key Master Trainer of PBCC(Punajb Boards Committee of Chairman)
- Conducting training on Office Professional with DataTech Distribution Pakistan since 2007
- Instructor at Vocational Training Institute, Chunian since 2008 to 2011

Quick Overview

Discrete Math is essentially that branch of mathematics that does not depend on limits; in this sense, it is the anti-thesis of Calculus. As computers are discrete object operating one jumpy, discontinuous step at a time, Discrete Math is the right framework for describing precisely Computer Science concepts.

Quick Overview

The conceptual center of computer science is the **ALGORITHM**.

Quick Overview

Discrete Math helps provide…

- the machinery necessary for creating sophisticated algorithms
- the tools for analyzing their efficiency
- the means of proving their validity

Quick Overview - Topics

- Logic and Sets
  - Make notions you’re already used to from programming a little more rigorous (operators)
  - Fundamental to all mathematical disciplines
  - Useful for digital circuits, hardware design
- Elementary Number Theory
  - Get to rediscover the old reliable number and find out some surprising facts
  - Very useful in crypto-systems

Quick Overview - Topics

- Proofs (especially induction)
  - If you want to debug a program beyond a doubt, prove that it's bug-free
  - Proof-theory has recently also been shown to be useful in discovering bugs in pre-production hardware
- Counting and Combinatorics
  - Compute your odds of winning lottery
  - Important for predicting how long certain computer program will take to finish
  - Useful in designing algorithms

Quick Overview - Topics

- Graph Theory
  - Many clever data-structures for organizing information and making programs highly efficient are based on graph theory
  - Very useful in describing problems in
    - Databases
    - Operating Systems
    - Networks
    - EVERY CS DISCIPLINE!!!!

Objectives

- Express statements precisely of formal logic
- Check the validity of arguments after analyzing them
- Use basic properties and operations that can be performed over sets.
- Use basic properties and operations related to relations and function
- Formula proof using mathematical phenomenon
- Working on statements to prove them using direct and indirect methods.

Objectives

- Compute probability of simple and conditional events
- Illustrate the basic definitions of graph theory and properties of graphs
- Relate each major topic in Discrete Mathematics to an application area in computing

Statements: (Proposition)

- A declarative sentence that is either true or false.
- law of the excluded middle: a proposition cannot be partially true or partially false
- law of contradiction: a proposition cannot be both true and false

Propositions: Examples

- Earth is round.
- I can speak English
- $x<98$
- $5+1=6$
- What time is it?

Each of these proposition has a truth value, True or False but not both.

Propositions:

- 9*10=9
- You are studying Discrete Mathematics
- Sky is blue
- Computers have feelings

Propositional Variable

- A name given to the proposition or statement is called a Propositional Variable.
- p1=Earth is round. True
- p2=I can speak English True
- p3=5+1=9 False

Propositional Logic

Axiomatic concepts in math:

- Equals
- Opposite
- Truth and falsehood
- Statement
- Objects
- Collections

Propositional Logic

We intuitively know that Truth and Falsehood are opposites. That statements describe the world and can be true/false. That the world is made up of objects and that objects can be organized to form collections.

False, True, Statements

Axiom: ***False*** is the opposite to ***Truth***.

A ***statement*** is a description of something.

Examples of statements:

- I'm 31 years old.
- I always tell the truth.
- I'm lying to you.

Q's: Which statements are True? False? Both? Neither?

False, True, Statements

True: I'm 31 years old.

False: I have 17 children.

I always tell the truth.

Both: IMPOSSIBLE, by our Axiom.

False, True, Statements

Neither: I'm lying to you. (*If viewed on its own*)

HUH? Well suppose that

$$S = \text{“I” am lying to you.”}$$

were **true**. In particular, I am actually lying, so *S* is false. So it's both **true and false**, impossible by the Axiom.

Okay, so I guess *S* must be **false**. But then I must not be lying to you. So the statement is true. Again it's both **true and false**.

In both cases we get the opposite of our assumption, so *S* is neither true nor false.

Propositions

To avoid painful head-aches, we ban such silly non-sense and avoid the most general type of statements limiting ourselves to statements with valid truth-values instead:

DEF: A ***proposition*** is a statement that is true or false.

Propositions

Propositional Logic is a *static* discipline of statements which lack *semantic content*.

