Discrete Mathematics and Its Applications

Introductory Lecture

What is Discrete Mathematics?

- Discrete mathematics is the part of mathematics devoted to the study of discrete (as opposed to continuous) objects.
- Calculus deals with continuous objects and is not part of discrete mathematics.
- Examples of discrete objects: integers, steps taken by a computer program, distinct paths to travel from point A to point B on a map along a road network, ways to pick a winning set of numbers in a lottery.
- A course in discrete mathematics provides the mathematical background needed for all subsequent courses in computer science and for all subsequent courses in the many branches of discrete mathematics.

Kinds of Problems Solved Using Discrete Mathematics

- How many ways can a password be chosen following specific rules?
- How many valid Internet addresses are there?
- What is the probability of winning a particular lottery?
- Is there a link between two computers in a network?
- How can I identify spam email messages?
- How can I encrypt a message so that no unintended recipient can read it?
- How can we build a circuit that adds two integers?

Kinds of Problems Solved Using Discrete Mathematics

- What is the shortest path between two cities using a transportation system?
- Find the shortest tour that visits each of a group of cities only once and then ends in the starting city.
- How can we represent English sentences so that a computer can reason with them?
- How can we prove that there are infinitely many prime numbers?
- How can a list of integers be sorted so that the integers are in increasing order?
- How many steps are required to do such a sorting?
- How can it be proved that a sorting algorithm always correctly sorts a list?

Goals of a Course in Discrete Mathematics

- **Mathematical Reasoning**: Ability to read, understand, and construct mathematical arguments and proofs.
- **Combinatorial Analysis**: Techniques for counting objects of different kinds.
- **Discrete Structures**: Abstract mathematical structures that represent objects and the relationships between them. Examples are sets, permutations, relations, graphs, trees, and finite state machines.

Goals of a Course in Discrete Mathematics

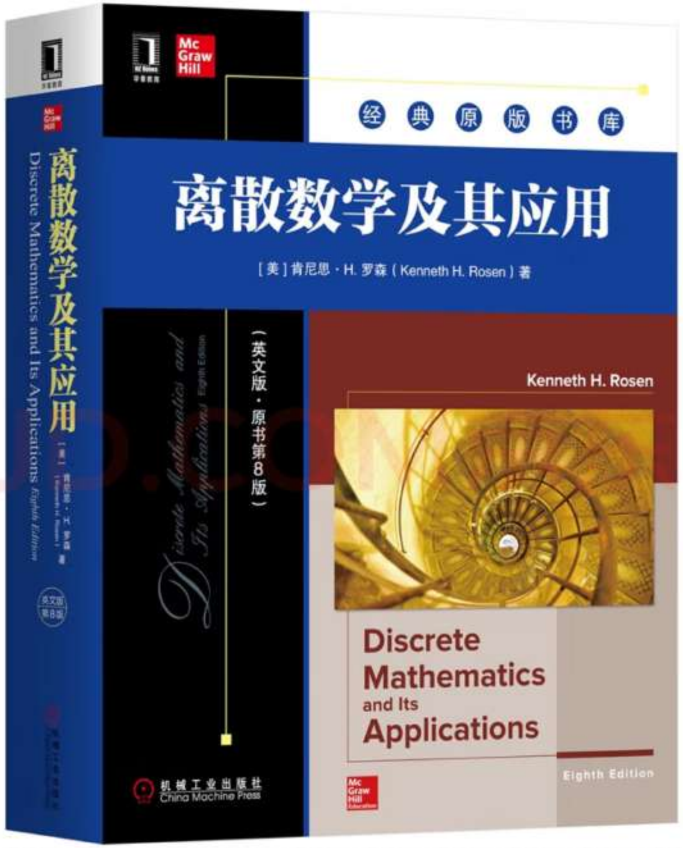
- **Algorithmic Thinking**: One way to solve many problems is to specify an algorithm. An algorithm is a sequence of steps that can be followed to solve any instance of a particular problem. Algorithmic thinking involves specifying algorithms, analyzing the memory and time required by an execution of the algorithm, and verifying that the algorithm will produce the correct answer.
- **Applications and Modeling**: It is important to appreciate and understand the wide range of applications of the topics in discrete mathematics and develop the ability to develop new models in various domains. Concepts from discrete mathematics have not only been used to address problems in computing, but have been applied to solve problems in many areas such as chemistry, biology, linguistics, geography, business, etc.

