Discrete Mathematics and Its Applications

Introduction

Wenjing Li

wjli@bupt.edu.cn

School of Computer Science

Beijing University of Posts & Telecommunications

Course Staff

- **Instructor**
  - 李文璟 *Wenjing Li*
  - Office: 新科研楼506
  - Email: *1228775678@qq.com*
- **QQ Group**
  - ***260239964***

群名称：离散数学2023春320...
群　号：260239964

- **Assistants**
  - 郭丞威，Email：*1609786514@qq.com*
  - 李辰旭，Email：*2315604672@qq.com*

Course time

- **Morning 8:00~8:45, 8:50~9:35**
- **Tuesday， S1-405**
- **Week 1~16**

北京邮电大学 本科教学校历

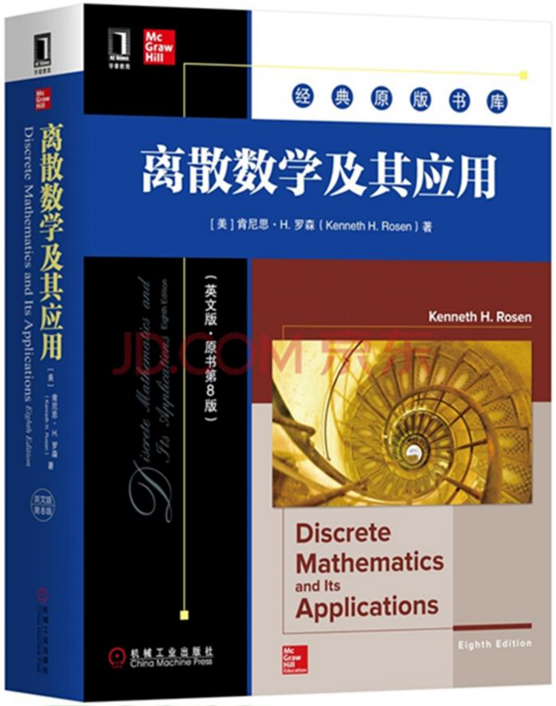
Beijing University of Posts and Telecommunications

（2022—2023 学年）

学期	/	春季学期																		暑假							
年份	二〇二三年																										
月份	二月		三月					四月				五月					六月				七月				八月			
周次	〇	一	二	三	四	五	六	七	八	九	十	十一	十二	十三	十四	十五	十六	十七	十八	十九	二十	廿一	廿二	廿三	廿四	廿五	廿六	廿七
星期一	13	20	27	6	13	20	27	3	10	17	24	劳动节	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21
星期二	14	21	28	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22
星期三	15	22	1	8	15	22	29	清明	12	19	26	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23
星期四	16	23	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	端午节	29	6	13	20	27	3	10	17	24
星期五	17	24	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25
星期六	18	25	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26
星期日	19	26	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27

Textbook & References

- **Discrete Mathematics and Its Applications/8e, Kenneth H. Rosen, 2020**
- 离散数学结构(第6版) 科曼(BernardKolman) 高等教育出版社:2010
- Liu, C.L. Elements of Discrete Mathematics, New York, McGraw-Hill, 1977
- 陈崇昕 等，离散数学，北京邮电大学出版社，1992
- 石纯一 等，数理逻辑与集合论/2e，清华大学出版社，2000



Introduction

- Computer Science/Computer Science and Technology
- Why to study Discrete Mathematics
- What Will We Study
- Course Arrangement

计算机科学与技术学科与离散数学

- **计算机科学与技术** 培养具有良好的科学素养、有深厚通信背景的从事计算机软硬件及网络的研究、设计、开发及综合应用的高级工程技术人才。毕业生能够在计算机和通信领域以及相关产业从事科研、应用开发、技术管理等工作。

- **计算机科学与技术**是研究计算机的设计与制造，利用计算机进行信息获取、表示、存储、处理、控制等的理论、原则、方法和技术的学科，是**科学性与工程性**并重的学科
  - 科学性：基础—逻辑能力，思辨精神
  - 工程性：应用—分析问题，解决问题

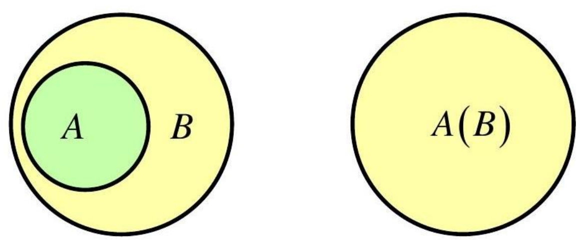
计算机科学与技术学科与离散数学

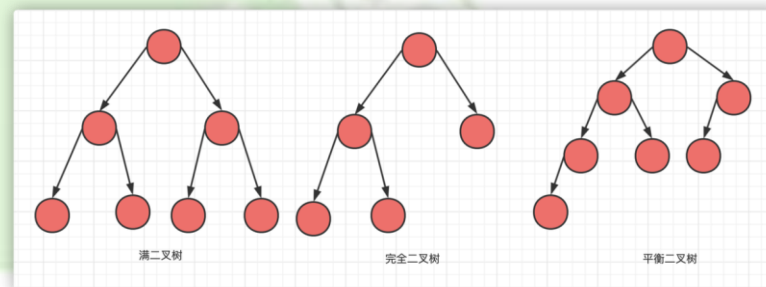
- 离散数学对计算机学科的重要性
  - 用数学思维和数学方法来抽象问题，描述问题
  - 问题求解
  - 处理对象离散
- 计算机专业核心课程
  - **离散数学、**计算导论与程序设计、数据结构、算法设计与分析、数据库系统原理、编译原理与技术、计算机网络、操作系统、软件工程、数字逻辑与数字系统、计算机组成原理、计算机系统结构、现代交换原理等

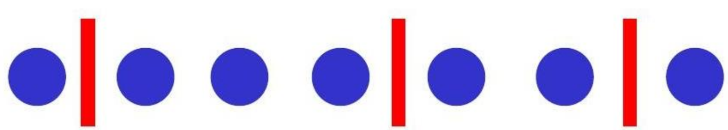
数学基础打牢固才能在计算机道路上走得更远更高

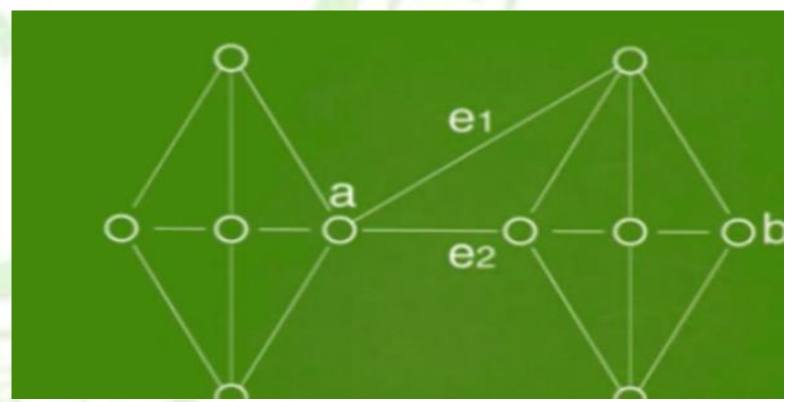
What is Discrete Mathematics?

