Discrete Mathematics

Yongmei Liu
ymliu@mail.sysu.edu.cn

Dept. of Computer Science
Sun Yat-sen University

Fall 2015

Instructor and TA information

- The first part of the course: Logic and proofs
- My research interests: Logic in Computer Science (CS) and Artificial Intelligence (AI)
  - How to formally specify and prove correctness of programs
  - How to endow machines with the abilities of thinking through the means of computation (Thinking as Computation)
- TA: Kailun Luo, first-year PhD student, 231353612@qq.com
- Main duty: marking assignments

Course information

- Lectures: Wed 3-5th class in D203 (the 5th class is devoted to in-class exercises)
- Textbook: Discrete Mathematics and Its Applications, by Kenneth H. Rosen, 7th Edition

What is discrete mathematics?

- The part of mathematics devoted to the study of discrete objects, which are separated from each other.
- Examples of discrete objects: symbols, natural numbers, ...
- This course covers the fundamentals of discrete mathematics

Course topics (this term)

- Logic and proofs (Chap 1)
- Sets and functions (Chap 2)
- Algorithms (Chap 3)
- Number theory (chap 4)
- Induction and recursion (Chap 5)

Course topics (next term)

- Counting (Chaps 6&8)
- Relations (Chap 9)
- Graphs (Chap 10)
- Trees (Chap 11)

Why do CS students study this course?

There are two main reasons:

- To lay the mathematical foundations for successive computer science courses, for example, data structures, algorithms, database systems, artificial intelligence, etc.
- To develop mathematical maturity, and the ability to reason logically, and think abstractly and formally.

Course goals

- Mathematical reasoning: must understand mathematical reasoning in order to read, comprehend, and construct mathematical arguments.
- Combinatorial analysis: an important problem-solving skill is the ability to count or enumerate objects.
- Discrete structures: abstract mathematical structures used to represent discrete objects and relationships between them

Course goals (cont'd)

- Algorithmic thinking:
  - The specification of the algorithm
  - Verification that it works properly
  - The analysis of the computer memory and time required to perform it
- Applications and modeling:
  - DM has many applications in CS and other areas such as biology, chemistry, etc.
  - Modeling with DM is an extremely important problem-solving skill

Course evaluation

- 5 assignments (30%) + class participation (10%) + final exam (60%)
- Assignments are due in the beginning of classes; Late assignments will not be accepted except for documented medical or other emergencies
- Use A4-size paper for your assignments, write legibly, and staple your assignments properly
- The work you submit must be your own. If plagiarism is caught, all parties involved will receive 0 on the assignment

- Propositional logic
- Predicate logic
- Rules of inference
- Relating logic and mathematical proofs

What is logic?

- Logic is the formal systematic study of the principles of valid inference and correct reasoning
- An example of incorrect reasoning:
  If one has a driver's licence, then he / she is over age 18.
  Ann is over age 18. Thus Ann has a driver's license.
- Mathematical logic is the study of logic using mathematical methods

Propositional vs. predicate logic

- Compare the following inferences:
  1. If the solution is acid, then the litmus paper will turn red. The litmus paper didn't turn red. Thus, the solution is not acid.
  2. Any graduate student is a student. Ann is a graduate student. Thus Ann is a student.
- Propositional logic: we do not handle the connection between atomic propositions.
- Predicate logic: we further analyze the inner structure of atomic propositions.

What is a proposition?

A declarative sentence that is either true or false, but not both

Example

1. Washington, D.C., is the capital of USA.
2. Toronto is the capital of Canada.
3. $1 + 1 = 2$.
4. $2 + 2 = 3$.

Example

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

On propositions

- We use letters to denote propositional variables, *e.g.*, p, q, r, s
- The truth value of a true (resp. false) proposition is true (resp. false), denoted by T or 1 (resp. F or 0)
- New propositions, called compound propositions, are formed from existing propositions using logical operators, also called logical connectives.

Negation

- Let p be a proposition.
- The negation of p, denoted by $\neg p$ (read "not p"), is the statement "It is not the case that p."
- The truth value of $\neg p$ is the opposite of the truth value of p.
- Truth table
- Examples
  - "Today is Friday".
  - "At least 10 inches of rain fell today in Miami".