Discrete Mathematics is a Gateway Course

- Topics in discrete mathematics will be important in many courses that you will take in the future:
  - **Computer Science**: Computer Architecture, Data Structures, Algorithms, Programming Languages, Compilers, Computer Security, Databases, Artificial Intelligence, Networking, Graphics, Game Design, Theory of Computation, ……
  - **Mathematics**: Logic, Set Theory, Probability, Number Theory, Abstract Algebra, Combinatorics, Graph Theory, Game Theory, Network Optimization, …
    - The concepts learned will also be helpful in continuous areas of mathematics.
  - **Other Disciplines**: You may find concepts learned here useful in courses in philosophy, economics, linguistics, and other departments.

Textbook

计算机科学丛书
原书第8版
离散数学及其应用
[美] 肯尼思·H. 罗森（Kenneth H. Rosen）著
徐六通 杨娟 吴斌 译
经典教材全面升级，全球数百万学生阅读，新版包含超过800道例题和4200道习题
Discrete Mathematics and Its Applications
Eighth Edition
机械工业出版社
China Machine Press

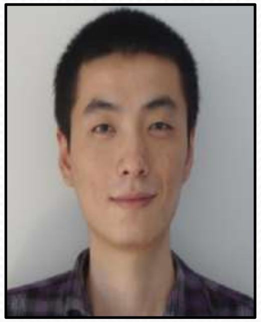


经典原版书库
离散数学及其应用
[美] 肯尼思·H. 罗森（Kenneth H. Rosen）著
（英文版·原书第8版）
Kenneth H. Rosen
Discrete Mathematics and Its Applications
Eighth Edition
机械工业出版社
China Machine Press