- Discrete mathematics is the part of mathematics devoted to the study of **discrete** (as opposed to **continuous**) objects.









Why Study Discrete Math?

- The basis of all of digital information processing is: *Discrete manipulations of discrete structures represented in memory.*
- It's the basic language and conceptual foundation for all of computer science.
- Discrete math concepts are also widely used throughout math, science, engineering, economics, biology, *etc.*, …
- A generally useful tool for rational thought!

Why Study Discrete Math?

Example: Product Marketing at Minimum Cost

- **Market** a new product, say cell phone.
- **Strategy:** Give away free cell phones to few individuals, who will be the brand ambassadors and advertise the product to their friends.
- **Our goal:** Give away **minimum number** of cell phones, while ensuring that the **whole community** knows about the phone

Is this a **Discrete Math** problem?

If **yes**, where are

- Graphs?
- Computation?
- Counting?
- Sets?
- Proofs?
- ....

Why Study Discrete Math?

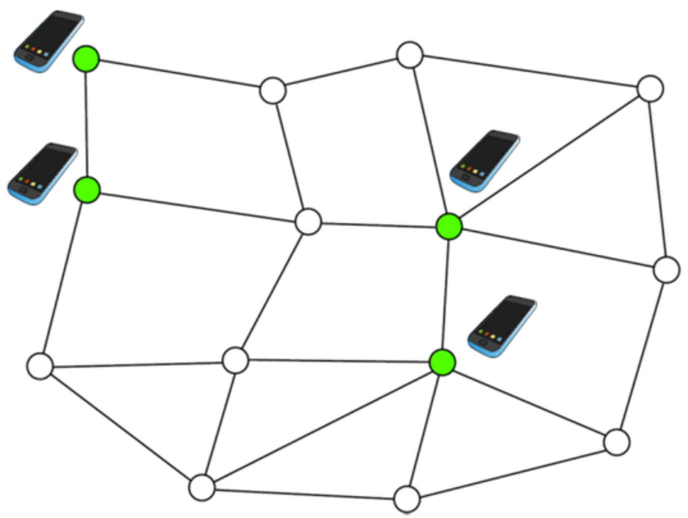
Example: Product Marketing at Minimum Cost

We can model individuals and their friendships as **graphs**.

How many free cell phones are needed? (3,4,5?) **(computation)**

Who should get the cell phone? How many possibilities are there? **(counting)**

Is this a best solution? **(proof)**



Five free phones should be sufficient? Can we do better?
Yes, **four** are sufficient.

Uses in Computer Science

- Advanced algorithms & data structures
- Programming language compilers & interpreters.
- Computer networks
- Operating systems
- Computer architecture
- Database management systems
- Cryptography
- Error correction codes
- Graphics & animation algorithms
- game engines, *etc.*…

almost the whole fields!

What Will We Study?

- 课程体系设计（课程大纲）
  - 课程目标
  - 课程内容

What Will We Study?

- **课程目标1：**
  - 理解逻辑和数学推理方法，这些方法是数学证明和程序设计的重要基础；培养学生严谨的逻辑推理能力

能从数学与工程角度对复杂工程问题进行表述、分析和建模

- **课程目标2：**
  - 了解包括集合、排列、关系、图、树等典型的抽象的离散结构，这些结构是计算机程序处理的主要对象；培养学生对复杂工程问题用计算机专业语言的刻画能力。

针对计算机系统和领域复杂工程问题，对任务目标给出需求描述

What Will We Study?

- **课程目标3：**
  - 学习基本计数技术，建立初步的组合分析思维方法；培养学生运用数学原理解决工程问题的能力

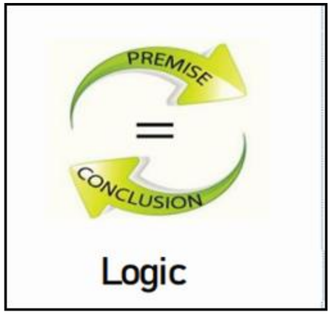
根据需求描述，运用数学、自然科学原理及方法进行分析，建立解决问题的抽象模型

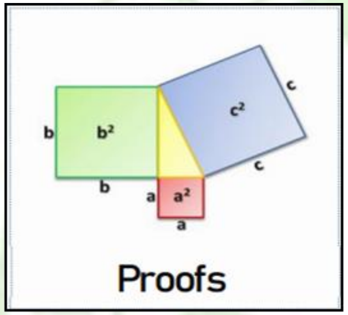
- **课程目标4：**
  - 认识到所学的离散数学知识在计算机等相关领域的实际应用，并能对算法、离散数学知识在解决复杂工程问题中应用和建模有深入理解；培养学生运用离散数学及方法解决工程问题的能力
- **课程目标5：**
  - 熟悉计算机相关数学知识的英文表述方法，培养计算机专业相关英文材料的阅读理解能力和初步交流沟通能力

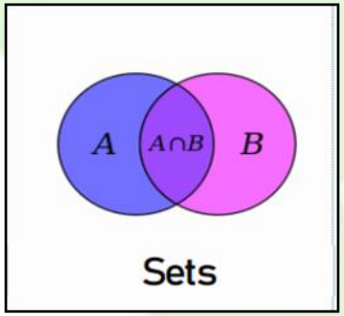
了解领域发展趋势，能在跨文化背景下进行沟通交流与合作

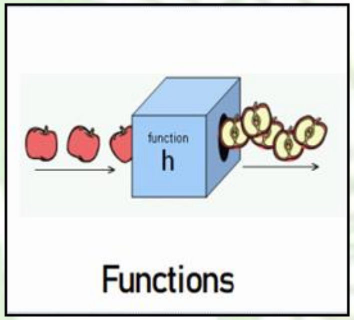
What Will We Study?

