一. 逻辑和证明

1. 可判别真伪的语句(sentence) → 命题(proposition)（原子命题） → 复合命题(compound propositions)
2. 命题变量(propositional variables) $\xrightarrow{\text{赋值}}$ 真值(truth value) → 真值表

3. 操作联结词(connectives)

- 取反(negation) $\neg$
- $\vee$ or (disjunction) 也叫(inclusive or)
- $\wedge$ and (conjunction)
- $\rightarrow$ conditional $p \rightarrow q$
  - 还有一些特殊语句表示 → p_6
  - only if / if then / when
  - ↳ implies

T	T	T
T	F	F
F	T	T
F	F	T

当一个q为F时为F

- $\oplus$ 异或 exclusive or (最多1个)
- $\leftrightarrow$ biconditional $p \leftrightarrow q$ 相同的输入必然得到相同的（真值）输出.
  - ↳ iff 同

4. 逆命题 converse；逆否 contrapositive；反命题 inverse

对 $p\rightarrow q$ 有 $p \rightarrow q \xrightarrow{\text{逆}} q \rightarrow p$

反：$\neg p \rightarrow \neg q$

逆否：$\neg q \rightarrow \neg p$

5. 优先级 (precedence)

6 逻辑和比特(bit)运算：用0/1代替F/T的运算，也叫按位运算

7. 逻辑命题的应用 建议全英读课本

8. 命题等价(equivalences)

8.1
- 永真式(tautology)
- 矛盾式(contradiction) 永远为假
- 可能式(contingency)

8.2 逻辑等价

① 若 $p \leftrightarrow q$ 为永真式，则 p与q逻辑等价，$p \equiv q$

② 德·摩根律(De Morgan laws) $\neg(p \vee q) \equiv \neg p \wedge \neg q$

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

③ 常用的逻辑等价式

P25

$p \wedge T \equiv p$
$p \vee F \equiv p$
Identity laws

$p \vee (p \wedge q) \equiv p$
$p \wedge (p \vee q) \equiv p$
absorption laws

$(p \to r) \overset{\wedge}{\vee} (q \to r) \Leftrightarrow (p \overset{\vee}{\wedge} q \to r)$

$(r \to p) \overset{\wedge}{\vee} (r \to q) \Leftrightarrow r \to (p \overset{\vee}{\wedge} q)$

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

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

$p \oplus q \Leftrightarrow (p \vee q) \wedge \neg(p \wedge q)$

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

9 书写逻辑式
- 证明永真
- 证明等价
- 证明矛盾

10 可满足性(satisfiable)：永真 V 可能 (只要有1个真即可)

不可满足 (unsatisfiable)

11 可满足性的应用——N皇后问题：构建逻辑命题并求证

12 谓词(predicates)和量词(quantifiers)

12.1 { ① 对比前面提到的命题变量分配真值，谓词把命题分配为变量+谓词
② n元谓词：语句有多少变量

12.1.1 变量 { 约束 bounded
自由 free

12.2 前置条件(precondition)，后置条件(postcondition)

12.3 量词 { 全称量词(universe of discourse) $\forall$
论域(domain of discourse)
存在量词(existential) $\exists$

(P(x)) (Q(x))
一些学生会做工 $\exists x(P(x)\land Q(x))$
所有学生会打工 $\forall x(P(x)\to Q(x))$

$\forall x>0(x^2>1)$ 表示实数域中，$\forall x>0$，有 $x^2>1$

12.4 逻辑等价式

$\forall x(P(x)\land Q(x)) \equiv \forall x P(x) \land \forall x Q(x)$

$\exists x(P(x)\lor Q(x)) \equiv \exists x P(x) \lor \exists x Q(x)$

$\forall x(P(x)\lor Q(x)) \Leftarrow \forall x P(x) \lor \forall x Q(x)$

$A \to \forall x B(x) \equiv \forall x(A\to B(x))$

$\exists x(P(x)\land Q(x)) \Leftarrow \exists x P(x) \land \exists x Q(x)$

$\forall x(P\lor Q(x)) \equiv P \lor \forall x Q(x)$

12.5 量词否定 { $\neg \forall x P(x) \equiv \exists x \neg P(x)$
$\neg \exists x P(x) \equiv \forall x \neg P(x)$

