MATH212: Discrete Mathematics

Intro. and

Propositional Logic

Dr. Fethi Fkih

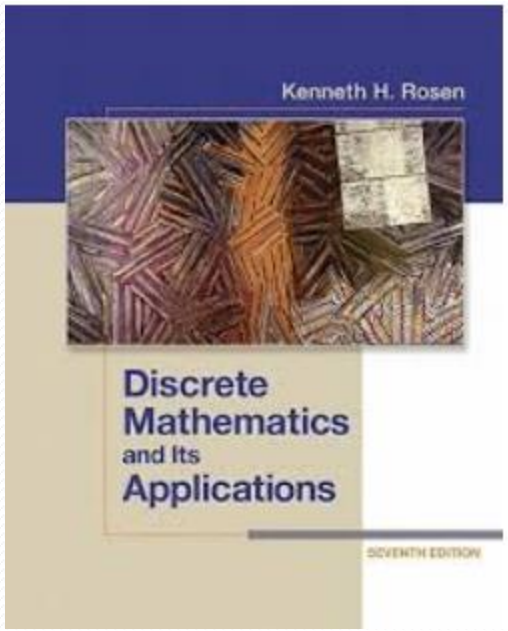
About this Course

- Give mathematical background you need for computer science.
- Topics: Logic and proof techniques, Basic Structures: Sets, Functions, Mathematical Induction, Relations, graphs.
- These will come up again and again in higher-level CS courses.

Goals of a Discrete Mathematics Course

- A discrete mathematics course has more than one purpose.
  - Students should learn a particular set of mathematical facts and how to apply them;
  - More importantly, such a course should teach students how to think logically and mathematically.

Textbook



- Textbook (optional): Discrete Mathematics and Its Applications by Kenneth Rosen
- Textbook not a substitute for lectures:
  - Class presentation may not follow book
  - Skip many chapters and cover extra material

Goals of a Course?

- A discrete mathematics course has more than one purpose:
  - Students should learn a particular set of mathematical facts and how to apply them;
  - More importantly, such a course should teach students how to think logically and mathematically.

Requirements?

- Weekly written homework assignments
- One midterm exam: in-class, closed-book
- Scheduled for week 6
- Final exam on week 12
- No make-up exams given unless you have serious, documented medical emergency.

Grading?

- Final exam: 50% of final grade.
- Midterm: 30% of final grade
- Homework + QUIZES + Class Participations : 20% of final grade

WHY STUDY DISCRETE MATHEMATICS?

- There are several important reasons for studying discrete mathematics. Through this course:
  - You can develop your mathematical maturity: that is, your ability to understand and create mathematical arguments.
  - Discrete mathematics provides the mathematical foundations for many computer science courses including data structures, algorithms, database theory, automata theory, formal languages, compiler theory, computer security, and operating systems.

WHY STUDY DISCRETE MATHEMATICS?

- Math courses based on the material studied in discrete mathematics include logic, set theory, number theory, linear algebra, abstract algebra, combinatorics, graph theory, and probability theory (the discrete part of the subject).

The Foundations: Logic and Proofs

Chapter 1, Part I: Propositional Logic

With Question/Answer Animations

Chapter Summary

- Propositional Logic
  - The Language of Propositions
  - Applications
  - Logical Equivalences
- Predicate Logic
  - The Language of Quantifiers
  - Logical Equivalences
  - Nested Quantifiers
- Proofs
  - Rules of Inference
  - Proof Methods
  - Proof Strategy

Propositional Logic

Section 1.1

Section Summary

- Propositions
- Connectives
  - Negation
  - Conjunction
  - Disjunction
  - Implication; contrapositive, inverse, converse
  - Biconditional
- Truth Tables

Introduction

- Logic rules is the basis of all mathematical reasoning.
- Besides the importance of logic in understanding mathematical reasoning, logic has numerous applications to computer science. These rules are used in the design of computer circuits, the construction of computer programs, the verification of the correctness of programs.
- Our discussion begins with an introduction to the basic building blocks of logic—propositions.