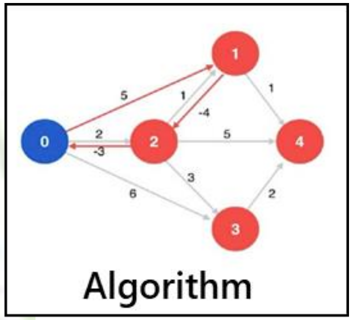
Discrete + **Structures**

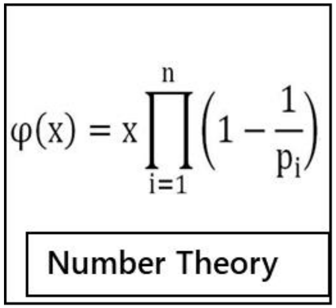


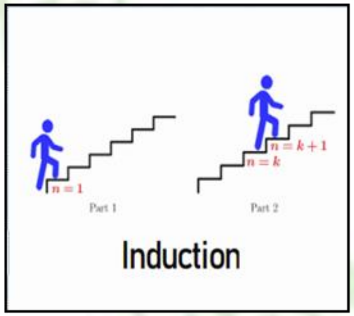


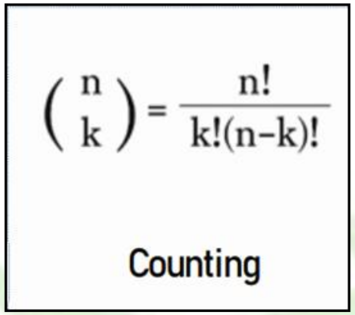


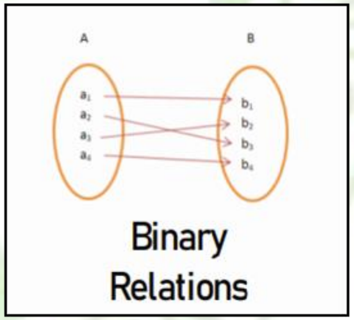


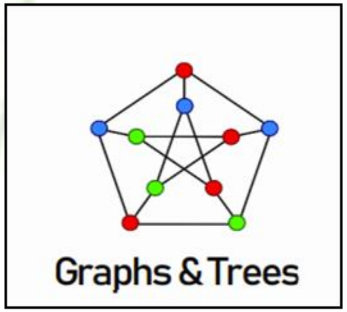












What Will We Study?

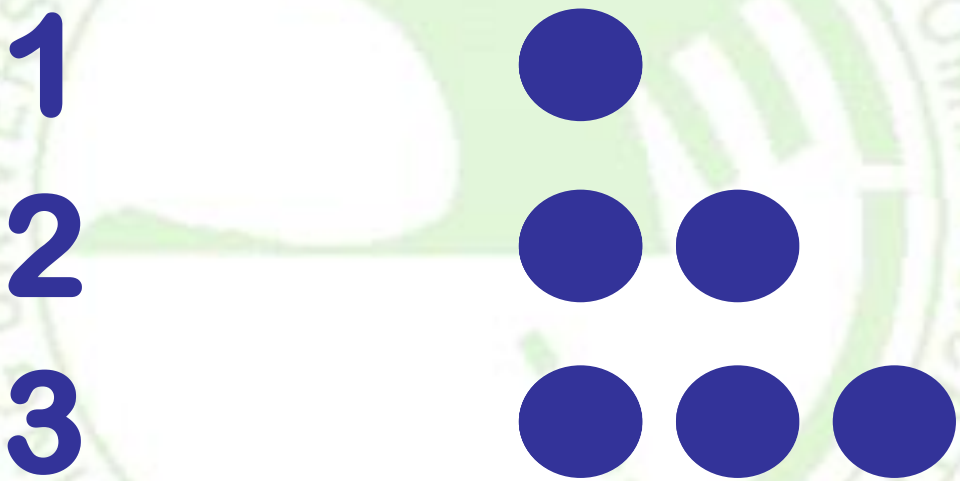
- **Method:**
  - Principles and techniques to solve the vast array of unfamiliar problems that arise in a rapidly changing field.

Exemplification

- Try out a problem or solution on small examples.

Representation

- Understand the relationship between different representations of the same information or idea.

