离散数学

Discrete Mathematics

极夜酱

目录

Chapter 1 逻辑

1.1 命题

1.1.1 命题（Proposition）

逻辑（logic）规则给出数学语句的准确含义，这些规则用来区分有效和无效的数学论证。逻辑不仅对理解数学推理十分重要，而且在计算机科学中有许多应用，逻辑可用于电路设计、程序构造、程序正确性证明等方面。

命题是逻辑的基本成分，一个命题是一个具有真值（truth value）的语句，命题可以为真也可以为假，但不能既为真又为假。

命题	非命题
I have a dog.	What day is today?
1 + 2 = 3	Shut the door!
Today is Wednesday.	1 + 2
It is snowing today.	x + 1 = 2

命题习惯上用字母 p, q, r, s 等来表示，如果一个命题是真命题，它的真值为真，用 T 表示；如果一个命题是假命题，它的真值为假，用 F 表示。

1.1.2 非运算符（NOT, Negation Operator）

非运算符 $\neg$ 只作用于一个命题，其作用是反转命题的真值。

真值表（truth table）可以给出命题真值之间的关系，在确定由简单命题组成的命题的真值时，真值表特别有用。

p	$\neg p$
T	F
F	T

表 1.1: NOT 真值表

Exercise $\neg p$

p: It snowed last night.

$\neg p$: It didn;t snow last night.

q: $2+3=6$

$\neg q$: $2+3\neq 6$

1.1.3 合取运算符（AND, Conjunction Operator）

命题 $p \wedge q$ 表示 p 并且 q，当 p 和 q 都为真时命题为真，否则为假。

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

表 1.2: AND 真值表

Exercise $p \wedge q$

p: 今天是星期五。

q: 今天会下雨。

$p \wedge q$: 今天是星期五并且会下雨。

1.1.4 析取运算符（OR, Disjunction Operator）

命题 $p \vee q$ 表示 p 或 q，当 p 和 q 都为假时命题为假，否则为真。

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

表 1.3: OR 真值表

Exercise $p \vee q$

p: 开关坏了。

q: 灯泡坏了。

$p \vee q$: 开关坏了或者灯泡坏了。

1.1.5 异或运算符（XOR, Exclusive Or）

命题 $p \oplus q$ 表示 p 和 q 的异或，当 p 和 q 中恰有一个为真时命题为真，否则为假。

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

表 1.4: XOR 真值表

Exercise $p \oplus q$

p: 他现在在上海。

q: 他现在在北京。

$p \vee q$: 他现在在上海或北京。

Exercise 某地发生了一件谋杀案，警察通过排查确定杀人凶手必为 4 个嫌疑犯的一个，根据以下信息确定凶手。

A 说：不是我。

B 说：是 C。

C 说：是 D。

D 说：C 在胡说。

已知 3 个人说了真话，1 个人说的是假话。

```python
def main():
    for killer in ['A', 'B', 'C', 'D']:
        if (killer != 'A') + (killer == 'C') \
            + (killer == 'D') + (killer != 'D') == 3:
            print(killer)

if __name__ == "__main__":
    main()
```

运行结果 C

1.2 复合命题

1.2.1 复合命题（Compound Proposition）

使用非运算符和已定义的各联结词可以构造复合命题。小括号用于规定复合命题中多个逻辑运算符的操作顺序，为了减少所需的小括号数量，规定了各联结词的优先级。

优先级	**运算符**
1	$\neg$
2	$\wedge$ / $\vee$
3	$\rightarrow$ / $\leftrightarrow$

表 1.5: 运算符优先级

1.2.2 蕴含运算符（Implication Operator）

命题 $p \rightarrow q$ 表示 p 蕴含 q，在 p 为真而 q 为假时命题为假，否则为真。其中 p 称为前提，q 称为结论。

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

表 1.6: 蕴含真值表

表示 $p \rightarrow q$ 的术语有很多种，常见的有：

- If p, then q.
- p only if q.
- q is necessary for p.