E.G. p = "Clinton was the president."

q = "The list of U.S. presidents includes Clinton."

r = "Lions like to sleep."

Propositions

Propositional logic is the study of how simple propositions can come together to make more complicated propositions. If the simple propositions were endowed with some meaning – *and they will be very soon*– then the complicated proposition would have meaning as well, and then finding out the truth value is actually important!

Compound Propositions

In Propositional Logic, we assume a collection of *atomic* propositions are given: *p*, *q*, *r*, *s*, *t*, ….

Then we form compound propositions by using ***logical connectives*** (***logical operators***) to form propositional "molecules".

Logical Connectives

Operator	Symbol	Usage	Java
Negation	¬	not	!
Conjunction	∧	and	&&
Disjunction	∨	or	||
Exclusive or	⊕	xor	(p||q)&&(!p||!q)
Conditional	→	if,then	p?q:true
Biconditional	↔	iff	(p&&q)||(!p&&!q)

Compound Propositions: Examples

p = "Cruise ships only go on big rivers."

q = "Cruise ships go on the River Ravi."

r = "The Ravi is a big river."

$\neg r$ = "The Ravi is not a big river."

$p \wedge q$ = "Cruise ships only go on big rivers and go on the Ravi."

$p \wedge q \rightarrow r$ = "If cruise ships only go on big rivers and go on the Ravi, then the Ravi is a big river."

Negation

This just turns a false proposition to true and the opposite for a true proposition.

EG: p = "23 = 15 +7"

p happens to be false, so $\neg p$ is true.

In Java, "!" plays the same role:

```
!(23 == 15+7)
```

has the `boolean` value `true` whenever evaluated.

Negation

- Let p be a proposition. The negation of p, denoted by $\neg$ p (also denoted by $\overline{p}$), 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.

Truth Table

- To analyze a compound statements, the most easy way is to create a truth table for it.
- A **truth table** specifies the truth value of a compound proposition for all possible truth values of its constituent propositions.

p	$\sim p$
T	F
F	T

Negation – truth table

Logical operators are defined by **truth tables** –tables which give the output of the operator in the right-most column.

Here is the truth table for negation:

p	$\neg p$
F	T
T	F

Conjunction

Conjunction is a ***binary*** operator in that it operates on two propositions when creating compound proposition. On the other hand, negation is a ***unary*** operator (the only non-trivial one possible).

Conjunction

Conjunction is supposed to encapsulate what happens when we use the word “and” in English. I.e., for “*p* and *q* ” to be true, it must be the case that **BOTH** *p* is true, as well as *q*. If one of these is false, than the compound statement is false as well.

Conjunction

EG. p = “Zardari was the president.”

q = “Nawz Sharif was the president.”

r = “Muslims are believers.”

Assuming p and r are true, while q false.

Out of $p\wedge q$, $p\wedge r$, $q\wedge r$

only $p\wedge r$ is **true**.

Conjunction – truth table

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

Disjunction – truth table

Conversely, disjunction is true when at least one of the components is true:

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

Disjunction – caveat

Note: English version of disjunction "or" does not always satisfy the assumption that one of *p*/*q* being true implies that "*p* or *q* " is true.

Q: Can someone come up with an example?

Disjunction – caveat

A: The starter is served with

soup **or** salad.

Most restaurants definitely don't allow you to get *both* soup *and* salad so that the statement is false when both soup and salad is served. To address this situation, exclusive-or is introduced next.

Exclusive-Or – truth table

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

Conditional (Implication)

This one is probably the least natural. It's only partly similar to the English usage of "if, then" or "implies".

DEF: $p \rightarrow q$ is true if q is true, or if p is false. In the final case (p is true while q is false) $p \rightarrow q$ is false.

Semantics: "p implies q" is true if one can mathematically derive q from p.

Conditional -- truth table

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

Conditional

Q: Does this makes sense? Let's try examples for each row of truth table

1. If rabbits like mud then rabbits like mud.
2. If rabbits can fly then rabbits like mud.
3. If rabbits like mud then rabbits can fly.
4. If rabbits can fly then rabbits can fly.

Conditional

1. If rabbits like mud then rabbits like mud.

   True: nothing about this statement is false.