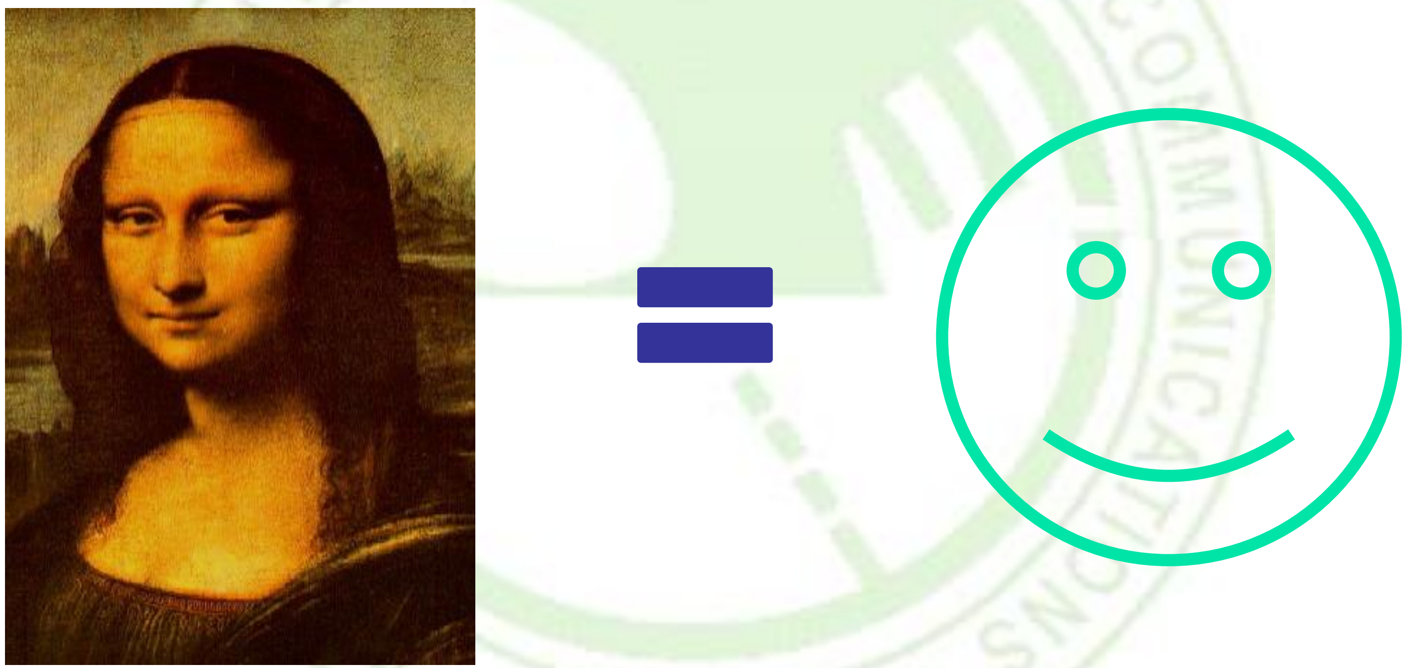


Modularity

- Decompose a complex problem into simpler subproblems.

Abstraction

- Abstract away the essential features of a problem.

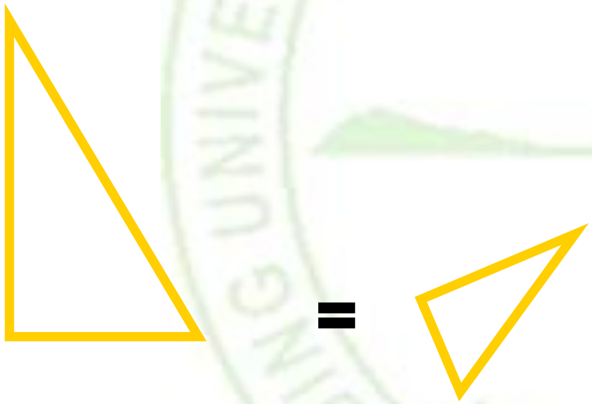


Refinement

- The best solution comes from a process of repeatedly refining and inventing alternative solutions.

Similarity

- **A significant form of intellectual progress is** to be able to classify and manipulate distinct objects with regard to a sense in which they are similar.



13 = 21 (modulo 2)

Toolbox

- Build your <u>toolbox</u> of abstract structures and concepts. Name your tools.
- Know the capacities and limits of each tool.

Course Arrangement

- Chap 1-The foundations：Logic and Proofs
- Chap 2-Basic Structures：Set, Function, Sequence, Matrix
- Chap 3-Algorithm
- Chap 4-Number Theory
- Chap 5-Induction and Recursion
- Chap 6-Counting

To you：How to study

- 上课认真听讲，下课按时做作业
- 英文教材：学习知识并非学习英语
- 常见问题：我上课都能听懂，但为什么不会解决实际问题呢？

练习，练习，再练习

Grading Scheme

- Homework assignments (30%)
- Midterm examination (20%)—Week 8^{th} /9^{th}
- Final examination (50%)

Q & A

Questions and Comments

QQ: ***260239964***

1. Logic and Proofs

Wenjing Li

wjli@bupt.edu.cn

School of Computer Science

Beijing University of Posts & Telecommunications

Logics

- **The discipline that deals with the *methods of reasoning*.**
  - On the elementary level, logic provides rules and techniques for determining whether a given argument valid.
  - In mathematics, logical reasoning is used to prove theorems.
  - In computer science to verify the correctness of programs and to prove theorems
  - In the natural and physical sciences to draw conclusions from experiments.
  - In the social sciences, and in our everyday lives to solve a multitude of problems.

- **Examples**
  - Crime Detection Problem
  - 8 Queens Problem

Crime Detection Problem

Example: A Crime Detection Problem

We know:

- One of them is **thief**
- Exactly one of them is speaking the truth

A.

B.

C.

I am not a thief | **A is the thief** | **I am not a thief**

Who is the thief ?

Crime Detection Problem

Example: A Crime Detection Problem

Suppose **A** is the thief,

A.

I am not a thief

B.

A is the thief

C.

I am not a thief

then both **B** and **C** are speaking the truth.

But, we know exactly one of them is speaking the truth. So, a contradiction. Hence,

Conclusion: **A** can't be the thief.

Crime Detection Problem

Example: A Crime Detection Problem

Suppose **B** is the thief,

A.

I am not a thief

B.

A is the thief

C.

I am not a thief

then both **A** and **C** are speaking the truth.

But, we know exactly one of them is speaking the truth. So, a contradiction. Hence,

<u>Conclusion:</u> **B** can't be the thief.

Crime Detection Problem

Example: A Crime Detection Problem

Suppose **C** is the thief,

A.

I am not a thief

B.

A is the thief

C.

I am not a thief

then **A** is speaking the truth, whereas, **B** and **C** are lying.

<u>Conclusion:</u> **C** is the thief.

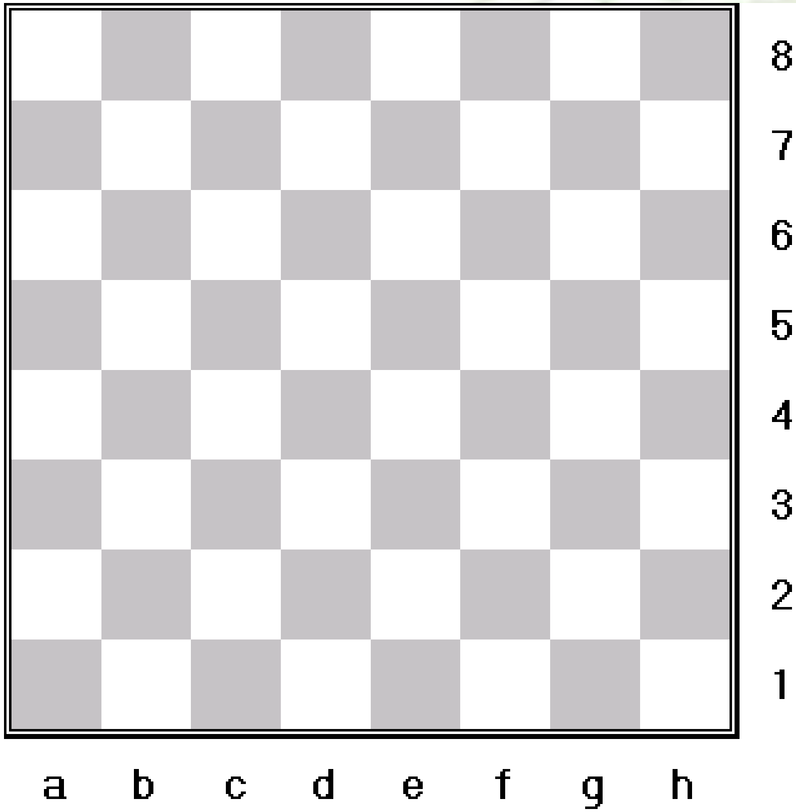
Crime Detection Problem

What if we have n persons, and exactly k of them are speaking the truth? Who is the thief?