Instructors

Qiao Xiang

Congming Gao



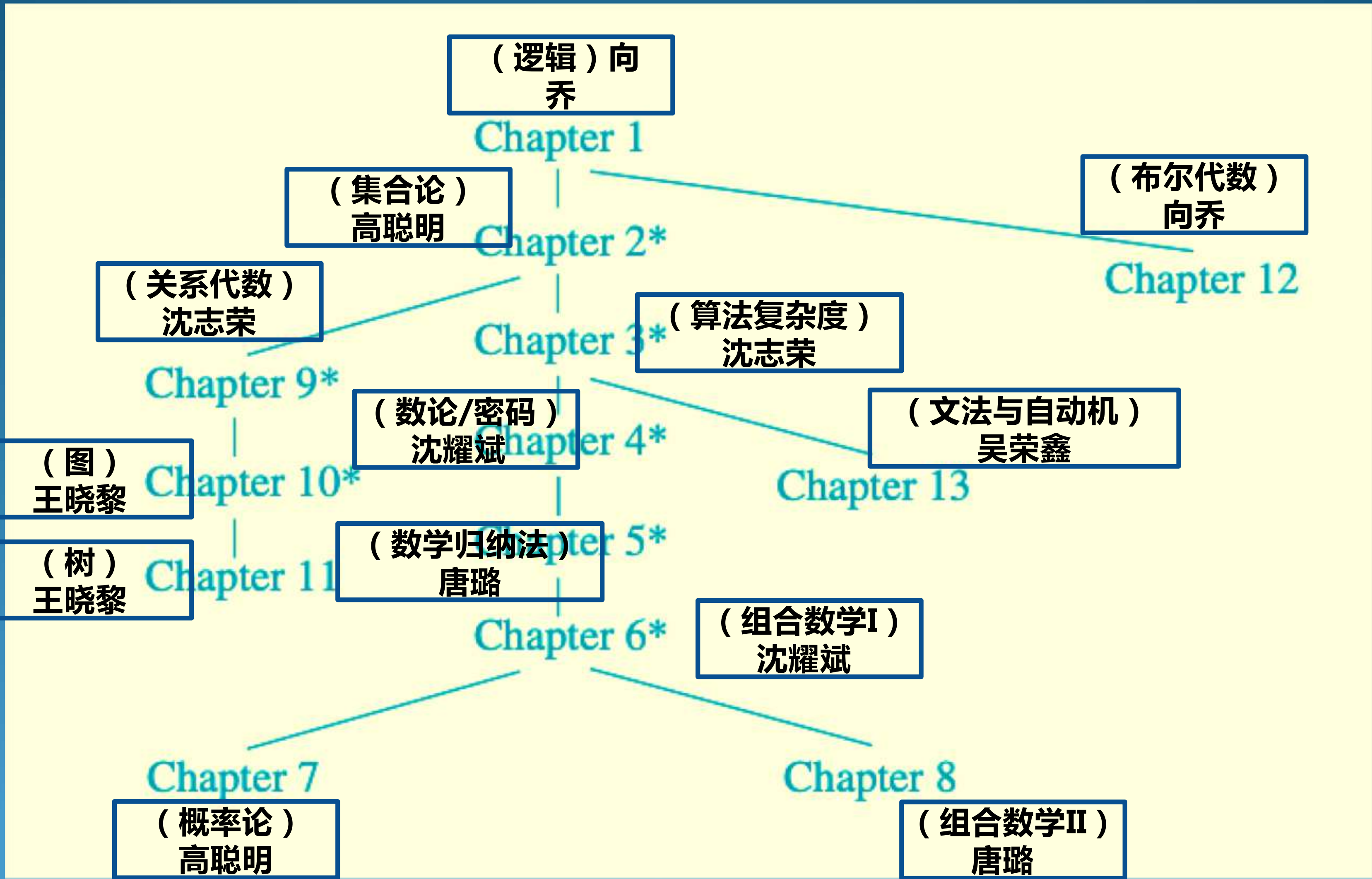
Zhirong Shen

Yaobin Shen

Xiaoli Wang

Rongxin Wu

Lu Tang


（逻辑）向乔
Chapter 1
（集合论）高聪明
Chapter 2*
（布尔代数）向乔
Chapter 12
（关系代数）沈志荣
Chapter 3*
（算法复杂度）沈志荣
Chapter 9*
（数论/密码）沈耀斌
Chapter 4*
（文法与自动机）吴荣鑫
Chapter 13
（图）王晓黎
Chapter 10*
（树）王晓黎
Chapter 11
（数学归纳法）唐璐
Chapter 5*
（组合数学I）沈耀斌
Chapter 6*
Chapter 7
（概率论）高聪明
Chapter 8
（组合数学II）唐璐

Workload

- One assignment per chapter (50%)
  - Expect to spend 1-3 hours on every assignment
- One mid-term exam (20%)
  - Chapters 1, 2, 3, 4, 5, and 12
- One final exam (30%)
  - Chapters 6, 7, 8, 9, 10, 11, 13

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 Summary

- The Language of Propositions
  - Connectives
  - Truth Values
  - Truth Tables
- Applications
  - Translating English Sentences
  - System Specifications
  - Logic Puzzles
  - Logic Circuits
- Logical Equivalences
  - Important Equivalences
  - Showing Equivalence
  - Satisfiability

Propositional Logic

Section 1.1

Section Summary

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

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) $x + y = z$

proposition 命题

Propositional Logic

- Constructing Propositions
  - Propositional Variables: $p, q, r, s, \ldots$
  - 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 ↔

Negation 否定
Conjunction 合取
Disjunction 析取
Implication 蕴含
Biconditional 双条件

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 $p \lor q$ and has this truth table:

p	q	$p \lor q$
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 $p \lor q$ 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

Inclusive Or 兼或
Exclusive Or 异或

Implication

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

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

hypothesis 假设
antecedent 前件
premise 前提

conclusion 结论
consequence 后件

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

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.

converse 逆命题
contrapositive 逆否命题
inverse 反命题

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 *equivalent* if they always have the same truth value.
- **Example**: Show using a truth table that the conditional is equivalent to the contrapositive.

Solution:

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

equivalent 等价

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
$\wedge$ $\vee$	2 3
$\rightarrow$ $\leftrightarrow$	4 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.

precedence 优先级
parentheses 括号

Applications of Propositional Logic

Section 1.2

Applications of Propositional Logic: Summary

- Translating English to Propositional Logic
- System Specifications
- Boolean Searching
- Logic Puzzles
- Logic Circuits
- AI Diagnosis Method (Optional)

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."

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

System Specifications

- System and Software engineers take requirements in English and express them in a precise specification language based on logic.