Propositions

- A *proposition* is a declarative sentence that is either true or false.
- Examples of propositions:
  a) The Moon is made of green cheese.
  b) Trenton is the capital of New Jersey.
  c) Toronto is the capital of Canada.
  d) $1 + 0 = 1$
  e) $0 + 0 = 2$
- Examples that are not propositions.
  a) Sit down!
  b) What time is it?
  c) $x + 1 = 2$
  d) $x + y = z$

Propositional Logic

- Constructing Propositions
  - Propositional Variables: p, q, r, s, …
  - The proposition that is always true is denoted by **T** and the proposition that is always false is denoted by **F**.
  - Compound Propositions; constructed from logical connectives and other propositions
    - Negation ¬
    - Conjunction ∧
    - Disjunction ∨
    - Implication →
    - Biconditional ↔

Compound Propositions: Negation

- The *negation* of a proposition p is denoted by $\neg p$ and has this truth table:

p	$\neg p$
T	F
F	T

- **Example**: If p denotes "The earth is round.", then $\neg p$ denotes "It is not the case that the earth is round," or more simply "The earth is not round."

Conjunction

- The *conjunction* of propositions p and q is denoted by $p \wedge q$ and has this truth table:

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

- **Example**: If p denotes "I am at home." and q denotes "It is raining." then $p \wedge q$ denotes "I am at home and it is raining."

Disjunction

- The *disjunction* of propositions p and q is denoted by $p \vee q$ and has this truth table:

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

- **Example**: If p denotes "I am at home." and q denotes "It is raining." then $p \vee q$ denotes "I am at home or it is raining."

The Connective Or in English

- In English "or" has two distinct meanings.
  - "Inclusive Or" - In the sentence "Students who have taken CS202 or Math120 may take this class," we assume that students need to have taken one of the prerequisites, but may have taken both. This is the meaning of disjunction. For $p \vee q$ to be true, either one or both of p and q must be true.
  - "Exclusive Or" - When reading the sentence "Soup or salad comes with this entrée," we do not expect to be able to get both soup and salad. This is the meaning of Exclusive Or (Xor). In $p \oplus q$, one of p and q must be true, but not both. The truth table for $\oplus$ is:

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

Implication

- If p and q are propositions, then $p \rightarrow q$ is a *conditional statement* or *implication* which is read as "if p, then q" and has this truth table:

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

- **Example**: If p denotes "I am at home." and q denotes "It is raining." then $p \rightarrow q$ denotes "If I am at home then it is raining."
- In $p \rightarrow q$, p is the *hypothesis* (*antecedent* or *premise*) and q is the *conclusion* (or *consequence*).

Understanding Implication

- In $p \rightarrow q$ there does not need to be any connection between the antecedent or the consequent. The "meaning" of $p \rightarrow q$ depends only on the truth values of p and q.
- These implications are perfectly fine, but would not be used in ordinary English.
  - "If the moon is made of green cheese, then I have more money than Bill Gates. "
  - "If the moon is made of green cheese then I'm on welfare."
  - "If 1 + 1 = 3, then your grandma wears combat boots."

Understanding Implication (cont)

- One way to view the logical conditional is to think of an obligation or contract.
  - "If I am elected, then I will lower taxes."
  - "If you get 100% on the final, then you will get an A+."
- If the politician is elected and does not lower taxes, then the voters can say that he or she has broken the campaign pledge. Something similar holds for the professor. This corresponds to the case where p is true and q is false.

Different Ways of Expressing $p \rightarrow q$

- **if** p**, then** q
- **if** p, q
- q **unless** $\neg p$
- q **if** p
- q **whenever** p
- q **follows from** p
- p **implies** q
- p **only if** q
- q **when** p
- q **when** p
- p **is sufficient for** q
- q **is necessary for** p

- **a necessary condition for** p **is** q
- **a sufficient condition for** q **is** p

Converse, Contrapositive, and Inverse

- From $p \rightarrow q$ we can form new conditional statements .
  - $q \rightarrow p$ is the **converse** of $p \rightarrow q$
  - $\neg q \rightarrow \neg p$ is the **contrapositive** of $p \rightarrow q$
  - $\neg p \rightarrow \neg q$ is the **inverse** of $p \rightarrow q$

Example: Find the converse, inverse, and contrapositive of "It raining is a sufficient condition for my not going to town."