12.6 量词顺序：从左到右

13 推理规则

表 1 推理规则

推理规则	永真式	名称
p $\underline{p \to q}$ $\therefore q$	$(p \wedge (p \to q)) \to q$	假言推理
$\neg q$ $\underline{p \to q}$ $\therefore \neg p$	$(\neg q \wedge (p \to q)) \to \neg p$	取拒式
$p \to q$ $\underline{q \to r}$ $\therefore p \to r$	$((p \to q) \wedge (q \to r)) \to (p \to r)$	假言三段论
$p \vee q$ $\underline{\neg p}$ $\therefore q$	$((p \vee q) \wedge \neg p) \to q$	析取三段论
$\underline{p}$ $\therefore p \vee q$	$p \to (p \vee q)$	附加律
$\underline{p \wedge q}$ $\therefore p$	$(p \wedge q) \to p$	化简律
p $\underline{q}$ $\therefore p \wedge q$	$((p) \wedge (q)) \to (p \wedge q)$	合取律
$p \vee q$ $\underline{\neg p \vee r}$ $\therefore q \vee r$	$((p \vee q) \wedge (\neg p \vee r)) \to (q \vee r)$	消解律

记住推理规则与名字即可,

13.2 谬误 (fallacies)

表 2 量化命题的推理规则

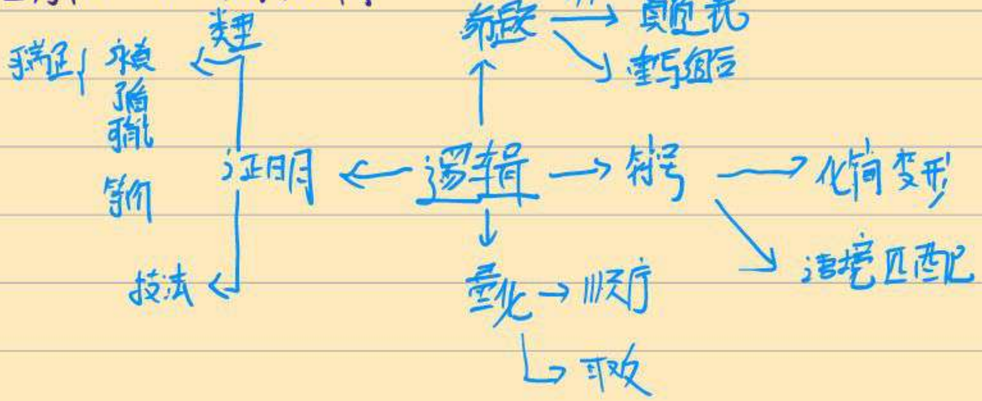
推理规则	名称
$\underline{\forall x P(x)}$ $\therefore P(c)$	全称实例
$\underline{P(c)\text{，任意 } c}$ $\therefore \forall x P(x)$	全称引入
$\underline{\exists x P(x)}$ $\therefore P(c)\text{，对某个元素 } c$	存在实例
$\underline{P(c)\text{，对某个元素 } c}$ $\therefore \exists x P(x)$	存在引入

14 定理 (theorem), 公理 (axiom) 引理 (lemma)

15 证明定理的方法

L 直接证明法, 反证法, 归谬证明法

L 穷举证明 分情形证明



二.基本结构

1. 集合(set)

1.1 元素(element) 成员(member)

1.2 N 自然数集 Z 整数集 Z^+ 正整数集 Q 有理数集 R 实数集 C

1.3 操作

- ① $a \in A$ $a \notin A$ a为元素,加{}
- ② $A \subseteq B$ / $A \subset B$ A为集合,加{}
- ③ $|A|$ 个数(元素不得相同) 基数(cardinality)
- ④ $A = B \Rightarrow A \subseteq B, B \subseteq A$

1.4 种类

- ① 空集(null set)
- ② 单元素集 (singleton set)
- ③ 子集 (subsets) / 超集 (superset) $\subseteq$
- ④ 真子集 (proper subset) $\subset$
- ⑤ 幂集(power set) 表示集合S的所有子集的集合