2. If rabbits can fly then rabbits like mud.

   True: argument for –only care about end-result. Argument against – counters common English hyperbole.

3. If rabbits like mud then rabbits can fly.

   False: seems to assert falsehood

4. If rabbits can fly then rabbits can fly.

   True. WAIT! By first reasoning in 3, when "if" part is false, should only care about "then" part!!!!!

   On other hand, standard English hyperbole.

Conditional: why F→F is True

Remember, all of these are mathematical constructs, not attempts to mimic English. Mathematically, *p* should imply *q* whenever it is possible to derive *q* by from *p* by using valid arguments. For example consider the mathematical analog of no. 4:

If 0 = 1 then 3 = 9.

Q: Is this true mathematically?

Conditional: why F→F is True

A: YES mathematically and YES by the truth table.

Here's a mathematical proof:

1. $0 = 1$ (assumption)

Conditional: why F→F is True

A: YES mathematically and YES by the truth table.

Here's a mathematical proof:

1. 0 = 1 (assumption)
2. 1 = 2 (added 1 to both sides)

Conditional: why F→F is True

A: YES mathematically and YES by the truth table.

Here's a mathematical proof:

1. 0 = 1 (assumption)
2. 1 = 2 (added 1 to both sides)
3. 3 = 6 (multiplied both sides by 3)

Conditional: why F→F is True

A: YES mathematically and YES by the truth table.

Here's a mathematical proof:

1. 0 = 1 (assumption)
2. 1 = 2 (added 1 to both sides)
3. 3 = 6 (multiplied both sides by 3)
4. 0 = 3 (multiplied no. 1 by 3)

Conditional: why F→F is True

A: YES mathematically and YES by the truth table.

Here's a mathematical proof:

1. 0 = 1 (assumption)
2. 1 = 2 (added 1 to both sides)
3. 3 = 6 (multiplied both sides by 3)
4. 0 = 3 (multiplied no. 1 by 3)
5. 3 = 9 (added no. 3 and no. 4)

An equivalent for implication

Is there an expression that is equivalent to $p \to q$ but uses only the operators ¬, Λ, V?

Consider the proposition ¬*p* V *q*

Implication		
p	*q*	$p \to q$
F	F	T
F	T	T
T	F	F
T	T	T

p	*q*	¬*p*	¬*p* V *q*
F	F	T	T
F	T	T	T
T	F	F	F
T	T	F	T

- Implication can be expressed by disjunction and negation:

$$p \rightarrow q \equiv \neg p \vee q$$

Conditional: why F→F is True

As we want the conditional to make sense in the semantic context of mathematics, we better define it as we have!

Other questionable rows of the truth table can also be justified in a similar manner.

Conditional: synonyms

There are many ways to express the conditional statement $p \rightarrow q$:

If *p* then *q*. *p* implies *q*. If *p*, *q*.

p only if *q*. *p* is sufficient for *q*.

Some of the ways **reverse** the order of *p* and *q* but have the same connotation:

q if *p*. *q* whenever *p*. *q* is necessary for *p*.

To aid in remembering these, I suggest inserting "is true" after every variable:

EG: "*p* is true only if *q* is true"

Bi-Conditional -- truth table

For $p \leftrightarrow q$ to be true, p and q must have the same truth value. Else, $p \leftrightarrow q$ *is false:*

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

Q : Which operator is $\leftrightarrow$ the opposite of?

Bi-Conditional

A : $\leftrightarrow$ has exactly the opposite truth table as $\oplus$.

This means that we could have defined the bi-conditional in terms of other previously defined symbols, so it is redundant. In fact, only really need negation and disjunction to define everything else.

Extra operators are for convenience.

Q: Could we define all other logical operations using only negation and exclusive or?

Bi-Conditional

A: No. Notice that negation and exclusive-or each maintain parity between truth and false: No matter what combination of these symbols, impossible to get a truth table with four output rows consisting of 3 T's and 1 F (such as implication and disjunction).

Precedence of Logical Operators

1. $\neg$
2. $\wedge$
3. $\vee$
4. $\rightarrow$
5. $\leftrightarrow$

Thus $p \vee q \rightarrow \neg r$ is equivalent to $(p \vee q) \rightarrow \neg r$.
If the intended meaning is $p \vee (q \rightarrow \neg r)$ then parentheses must be used.