Takeaways:

- We can infer new statements (conclusions) by carefully considering the given statements and premise.
- Things can get complex quickly, so we need to **formalize** and **systemize** our method of reasoning.

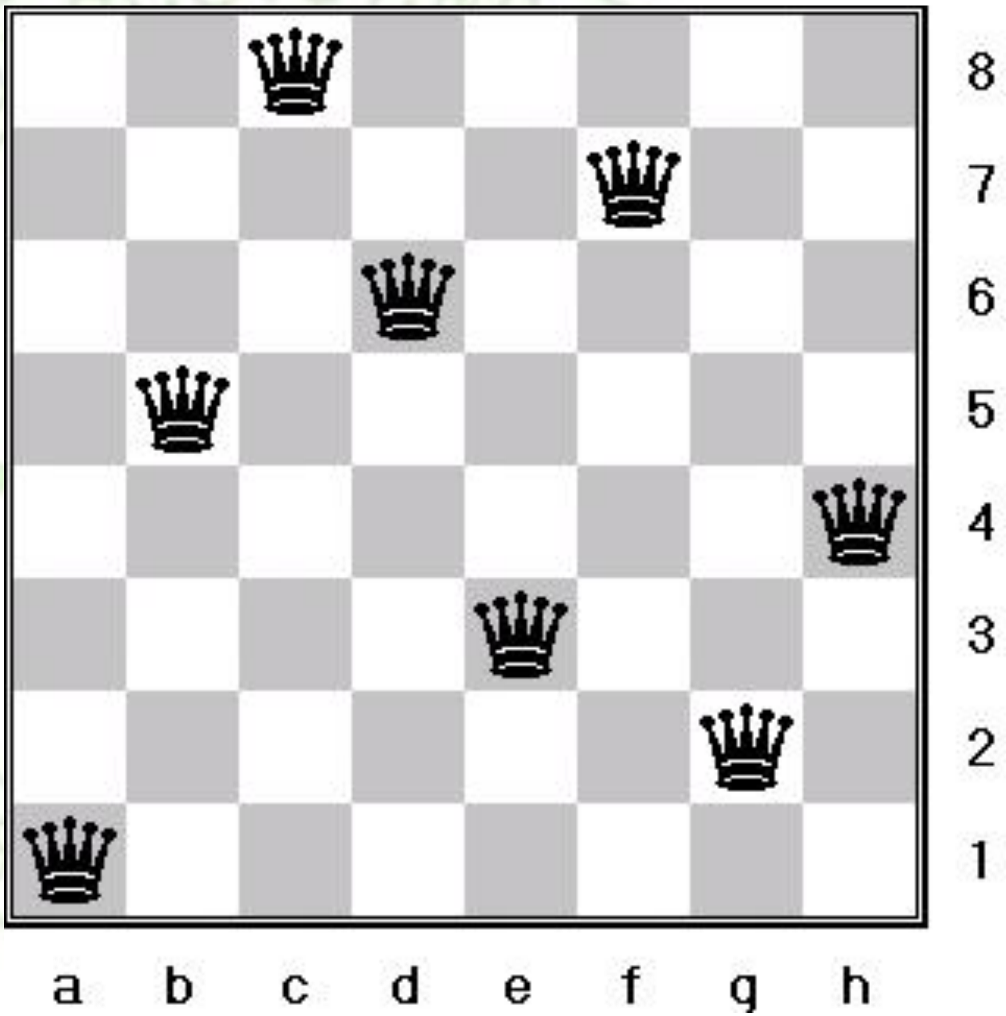
8 Queens Problem



- Eight queens are to be placed on a chess board in such a way that no queen checks against any other queen.
  - It was investigated by C. F. Guass in 1850, but he did not completely solve it.

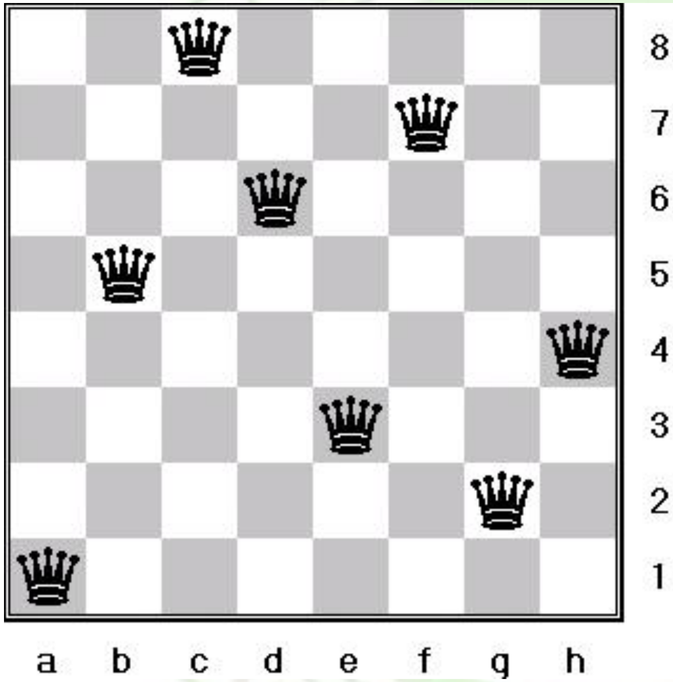
8 Queens Problem

- A solution



8 Queens Problem

- There are 92 solutions in all, but only 12 significantly differing solutions.