Conjunction

- Let p and q be propositions.
- The conjunction of p and q, denoted by $p \wedge q$, is the proposition "p and q".
- $p \wedge q$ is true when both p and q are true and is false otherwise.
- Truth table
- Note that the word "but" sometimes is used instead of "and" in a conjunction, *e.g.*, "The sun is shining, but it is raining."
- Example: "Today is Friday". "It is raining today".

Disjunction

- The disjunction of p and q, denoted by $p \vee q$, is the proposition "p or q".
- $p \vee q$ is false when both p and q are false and is true otherwise.
- Truth table
- Example: "Today is Friday". "It is raining today".

Inclusive vs. exclusive or

- Disjunction corresponds to inclusive or in English.
- For example, “Students who have taken calculus or computer science can take this class”.
- Compare with “Students who have taken calculus or computer science, but not both, can take this class”.
- The exclusive or of p and q, denoted by $p \oplus q$, is the proposition that is true when exactly one of p and q is true and is false otherwise.
- Truth table

Conditional Statements

- The conditional statement $p \to q$ is the proposition "if p, then q". p is called the hypothesis (or antecedent or premise) and q is called the conclusion (or consequence).
- $p \to q$ is false when p is true and q is false, and true otherwise.
- Truth table
- $p \to q$ is called a conditional statement because it asserts that q is true on the condition that p holds.
- A conditional statement is also called an implication.

Various ways to express $p \to q$

if p, then q
if p, q
* p is sufficient for q
q if p
q when p
a necessary condition for p is q
* q unless $\neg p$

p implies q
* p only if q
* a sufficient condition for q is p
q whenever p
q is necessary for p
q follows from p

Example

p: Maria learns discrete mathematics. q: Maria will find a good job.

Conditional statements in logic vs. in natural languages

- Natural language: "if it is sunny today, then we will go to the beach." There is a relationship between the hypothesis and the conclusion.
- Logic: "if today is Friday, then $2 + 3 = 5$."

Converse, contrapositive, and inverse

- The converse of $p \to q$ is $q \to p$.
- The contrapositive of $p \to q$ is $\neg q \to \neg p$.
- The inverse of $p \to q$ is $\neg p \to \neg q$.
- $\neg q \to \neg p$ has the same truth value as $p \to q$
- Example: The home team wins whenever it is raining.

Biconditional Statements

- The biconditional statement $p \leftrightarrow q$ is the proposition "p if and only if q".
- The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and false otherwise.
- Truth table
- Biconditional statements are also called bi-implications.
- $p \leftrightarrow q$ has the same truth value as $p \rightarrow q$ and $q \rightarrow p$.
- Other ways to express $p \leftrightarrow q$: p is necessary and sufficient for q; if p then q, and conversely; p iff q
- Example: p: "You can take the flight". q: "You buy a ticket".

Truth tables of compound propositions

- We can use the connectives to build up complicated compound propositions involving any number of propositional variables.
- We can use truth tables to determine the truth values of the compound propositions.

Example

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

Precedence of logical operators

- We generally use parentheses to specify the order in which logical operators in a compound proposition are to be applied. *e.g.*, $(p \vee q) \wedge (\neg r)$
- However, to reduce the number of parentheses, we specify the precedence of logical operators: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

Translating English sentences

English is often ambiguous. Translating sentence into logic removes the ambiguity.
Key: identify atomic propositions and logical connectives

Example

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

Example

You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old.

System Specifications

Use logic to specify the properties of hardware and software systems

Example

The automated reply cannot be sent when the file system is full.

System specifications should be consistent, that is, no contradiction can be derived.

Example

- The diagnostic message is stored in the buffer or it is retransmitted.
- The diagnostic message is not stored in the buffer.
- If the diagnostic message is stored in the buffer, then it is retransmitted.

How about adding the requirement "The diagnostic message is not retransmitted"

Logic Puzzles

Puzzles that can be solved using logical reasoning.

Example

- There are two kinds of people on an island: knights, who always tell the truth, and knaves, who always lie.
- You met two people A and B.
- A says: "B is a knight."
- B says: "The two of us are opposite types."
- What are A and B?