Example: Express in propositional logic:

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

Solution: One possible solution: Let *p* denote "The automated reply can be sent" and *q* denote "The file system is full."

$$q \rightarrow \neg p$$

Consistent System Specifications

Definition: A list of propositions is *consistent* if it is possible to assign truth values to the proposition variables so that each proposition is true.

Exercise: Are these specifications consistent?

- "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."

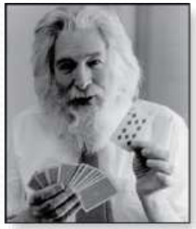
Solution: Let p denote "The diagnostic message is stored in the buffer." Let q denote "The diagnostic message is retransmitted" The specification can be written as: $p \lor q$, $\neg p$, $p \rightarrow q$. When p is false and q is true all three statements are true. So the specification is consistent.

- What if "The diagnostic message is not retransmitted is added."

  Solution: Now we are adding $\neg q$ and there is no satisfying assignment. So the specification is not consistent.

consistent 一致的

Logic Puzzles



Raymond
Smullyan
(Born 1919)

- An island has two kinds of inhabitants, *knights*, who always tell the truth, and *knaves*, who always lie.
- You go to the island and meet A and B.
  - A says "B is a knight."
  - B says "The two of us are of opposite types."

Example: What are the types of A and B?

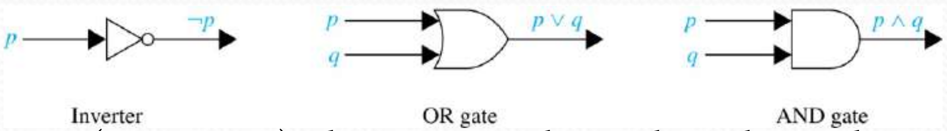
Solution: Let p and q be the statements that A is a knight and B is a knight, respectively. So, then $\neg p$ represents the proposition that A is a knave and $\neg q$ that B is a knave.

- If A is a knight, then p is true. Since knights tell the truth, q must also be true. Then $(p \wedge \neg q) \vee (\neg p \wedge q)$ would have to be true, but it is not. So, A is not a knight and therefore $\neg p$ must be true.
- If A is a knave, then B must not be a knight since knaves always lie. So, then both $\neg p$ and $\neg q$ hold since both are knaves.

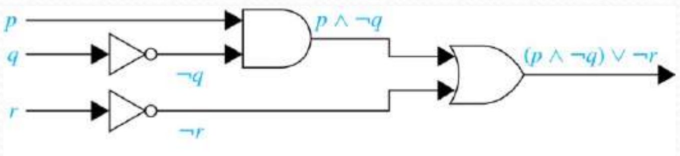
knaves 无赖

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.



  - 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
- Normal Forms (*optional, covered in exercises in text*)
  - Disjunctive Normal Form
  - Conjunctive Normal Form
- Propositional Satisfiability
  - Sudoku Example

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

tautology 永真式
contradiction 矛盾式
contingency 可能式

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 shows that $\neg p \vee q$ 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

Augustus De Morgan

1806-1871

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

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

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(恒等律): $p \wedge T \equiv p$, $p \vee F \equiv p$
- Domination Laws(支配律): $p \vee T \equiv T$, $p \wedge F \equiv F$
- Idempotent Laws(幂等律): $p \vee p \equiv p$, $p \wedge p \equiv p$
- Double Negation Law(双重否定律): $\neg(\neg p) \equiv p$
- Negation Laws(否定律): $p \vee \neg p \equiv T$, $p \wedge \neg p \equiv F$

Key Logical Equivalences (*cont*)

- Commutative Laws: $p \vee q \equiv q \vee p\ ,\quad p \wedge q \equiv q \wedge p$
  (交换律)
- Associative Laws: $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
  (结合律) $(p \vee q) \vee r \equiv p \vee (q \vee r)$
- Distributive Laws: $(p \vee (q \wedge r)) \equiv (p \vee q) \wedge (p \vee r)$
  (分配律) $(p \wedge (q \vee r)) \equiv (p \wedge q) \vee (p \wedge r)$
- Absorption Laws: $p \vee (p \wedge q) \equiv p \quad p \wedge (p \vee q) \equiv p$
  (吸收律)

More Logical Equivalences

TABLE 7 Logical Equivalences Involving Conditional Statements.