- 1 5 8 6 3 7 2 4
- 1 6 8 3 7 4 2 5
- 1 7 4 6 8 2 5 3
- 1 7 5 8 2 4 6 3
- 2 4 6 8 3 1 7 5
- 2 5 7 1 3 8 6 4
- 2 5 7 4 1 8 6 3
- 2 6 1 7 4 8 3 5
- 2 6 8 3 1 4 7 5
- 2 7 3 6 8 5 1 4
- 2 7 5 8 1 4 6 3
- 2 8 6 1 3 5 7 4

The problem of eight queens is a well-known example of the use of trial-and-error methods and of backtracking algorithms

1.1 Propositional logic(命题逻辑）

Wenjing Li

wjli@bupt.edu.cn

School of Computer Science

Beijing University of Posts & Telecommunications

OUTLINE

- **Propositions（命题）**
- **Compound propositions （复合命题）**
- **Propositional logic operators（命题逻辑联结符）**
  - Negation（否定）
  - Conjunction（合取）
  - Disjunction （析取）
  - Exclusive Or （异或）
  - Implication （蕴涵）
  - Biconditional （等价，双条件命题）
- **Bits and bit-strings（比特和比特串）**

Propositions（命题）

- A **statement** or **proposition** is a declarative sentence（陈述句）that is either *true* or *false*, but not both.
  - true = T (or 1) , false = F (or 0) (binary logic)

Statement	Proposition	Truth Value
17 is a prime number	√	True
The moon is made of green cheese.	√	False
For every positive integer n, there is a prime number larger than n	√	True
What time is it?	X（*interrogative*）	/
go to town!	X（*imperative*）	/

Examples

- Which of the following are statements?
  - (1) The earth is round.
  - (2) 2+3=5
  - (3) Do you speak English?
  - (4) $3-x=5$
  - (5) Take two aspirins.
  - (6) The temperature on the surface of the planet Venus is 800ºF.
  - (7) The sun will come out tomorrow.
  - (8) $x+y>5$

- **Concise, and the most un-ambiguous way of declaring a fact**
- **Has a definite truth value**

Logical Connectives and Compound Statements

- **Sometimes simple statements are not enough (to express complicated ideas).**
- **Combine propositions to get compound propositions using certain composition rules called logical operations or logical connectives.**

- propositional variables（命题变项）: $p,q, r,s, . . .$
  - p: The sun is shining today.
  - q: It is cold.
- Statements or propositional variables can be combined by logical connectives（逻辑联结词） to obtain compound statements(复合命题).
  - p and q: The sun is shining and it is cold.
  - p or q: The sun is shining or it is cold.

How to determine the truth value of a compound proposition?

Logical Connectives and Compound Statements

- The truth value of a compound statement depends only on the truth value of the statement being combined and on the type of connectives being used.
- **calculus of propositions（命题演算）**
  - New Propositions from old.
  - Relate new propositions to old using **TRUTH TABLES.**
- **Truth table（真值表）**
  - A table giving the truth values of a compound statement in terms of its component part.

Logical Connectives（逻辑联结词）

- Unary
  - **Negation（否定）**
- Binary
  - **Conjunction（合取）**
  - **Disjunction （析取）**
  - **Exclusive Or （异或）**
  - **Implication （蕴涵）**
  - **Biconditional （等价，双条件命题）**

Negation 'not' : ~/¬ （否定）

- **Definition:**
  - The negation of the statement p is the statement *not* p, denoted by $\sim p$ (*or* $\neg p$).
- **Example:**

  p: I am going to town.

  $\sim p$:

  - I am not going to town.
  - It is not the case that I am going to town.
- **Truth Table:**

p	$\sim p$
F (0)	**T (1)**
T (1)	**F (0)**

- Give the negation of the following statement:
  - (a) p: 2+3>1
  - (b) q: It is cold.

Conjunction 'and' : ∧(合取)

- **Definition:**
  - If p and q are statements, the ***conjunction*** of p and q is the compound statement "p and q", denoted by $p \wedge q$.
  - The connective ***and*** is denoted by the symbol $\wedge$.
- **Example:**
  - p - 'I am going to town'
  - q - 'It is going to rain'
  - $p \wedge q$: 'I am going to town and it is going to rain.'
- **Truth Table:**

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

Disjunction 'inclusive or': ∨ (析取)

- **Definition:**
  - If p and q are statements,the *disjunction* of p and q is the compound statement "p or q",denoted by $p \vee q$.
  - The connective *or* is denoted by the symbol ∨
- **Example:**
  - p - 'I am going to town'
  - q - 'It is going to rain'
  - $p \vee q$: 'I am going to town or it is going to rain.'
- **Truth Table:**

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

Note: *the* ***inclusive or*** *nature.*

Disjunction 'Exclusive or' : ⊕(不可兼或)

- **Example:**
  - *p* - 'I am going to town'
  - *q* - 'It is going to rain'
  - $p \oplus q$: 'Either I am going to town or it is going to rain.'
- **Truth Table:**

p	q	$p \oplus q$
0	0	0
0	1	1
1	0	1
1	1	0

- **可兼或：**
  - 明天下雨或刮风。
- **不可兼或：**
  - 今晚去影院看电影，或在图书馆学习。
  - 今天第一节课是语文课或数学课。
  - 他现在在401室或402室。

Note: Only one of *p* and *q* must be true.

Implication 'If…then…': →（蕴涵）

- **Definition:**
  - If p and q are statements, the compound statement "if p then q",denoted $p \to q$,is called a conditional statement(条件命题）,or implication.
  - The statement p is called the antecedent (前件）or hypothesis（假设）.
  - The statement q is called the *consequent*(后件）or *conclusion*（结论）.
  - The connective *if …then* is denoted by the symbol →.
- **Example:**

Hypothesis → Conclusion

  - p - 'I am going to town'
  - q - 'It is going to rain'
  - $p \to q$: 'If I am going to town then it is going to rain.'
- **Truth Table:**

p	q	$p \to q$
0	0	1
0	1	1
1	0	0
1	1	1

Implication 'If…then…': →（蕴涵）

- **Equivalent forms:**
  1) If P, then Q
  2) If P, Q
  3) P implies Q
  4) P is a sufficient condition (充分条件) for Q
  5) Q is a necessary condition (必要条件) for P
  6) a sufficient condition for Q is P
  7) Q if P
  8) Q when P
  9) Q follows from P
  10) Q whenever P
  11) Q unless not P
  12) P only if Q

- Note: The implication is false only when P is true and Q is false!