Syntax of propositional logic

Propositional formulas are built from the following symbols:

1. *atoms* $p, q, r, \ldots$
2. *unary connective* $\neg$, binary connectives $\wedge, \vee, \rightarrow, \leftrightarrow$
3. parentheses (,)

Propositional formulas are defined recursively as follows:

1. Any atom P is a formula.
2. If A is a formula so is $\neg A$.
3. If A, B are formulas, so are $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, and $(A \leftrightarrow B)$.

Semantics of propositional logic

A *truth assignment* is a mapping τ : the set of atoms $\to \{T, F\}$.
A truth assignment τ can be extended to assign either T or F to every formula, as follows:

1. $(\neg A)^\tau = T$ iff $A^\tau = F$
2. $(A \wedge B)^\tau = T$ iff $A^\tau = T$ and $B^\tau = T$
3. $(A \vee B)^\tau = T$ iff $A^\tau = T$ or $B^\tau = T$
4. $(A \to B)^\tau = T$ iff $A^\tau = F$ or $B^\tau = T$
5. $(A \leftrightarrow B)^\tau = T$ iff $A^\tau = B^\tau$

Some definitions

- τ *satisfies* A iff $A^\tau = T$; τ *satisfies* a set Φ of formulas iff τ satisfies A for all $A \in \Phi$. Φ is *satisfiable* iff some τ satisfies Φ; otherwise Φ is *unsatisfiable*. Similarly for A.
- A formula A is *valid* iff $A^\tau = T$ for all τ. A valid propositional formula is called a *tautology*.
- A and B are *logically equivalent* (written $A \Longleftrightarrow B$, or $A \equiv B$) iff $A^\tau = B^\tau$ for any τ.
- A is a logical consequence of Φ (written $\Phi \models A$) iff for any τ, if τ satisfies Φ, then τ satisfies A.

Examples

- $p \wedge q$ is satisfiable, $p \wedge \neg p$ is unsatisfiable
  $\{b \vee r, \neg b, b \rightarrow r\}$ is satisfiable, $\{b \vee r, \neg b, b \rightarrow r, \neg r\}$ is unsatisfiable
- $p \vee \neg p$ is valid (a tautology)
- $p \rightarrow q \Leftrightarrow \neg q \rightarrow \neg p$
- $\{p \vee q, \neg p\} \models q$, $\{p \rightarrow q, \neg q\} \models \neg p$

Proving logical equivalence: Method 1

Use truth table

- The truth assignments are listed in increasing order, starting with all 0, ending with all 1
- There is a column for each subformula
  A subformula of A is a substring of A which is itself a formula
  How many subformulas are there in $(p \vee \neg q) \rightarrow (p \wedge q)$

Examples:

- $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$
- $p \rightarrow q \Leftrightarrow \neg p \vee q$
- $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$

TABLE 6 Logical Equivalences.

Equivalence	*Name*
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws

$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

Associative laws

- $p \vee q \vee r$, $p \wedge q \wedge r$
- $\neg(p_1 \vee p_2 \vee \ldots \vee p_n) \equiv \neg p_1 \wedge \neg p_2 \wedge \ldots \neg p_n$
- $\neg(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \equiv \neg p_1 \vee \neg p_2 \vee \ldots \neg p_n$

Using De Morgan's laws

- Express the negation of "Miguel has a cell phone and he has a laptop"
- Express the negation of "Heather will go to the concert or Steve will go to the concert"

Commonly used logical equivalences

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

Proving logical equivalence: Method 2

A drawback of using truth table: when there are n atoms, there are 2^n rows in the truth table

Use already-known logical equivalences and the following results

- If $A \Leftrightarrow B$ and $B \Leftrightarrow C$, then $A \Leftrightarrow C$
- Replacement theorem: If B is a subformula of A and $B \Leftrightarrow B'$, let A' be the result of replacing B in A by B', then $A \Leftrightarrow A'$

Examples:

- $\neg(p \to q) \Leftrightarrow p \wedge \neg q$
- $\neg(p \vee (\neg p \wedge q)) \Leftrightarrow \neg p \wedge \neg q$
- $p \wedge q \to p \vee q \Leftrightarrow T$