Exercise $p \to q$

p：我去看电影。

q：我买奶茶。

$p \to q$：如果我去看电影，那么我会买奶茶。

由 $p \to q$ 可以构造出几个相关的蕴含：

1. $q \to p$ 称为 $p \to q$ 的逆命题（converse）。
2. $\neg q \to \neg p$ 称为 $p \to q$ 的逆否命题（contrapositive）。

Exercise 逆命题与逆否命题

p：今天是星期四。

q：我今天有考试。

$p \to q$：如果今天是星期四，那么我今天有考试。

$q \to p$：如果我今天有考试，那么果今天是星期四。

$\neg q \to \neg p$：如果我今天没有考试，那么今天不是星期四。

1.2.3 双向蕴含（Biconditional Operation）

命题 $p \leftrightarrow q$ 表示 p 双向蕴含 q，在 p 和 q 有相同的真值时命题为真，否则为假。

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

表 1.7: 双向蕴含真值表

双蕴含的真值表与异或的真值表正好相反，因此 $p \leftrightarrow q$ 与 $\neg(p \oplus q)$ 等价。

1.3 逻辑等价

1.3.1 逻辑等价（Logical Equivalence）

两个不同的复合命题可能拥有完全相同的真值，则称这两个命题在逻辑上是等价的。如果无论复合命题中各个命题的真值是什么，复合命题的真值总是为真，这样的复合命题称为永真式（tautology）。如果真值永远为假的复合命题称为矛盾（contradiction）。

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

表 1.8: 逻辑等价

如果复合命题 s 和是 r 逻辑等价的，可表示为 $s \equiv r$。只有当 $s \leftrightarrow r$ 是永真式时，s 和 r 才是逻辑等价的。

Exercise 使用真值表证明 $p \vee q \equiv \neg(\neg p \wedge \neg q)$

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

1.3.2 逻辑等价定理

幂等律 Idempotent Laws

$$p \wedge p \equiv p \tag{1.1}$$

$$p \vee p \equiv p \tag{1.2}$$

恒等律 Identity Laws

$$p \wedge T \equiv p \tag{1.3}$$

$$p \vee F \equiv p \tag{1.4}$$

支配律 Domination Laws

$$p \vee T \equiv T \tag{1.5}$$

$$p \wedge F \equiv F \tag{1.6}$$

双非律 Double Negation Law

$$\neg(\neg p) \equiv p \tag{1.7}$$

交换律 Commutative Laws

$$p \wedge q \equiv q \wedge p \tag{1.8}$$

$$p \vee q \equiv q \vee p \tag{1.9}$$

结合律 Associative Laws

$$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r) \tag{1.10}$$

$$(p \vee q) \vee r \equiv p \vee (q \vee r) \tag{1.11}$$

分配率 Distributive Laws

$$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r) \tag{1.12}$$

$$(p \vee q) \vee r \equiv p \vee (q \vee r) \tag{1.13}$$

德摩根律 De Morgan's Laws

$$\neg(p \wedge q) \equiv \neg p \vee \neg q \tag{1.14}$$

$$\neg(p \vee q) \equiv \neg p \wedge \neg q \tag{1.15}$$

吸收律 Absorption Laws

$$p \wedge (p \vee q) \equiv p \tag{1.16}$$

$$p \vee (p \wedge q) \equiv p \tag{1.17}$$

条件恒等

$$p \to q \equiv \neg p \vee q \tag{1.18}$$

$$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p) \tag{1.19}$$

Exercise 证明 $(p \vee q) \to p$ 永真

$$\begin{aligned}
&(p \vee q) \to p \\
&\equiv \neg(p \wedge q) \vee p \\
&\equiv (\neg p \vee \neg q) \vee p \\
&\equiv (\neg q \vee \neg p) \vee p \\
&\equiv \neg q \vee (\neg p \vee p) \\
&\equiv \neg q \vee T \\
&\equiv T
\end{aligned}$$

1.4 谓词与量词

1.4.1 谓词（Predicate）

命题逻辑并不能表达数学语言和自然语言中所有语句的确切含义。在数学表达式和计算机程序中经常可以看到含有变量的语句，例如 $x > 3$、$x = y + 3$、程序 x 正在运行等。当变量值未指定时，这些语句既不为真也不为假。

利用 $P(x)$ 可以表示语句，其中 x 是变量，语句 $P(x)$ 可以说是命题函数 P 在 x 的值。一旦给变量 x 赋一个值，语句 $P(x)$ 就称为命题并具有真值。