Implication ‘If…then…’: →（蕴涵）

- **There is no causality（因果关系） implied here!**
  - *‘If the moon is made of green cheese then I have more money than Bill Gates’ (T)*
  - *‘If the moon is made of green cheese then I’m on welfare’ (T)*
  - *‘If 1+1=3 then your grandma wears combat boots’ (T)*
  - *‘If I’m wealthy then the moon is not made of green cheese.’ (T)*
  - *‘If I’m not wealthy then the moon is not made of green cheese.’ (T)*

Implication ‘If…then…’: $\rightarrow$（蕴涵）

- **Terminology**
  - P = premise（前提）, hypothesis, antecedent
  - Q = conclusion, consequence
- **More terminology**
  - Q → P is the **CONVERSE（逆式）** of P → Q
  - $\sim P \rightarrow \sim Q$ is the **INVERSE（反式）** of P→ Q
  - $\sim Q \rightarrow \sim P$ is the **CONTRAPOSITIVE（逆反式）** of P → Q
- **Thinking**
  - One of these three has the *same meaning* (same truth table) as $p \rightarrow q$. Can you figure out?

Implication 'If…then…': → （蕴涵）

- **Truth tables**

p	q	$\sim p$	$\sim q$	$p \rightarrow q$	$q \rightarrow p$	$\sim q \rightarrow \sim p$	$\sim p \rightarrow \sim q$
0	0	1	1	1	1	1	1
0	1	1	0	1	0	1	0
1	0	0	1	0	1	0	1
1	1	0	0	1	1	1	1

A method to prove the equivalence of $p \rightarrow q$ and its contrapositive.

Implication 'If…then…': →（蕴涵）

- **Exercise:** Find the converse, inverse and contrapositive of the following statement:

  R: 'Raining tomorrow is a sufficient condition for my not going to town.'

  - Step 1: Assign propositional variables to component propositions
    - *p: It will rain tomorrow.*
    - *q: I will go to town.*
  - Step 2: Symbolize the assertion: ***R:*** $\boldsymbol{p \rightarrow \sim q}$
  - Step 3: Symbolize the converse: $\boldsymbol{\sim q \rightarrow p}$
  - Step 4: Convert the symbols back into words:
    - *'If I don't go to town then it will rain tomorrow'* or
    - *'Raining tomorrow is a necessary condition for my not going to town.' or*
    - *'My not going to town is a sufficient condition for it raining tomorrow.'*

Biconditional
'if and only if', 'iff': ↔（等价）

- **Definition:**
  - If p and q are statements, the compound statement p if and only if q denoted $p \leftrightarrow q$, is called an equivalence(等价）or biconditional（双条件命题）.
  - The connective *if and only if* is denoted by the symbol ↔ .
- **Example:**
  - p - 'I am going to town'
  - q - 'It is going to rain'
  - $p \leftrightarrow q$: 'I am going to town if and only if it is going to rain.'
- **Truth Table:**

p	q	$p \leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

Note: Both p and q must have the *same* truth value.

Boolean Operations Summary

- **Summary**
  - We have seen 1 unary operator and 5 binary operators.

- **Truth tables**

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
F	F	T	F	F	F	T	T
F	T	T	F	T	T	T	F
T	F	F	F	T	T	F	F
T	T	F	T	T	F	T	T

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

2^n

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.

Logic and Bit Operations

- ***Bit***
  - A **bit** is a **binary digit**: 0 or 1, which may be used to represent truth values.
  - 0 represents “false”; 1 represents “true”.
- ***Boolean algebra***
  - is like ordinary algebra except that variables stand for bits, + means “or”, and multiplication means “and”.
- ***A Bit string of length n***
  - is an ordered sequence (series, tuple) of $n \geq 0$ bits. (More on sequences in § 2.4)
  - By convention, bit strings are (sometimes) written left to right.
  - *Watch out! Another common convention is that the rightmost bit is bit #0, the 2nd-rightmost is bit #1, etc.*
  - When a bit string represents a base-2 number, by convention, the first (leftmost) bit is the *most significant* bit. *Ex.* $1101_2=8+4+1=13$.

Bitwise Operations

- Boolean operations can be extended to operate on bit strings as well as single bits.
- E.g.:
  01 1011 0110
  <u>11 0001 1101</u>
  11 1011 1111 Bit-wise OR
  01 0001 0100 Bit-wise AND
  10 1010 1011 Bit-wise XOR

End of § 1.1

You have learned about:

- Propositions: What they are.
- Propositional logic operators
  - Symbolic notations.
  - English equivalents.
  - Logical meaning.
  - Truth tables.
- Atomic vs. compound propositions.
- Bits and bit-strings.

Homework

- § 1.1
  - 2,14,16,28,30,32,40

- 交作业
  - 命名方式：“x班-姓名-第X次作业”

    (eg. 320班-王小帅-第1次作业）
  - 提交时间：每周五下午5点前
  - 提交方式：邮件提交给助教老师

    320班提交给郭丞威，321班提交给李辰旭

1.2 Application of propositional logic

Wenjing Li

wjli@bupt.edu.cn

School of Computer Science

Beijing University of Posts & Telecommunications

OUTLINE

- 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.
- **Example 1**:
  - *"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.

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

Translating English Sentences

- **Example 2:**
  - 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**:
  - *a* : *"You can access the internet from campus,"*
  - *c* : *"You are a computer science major,"*
  - *f* : *"You are a freshman."*

a only if (c or not f)

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

Translating English Sentences

- **Example 3**
  - *You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old.*
- **One Solution**:
  - *q: "You can ride the roller coaster."*
  - *r: "You are under 4 feet tall."*
  - *s: "You are older than 16 years old."*

¬q if (r and ¬s)

$$(r \wedge \neg s) \rightarrow \neg q.$$

System Specifications

- **Definition:**
  - System and Software engineers take requirements in English and express them in a **precise** specification language based on logic.
  - A list of propositions is ***consistent*** if it is possible to assign truth values to the proposition variables so that each proposition is true.
- **Example 1**:
  - Express in propositional logic: "*The automated reply cannot be sent when the file system is full*"
- **Solution:**
  - *p : "The automated reply can be sent"*
  - *q : "The file system is full."*

$\neg p$ **when** q

$q \rightarrow \neg p$

Consistent System Specifications

- **Example 2**
  - *The diagnostic message is stored in the buffer or it is retrasmitted.*
  - *The diagnostic message is not stored in the buffer.*
  - *If the diagnostic message is stored in the buffer, then it is retransmitted*
- **Solution:**
  - *p*： *"The diagnostic message is stored in the buffer."*
  - *q*： *"The diagnostic message is retransmitted"*
  - $p \lor q$, $\neg p$, $p \rightarrow q$.
  - *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?**
  - *Now we are adding* $\neg q$ *and there is no satisfying assignment. So the specification is **not consistent**.*

Boolean searches

- **Definition:**
  - Logical connectives are used extensively in searches of large collections of information. Because these searches employ techniques from propositional logic, they are called **Boolean Searches**.
  - *Connectives:* ***AND, OR, NOT***
- **Example:**
  - Web page searching.