Solution:

converse: If I do not go to town, then it is raining.

inverse: If it is not raining, then I will go to town.

contrapositive: If I go to town, then it is not raining.

Biconditional

- If p and q are propositions, then we can form the *biconditional* proposition $p \leftrightarrow q$, read as "p if and only if q." The biconditional $p \leftrightarrow q$ denotes the proposition with this truth table:

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

- If p denotes "I am at home." and q denotes "It is raining." then $p \leftrightarrow q$ denotes "I am at home if and only if it is raining."

Expressing the Biconditional

- Some alternative ways "*p* if and only if *q*" is expressed in English:

  - ***p* is necessary and sufficient for *q***
  - **if *p* then *q* , and conversely**
  - ***p* iff *q***

Truth Tables For Compound Propositions

- Construction of a truth table:
- Rows
  - Need a row for every possible combination of values for the atomic propositions.
- Columns
  - Need a column for the compound proposition (usually at far right)
  - Need a column for the truth value of each expression that occurs in the compound proposition as it is built up.
    - This includes the atomic propositions

Example Truth Table

- Construct a truth table for $p \vee q \rightarrow \neg r$

p	q	r	¬r	p ∨ q	p ∨ q → ¬r
T	T	T	F	T	F
T	T	F	T	T	T
T	F	T	F	T	F
T	F	F	T	T	T
F	T	T	F	T	F
F	T	F	T	T	T
F	F	T	F	F	T
F	F	F	T	F	T

Equivalent Propositions

- Two propositions are **e***quivalent* if they always have the same truth value.
- **Example**: Show using a truth table that the biconditional is equivalent to the contrapositive.

Solution:

p	q	$\neg p$	$\neg q$	$p \rightarrow q$	$\neg q \rightarrow \neg p$
T	T	F	F	T	T
T	F	F	T	F	F
F	T	T	F	T	T
F	F	T	T	T	T

Using a Truth Table to Show Non-Equivalence

Example: Show using truth tables that neither the converse nor inverse of an implication are not equivalent to the implication.

Solution:

p	q	$\neg p$	$\neg q$	$p \rightarrow q$	$\neg p \rightarrow \neg q$	$q \rightarrow p$
T	T	F	F	T	T	T
T	F	F	T	F	T	T
F	T	T	F	T	F	F
F	F	T	T	T	T	T

Problem

- How many rows are there in a truth table with *n* propositional variables?

Solution: 2^n We will see how to do this in Chapter 6.

- Note that this means that with n propositional variables, we can construct 2^n distinct (i.e., not equivalent) propositions.

Precedence of Logical Operators

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

$p \vee q \rightarrow \neg r$ 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.

Applications of Propositional Logic

Section 1.2

Applications of Propositional Logic: Summary

- Translating English to Propositional Logic
- 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
- "If I go to Harry's or to the country, I will not go shopping."
  - *p*: I go to Harry's
  - q: I go to the country.
  - *r*: I will go shopping.

If *p* or *q* then not *r*.

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

Example

Problem: Translate the following sentence into propositional logic:

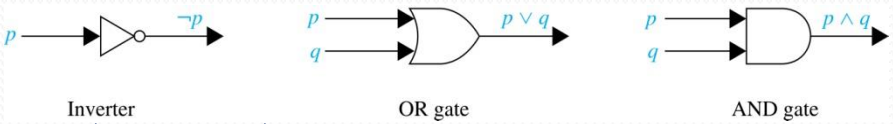
"You can access the Internet from campus only if you are a computer science major or you are not a freshman."