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

TABLE 8 Logical Equivalences Involving Biconditional Statements.

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

Constructing New Logical Equivalences

- We can show that two expressions are logically equivalent by developing a series of logically equivalent statements.
- To prove that $A \equiv B$ we produce a series of equivalences beginning with A and ending with B.

$$A \equiv A_1$$
$$\vdots$$
$$A_n \equiv B$$

- Keep in mind that whenever a proposition (represented by a propositional variable) occurs in the equivalences listed earlier, it may be replaced by an arbitrarily complex compound proposition.

Equivalence Proofs

Example: Show that $\neg(p \vee (\neg p \wedge q))$

is logically equivalent to $\neg p \wedge \neg q$

Solution:

$$\begin{aligned}
\neg(p \vee (\neg p \wedge q)) &\equiv \neg p \wedge \neg(\neg p \wedge q) && \text{by the second De Morgan law} \\
&\equiv \neg p \wedge [\neg(\neg p) \vee \neg q] && \text{by the first De Morgan law} \\
&\equiv \neg p \wedge (p \vee \neg q) && \text{by the double negation law} \\
&\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q) && \text{by the second distributive law} \\
&\equiv F \vee (\neg p \wedge \neg q) && \text{because } \neg p \wedge p \equiv F \\
&\equiv (\neg p \wedge \neg q) \vee F && \text{by the commutative law for disjunction} \\
&\equiv (\neg p \wedge \neg q) && \text{by the identity law for } \mathbf{F}
\end{aligned}$$

Equivalence Proofs

Example: Show that $(p \wedge q) \rightarrow (p \vee q)$ is a tautology.

Solution:

$$
\begin{aligned}
(p \wedge q) \rightarrow (p \vee q) &\equiv \neg(p \wedge q) \vee (p \vee q) && \text{by truth table for } \rightarrow \\
&\equiv (\neg p \vee \neg q) \vee (p \vee q) && \text{by the first De Morgan law} \\
&\equiv (\neg p \vee p) \vee (\neg q \vee q) && \text{by associative and} \\
& && \text{commutative laws} \\
& && \text{laws for disjunction} \\
&\equiv T \vee T && \text{by truth tables} \\
&\equiv T && \text{by the domination law}
\end{aligned}
$$

Disjunctive Normal Form (*optional*)

- A propositional formula is in *disjunctive normal form* if it consists of a disjunction of (1, … ,n) disjuncts where each disjunct consists of a conjunction of (1, …, m) atomic formulas or the negation of an atomic formula.
  - Yes $(p \wedge \neg q) \vee (\neg p \vee q)$
  - No $p \wedge (p \vee q)$
- Disjunctive Normal Form is important for the circuit design methods discussed in Chapter 12.

disjunctive Normal Form 析取范式

Disjunctive Normal Form (optional)

Example: Show that every compound proposition can be put in disjunctive normal form.

Solution: Construct the truth table for the proposition. Then an equivalent proposition is the disjunction with n disjuncts (where n is the number of rows for which the formula evaluates to **T**). Each disjunct has m conjuncts where m is the number of distinct propositional variables. Each conjunct includes the positive form of the propositional variable if the variable is assigned **T** in that row and the negated form if the variable is assigned **F** in that row. This proposition is in disjunctive normal from.

Disjunctive Normal Form (optional)

Example: Find the Disjunctive Normal Form (DNF) of

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

Solution: This proposition is true when r is false or when both p and q are false.

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

Conjunctive Normal Form (optional)

- A compound proposition is in *Conjunctive Normal Form* (CNF) if it is a conjunction of disjunctions.
- Every proposition can be put in an equivalent CNF.
- Conjunctive Normal Form (CNF) can be obtained by eliminating implications, moving negation inwards and using the distributive and associative laws.
- Important in resolution theorem proving used in artificial Intelligence (AI).
- A compound proposition can be put in conjunctive normal form through repeated application of the logical equivalences covered earlier.

Conjunctive Normal Form 合取范式

Conjunctive Normal Form (optional)

Example: Put the following into CNF:

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

Solution:

1. Eliminate implication signs:

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

2. Move negation inwards; eliminate double negation:

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

3. Convert to CNF using associative/distributive laws

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