通常使用大写字母 P, Q, R 等表示谓词，小写字母 x, y, z 等表示变量。

Exercise 谓词

谓词	**真值**
$P(x) : x + 3 = 6$	$P(3)$ 为 True
$Q(x, y) : x = y + 2$	$Q(4, 1)$ 为 False

1.4.2 量词（Quantifier）

量词用量化表示在何种程度上谓词对于一定范围的个体成立。量词有全称量词（universal quantifier）和存在量词（existential quantifer）。

全称量词 $\forall$ 表示 all。$\forall_x P(x)$ 是一个命题，当范围内所有的 x 都能使语句 $P(x)$ 为真时，命题为真。

$$\forall_x P(x) = P(a_1) \land P(a_2) \land \cdots \land P(a_k)$$

Exercise 全称量词

假设 x 表示全班所有学生，$P(x)$ 表示 x 完成了作业。

$\forall_x P(x)$：全班所有学生都完成了作业。

存在量词 ∃ 表示 exists。$\exists_x P(x)$ 是一个命题，当范围内存在至少一个 x 能够语句 $P(x)$ 为真时，命题为真。

$$\exists_x P(x) = P(a_1) \vee P(a_2) \vee \cdots \vee P(a_k)$$

Exercise 存在量词

假设 x 表示全班所有学生，$P(x)$ 表示 x 完成了作业。

$\exists_x P(x)$：班里存在有一个学生完成了作业。

Exercise 嵌套量词

假设 x 表示某个人，$P(x)$ 表示有父母。

$\forall_x P(x)$：所有人都有父母。

$\exists_x \neg P(x)$：存在至少有一个人没有父母。

$\exists_x \exists_y (P(x) \wedge P(y))$：至少存在一个人 x 和一个人 y 有父母。

Exercise $P(x)$：x 是偶数，$Q(x)$：x 能被 3 整除，$x \in \mathbb{Z}^+$

语句	**真值**
$\exists_x (P(x) \wedge Q(x))$	True
$\forall_x (P(x) \rightarrow \neg Q(x))$	False

1.4.3 全称量词的否定

否定运算符可以使用在全称量词上。

$$\neg \forall_x P(x) \equiv \exists_x \neg P(x) \tag{1.20}$$

$$\neg \exists_x P(x) \equiv \forall_x \neg P(x) \tag{1.21}$$

Exercise 全称量词的否定

$P(x)$: x will pass the course (x is a student).

$\neg\forall_x P(x)$: Not all students will pass the course.

$\forall_x \neg P(x)$: No student will pass the course.

$\neg\exists_x P(x)$: There does not exist a student that will pass the course.

$\exists_x \neg P(x)$: There exists a student that will not pass the course.

1.5 证明

1.5.1 证明（Proof）

证明方法非常重要，不仅因为它们可用于证明数学定理，而且在计算机科学中也有许多应用，包括验证程序正确性、建立安全的操作系统、人工智能领域做推论等。证明就是建立定理真实性的有效论证。

证明定理有很多方法：

1. 直接证明法（direct proof）

   证明 如果 n 是奇数，那么 n^2 也是奇数，$n \in \mathbb{Z}$。

   $$\begin{aligned} n^2 &= (2k+1)^2 \\ &= 4k^2 + 4k + 1 \\ &= 2(2k^2 + 2k) + 1 \end{aligned}$$

2. 反证法（proof by contrapositive）：由于 $p \to q \equiv \neg q \to \neg p$，因此可以通过证明原命题的逆否命题来反证原命题。

   证明 如果 xy 是偶数，那么 x 是偶数或 y 是偶数，$x, y \in \mathbb{Z}$。
   逆否命题：如果 x 是奇数并且 y 是奇数，那么 xy 是奇数，$x, y \in \mathbb{Z}$。

   $$\begin{aligned} xy &= (2m+1)(2n+1) \\ &= 4mn + 2m + 2n + 1 \\ &= 2(2mn + m + n) + 1 \end{aligned}$$

1.6 布尔代数

1.6.1 布尔代数（Boolean Algebra）

计算机和其它电子设备中的电路都有输入和输出，输入是 0 或 1，输出也是 0 或 1。电路可以用任何具有两个不同状态的基本元件来构造，例如开关和光学装置就是这样的原件，开关可位于开或关的位置，光学装置可能是点亮或未点亮。18 世纪，乔治 · 布尔（George Boole）给出了逻辑的基本规则。