One Solution: Let *a*, *c*, and *f* represent respectively "You can access the internet from campus," "You are a computer science major," and "You are a freshman."

$$a \rightarrow (c \vee \neg f)$$

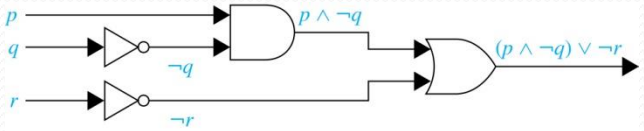
Logic Circuits (Studied in depth in Chapter 12)

- Electronic circuits; each input/output signal can be viewed as a 0 or 1.
  - 0 represents **False**
  - 1 represents **True**
- Complicated circuits are constructed from three basic circuits called gates.



Inverter OR gate AND gate

  - The inverter (**NOT gate**)takes an input bit and produces the negation of that bit.
  - The **OR gate** takes two input bits and produces the value equivalent to the disjunction of the two bits.
  - The **AND gate** takes two input bits and produces the value equivalent to the conjunction of the two bits.
- More complicated digital circuits can be constructed by combining these basic circuits to produce the desired output given the input signals by building a circuit for each piece of the output expression and then combining them. For example:



Propositional Equivalences

Section 1.3

Section Summary

- Tautologies, Contradictions, and Contingencies.
- Logical Equivalence
  - Important Logical Equivalences
  - Showing Logical Equivalence

Tautologies, Contradictions, and Contingencies

- A tautology is a proposition which is always true.
  - Example: $p \vee \neg p$
- A *contradiction* is a proposition which is always false.
  - Example: $p \wedge \neg p$
- A *contingency* is a proposition which is neither a tautology nor a contradiction, such as p

P	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

Logically Equivalent

- Two compound propositions p and q are logically equivalent if $p \leftrightarrow q$ is a tautology.
- We write this as $p \Leftrightarrow q$ or as $p \equiv q$ where p and q are compound propositions.
- Two compound propositions p and q are equivalent if and only if the columns in a truth table giving their truth values agree.
- This truth table show $\neg p \vee q$ is equivalent to $p \rightarrow q$.

p	q	$\neg p$	$\neg p \vee q$	$p \rightarrow q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

De Morgan's Laws

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

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



Augustus De Morgan

1806-1871

This truth table shows that De Morgan's Second Law holds.

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

Key Logical Equivalences

- Identity Laws: $p \wedge T \equiv p \ , \quad p \vee F \equiv p$
- Domination Laws: $p \vee T \equiv T \ , \quad p \wedge F \equiv F$
- Idempotent laws: $p \vee p \equiv p \ , \quad p \wedge p \equiv p$
- Double Negation Law: $\neg(\neg p) \equiv p$
- Negation Laws: $p \vee \neg p \equiv T \ , \quad p \wedge \neg p \equiv F$

Key Logical Equivalences (*cont*)

- Commutative Laws: $p \vee q \equiv q \vee p\,, \quad p \wedge q \equiv q \wedge p$
- Associative Laws: $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
  $(p \vee q) \vee r \equiv p \vee (q \vee r)$
- Distributive Laws: $(p \vee (q \wedge r) \equiv (p \vee q)) \wedge (p \vee r)$
  $(p \wedge (q \vee r)) \equiv (p \wedge q) \vee (p \wedge r)$
- Absorption Laws: $p \vee (p \wedge q) \equiv p \quad p \wedge (p \vee q) \equiv p$

More Logical Equivalences

TABLE 7 Logical Equivalences Involving Conditional Statements.

$$p \to q \equiv \neg p \vee q$$

$$p \to q \equiv \neg q \to \neg p$$

$$p \vee q \equiv \neg p \to q$$

$$p \wedge q \equiv \neg(p \to \neg q)$$

$$\neg(p \to q) \equiv p \wedge \neg q$$

$$(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$$

$$(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$$

$$(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$$

$$(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$$

TABLE 8 Logical Equivalences Involving Biconditional Statements.

$$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$$

$$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$$

$$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$$

$$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$$