Logic Puzzles

Raymond Smullyan
(Born 1919)

- **A famous puzzle by Smullyan:**
  - *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."*
- **Question:** ***What are the types of A and B?***
- **Solution:**
  - *p: A is a knight. q: B is a knight. $\neg p$: A is a knave. $\neg q$: B is a knave.*
  - *If A is a knight, then p is true. Since knights tell the truth, q must also be true. Then $(p \land \neg q) \lor (\neg p \land q)$ 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.*

Logic Puzzles

- **Muddy children puzzle:**
  - *The father says "At least one of you has a muddy forehead," and then asks the children to answer "Yes" or "No" to the question: "Do you know whether you have a muddy forehead?"*
  - *The father asks this question twice. What will the children answer each time this question is asked, assuming that:*
  - *A child can see whether his or her sibling has a muddy forehead, but cannot see his or her own forehead?*
  - *Assume that both children are honest and that the children answer each question simultaneously.*
- **Solution:**
  - *s: Son has a muddy forehead. d: Daughter has a muddy forehead.* $\mathbf{s \lor d}$
  - *Both will answer 'no' the first time because each know other is true.*
  - *Both will answer 'yes' the second time.*

Logic Circuits

- **Propositional logic can be applied to the design of computer hardware.**
  - *Logic circuit or digital circuit receives input signals and produces output signals.*

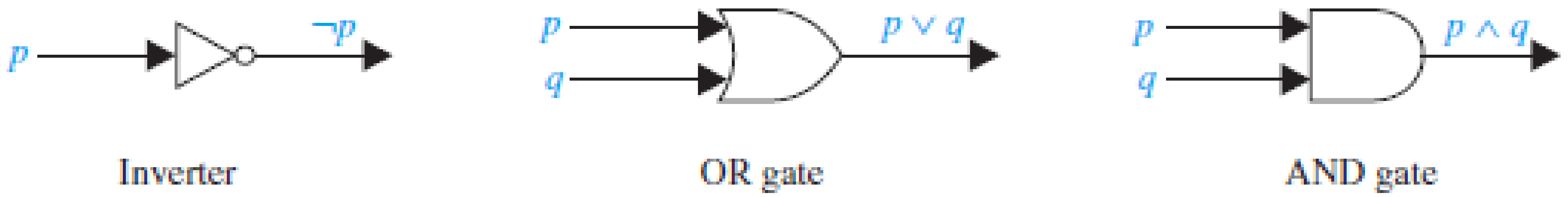


FIGURE 1 Basic logic gates.

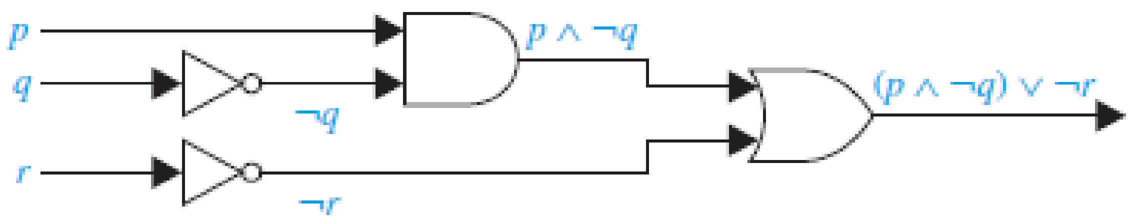


FIGURE 2 A combinatorial circuit.

Logic Circuits

- **Example**
  - *Build a digital circuit that produces the output*

$(p \vee \neg r) \wedge (\neg p \vee (q \vee \neg r))$ *when given input bits p, q, and r.*

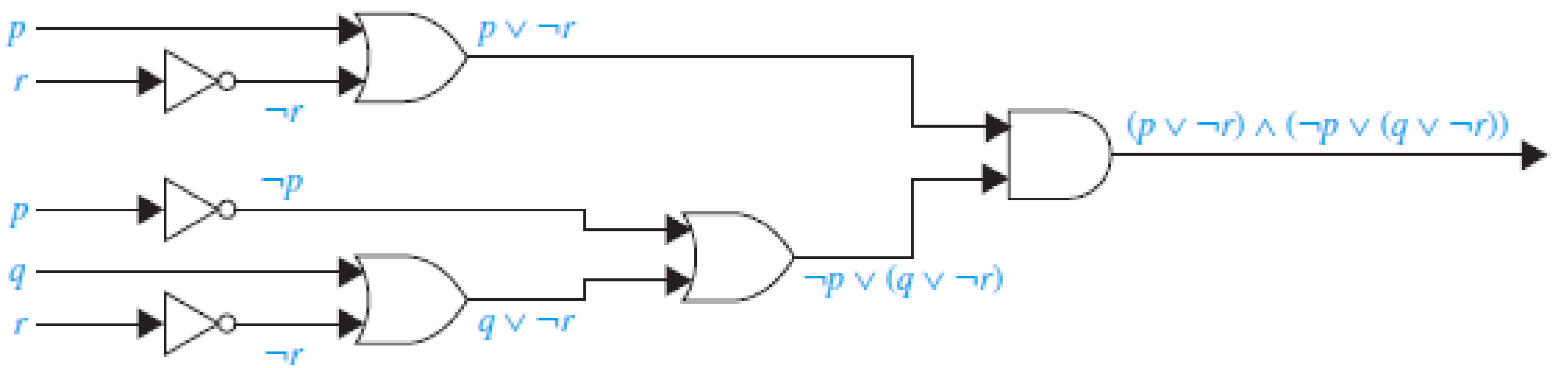


FIGURE 3 The circuit for $(p \vee \neg r) \wedge (\neg p \vee (q \vee \neg r))$.

Homework

- § 1.2
  - 4、22、36、43

交作业

命名方式： “x班-姓名-第X次作业”

(eg. 320班-王小帅-第1次作业）

提交时间： 每周五下午5点前

提交方式： 邮件提交给助教老师

320班提交给郭丞威，321班提交给李辰旭