电路的操作可以用布尔函数来定义，这样的布尔函数对任意一组输入都能指出其输出的值。

布尔代数提供的是集合 $\{0,1\}$ 上的运算和规则，布尔代数的规则类似于命题逻辑的规则。1 相当于逻辑中的真，0 相当于逻辑中的假。

布尔代码运算主要有三种：

1. 补（complement）

x	$\overline{x}$
1	0
0	1

2. 布尔积（boolean multiplication）

x	y	$x \cdot y$
1	1	1
1	0	0
0	1	0
0	0	0

3. 布尔和（boolean addition）

x	y	$x+y$
1	1	1
1	0	1
0	1	1
0	0	0

布尔表达式 当 $x=y=1$，$w=z=0$ 时，

$x\cdot y+(w+z)=1\cdot 1+(0+0)=1+0=1$

$x\cdot\overline{y}+z\cdot\overline{(w+z)}=1\cdot\overline{1}+0\cdot\overline{(0+0)}=0+0=0$

$x\cdot\overline{y}+\overline{(\overline{x}+y+\overline{yz})}=1\cdot\overline{1}+\overline{(\overline{1}+1+\overline{1\cdot 0})}=0+\overline{1}=0$

1.6.2 布尔代数定理

幂等律 Idempotent Laws

$$x\cdot x=0 \tag{1.22}$$

$$x+x=x \tag{1.23}$$

恒等律 Identity Laws

$$x\cdot 1=x \tag{1.24}$$

$$x+0=x \tag{1.25}$$

支配律 Domination Laws

$$x\cdot 0=0 \tag{1.26}$$

$$x+1=1 \tag{1.27}$$

双非律 Double Negation Law

$$\overline{\overline{x}} = x \tag{1.28}$$

交换律 Commutative Laws

$$x \cdot y = y \cdot x \tag{1.29}$$

$$x + y = y + x \tag{1.30}$$

结合律 Associative Laws

$$(x \cdot y) \cdot z = x \cdot (y \cdot z) \tag{1.31}$$

$$(x + y) + z = x + (y + z) \tag{1.32}$$

分配律 Distributive Laws

$$x \cdot (y + z) = x \cdot y + x \cdot z \tag{1.33}$$

$$x + (y \cdot z) = (x + y) \cdot (x + z) \tag{1.34}$$

德摩根律 De Morgan's Laws

$$\overline{x \cdot y} = \overline{x} + \overline{y} \tag{1.35}$$

$$\overline{x + y} = \overline{x} \cdot \overline{y} \tag{1.36}$$

吸收律 Absorption Laws

$$x \cdot (x + y) = x \tag{1.37}$$

$$x + (x \cdot y) = x \tag{1.38}$$

证明 $xy + x\overline{y} = x$

$$xy + x\overline{y} \tag{1.39}$$

$$= x \cdot (y + \overline{y}) \tag{1.40}$$

$$= x \cdot 1 \tag{1.41}$$

$$= x \tag{1.42}$$

1.6.3 布尔函数（Boolean Function）

含有 n 个变量的布尔函数能够构造出 2^n 行的输入输出表。

Exercise 计算 $F(x, y, z) = xy + \overline{z}$

x	y	z	$F(x,y,z)$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

1.6.4 NAND 与 NOR

NAND 运算符用 $\uparrow$ 表示 Not And：

$$x \uparrow y = \overline{x \cdot y}$$

NOR 运算符用 $\downarrow$ 表示 Not Or：

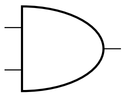
$$x \downarrow y = \overline{x + y}$$

1.7 逻辑门电路

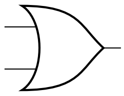
1.7.1 逻辑门电路（Logical Gate Circuit）

基础的逻辑门主要有三种：

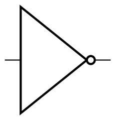
1. 与门（AND gate）



2. 或门（OR gate）：



3. 非门（NOT gate）：



Exercise 设计一个投票表决电路，三个人中有两人赞成即通过。赞成票为 1，否决表为 0。

$$F(x, y, z) = xy + xz + yz$$