2. 笛卡儿积(cartesian products)

2.1 有序n元组(ordered n-tuple)

2.2 $A \times B = \{(a,b) \mid a \in A, b \in B\}$

$A = \{a_1, a_2, a_3\}$ $B = \{b_1, b_2, b_3\}$

$A \times B = \{(a_1,b_1)\ (a_1,b_2)\ (a_1,b_3), (a_2,b_1) \cdots\}$

有序指前后有序,而不是大小有序

3 运算(operations)

$$\left.\begin{array}{l} A \cup B \\ A \cap B \end{array}\right\} |A \cup B| = |A| + |B| - |A \cap B|$$

$A - B$ (difference)

$\bar{A}$ (相对于全集U (universal set))

4. 集合恒等式 (set identities)

$$\begin{cases} \overline{A\cup B} = \bar{A}\cap\bar{B} \\ \overline{A\cap B} = \bar{A}\cup\bar{B} \end{cases}$$

5. 并/交 $\bigcup_{i=1}^{n} A_i$ / $\bigcap_{i=1}^{n} A_i$

6. 集合的按位表示 (string) $A=\{1,2,4,5\}$ 11011

7. ==函数== (function) 指派 (assigned) 映射 (mapping)

7.1 定义域 (domain) 陪域 (codomain)

$f(a)=b$ 象 (image)

定义域 → A 原象 (preimage) B ← 陪域

$A \to f \to B'$ 值域 (range) $B' \subseteq B$

f: 也代表了一种集合, 也为有序对

8 关系

xRy 表示为笛卡儿积 (x,y)

① 一对一/单射 (one-to-one / injection) $\forall x\in A$ 若 $f(a)=f(b)$, 则 $a=b$

② 递增/递减 (increasing / decreasing)

③ 映上/满射 (onto/surjection) $B'=B$

④ 双射/一一对应 (bijective) (一对一)+满射

⑤ 反函数 (inverse function) f^{-1}, 双射起手

9. 反函数和函数合成 (inverse function + compositions of functions)

可逆的 (invertible) = 一一对应关系

不可逆的 (not invertible)

9.1 复合(composition) $f \circ g$

$A \xrightarrow{g} B \xrightarrow{f} C$ g代入f

记 $f \circ g$

9.2 重要函数

下取整 $\lfloor \ \rfloor$ (floor)
上取整 $\lceil \ \rceil$ (ceiling)

9.3 部分函数 (partial functions)

$A' \subseteq A$ $A' \xrightarrow{f} B$ $B \subseteq B$

domain codomain

全函数(total function), 当 $A' = A$

10 序列和求和 (sequences and summations)

$\{a_n\}$, a_n 为项(term)

10.1 几何级数 (geometric progression) 形式(sequence)

$f(x) = ar^x$

10.2 算术级数 (arithmetic progression)

$f(x) = dx + a$

串(string) 空串(empty string)

10.3 递推 (Recurrence relations)

斐波那契 (fibonacci sequence) $f_n = f_{n-1} + f_{n-2}$

10.4 求和 $\sum a_i$

11 集合的基数 (cardinality of sets)

$|A|$ 集合A的基数

可数集合(countable)

不可数集合(uncountable)

可计算的(computable)

不可计算的(uncomputable) } 可被计算机计算的

12 矩阵(matrices)

方阵(square),元素,项(entry)

12.1 算术

① $A+B \ / \ A-B$

② $A: \overset{3}{m} \times \overset{2}{n} \quad B: \overset{2}{n} \times \overset{3}{m}$

$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \\ a_5 & a_6 \end{bmatrix} \times \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \end{bmatrix}$$

$$= \begin{bmatrix} a_1b_1+a_2b_4, & a_1b_2+a_2b_5, & a_1b_3+a_2b_6 \\ a_3b_1+a_4b_4, & \dots & \\ \dots & & \end{bmatrix}$$

3×3

12.2 转置(transposes) 幂(powers)

对称的(symmetric) $A=A^T$

12.2.1 转置

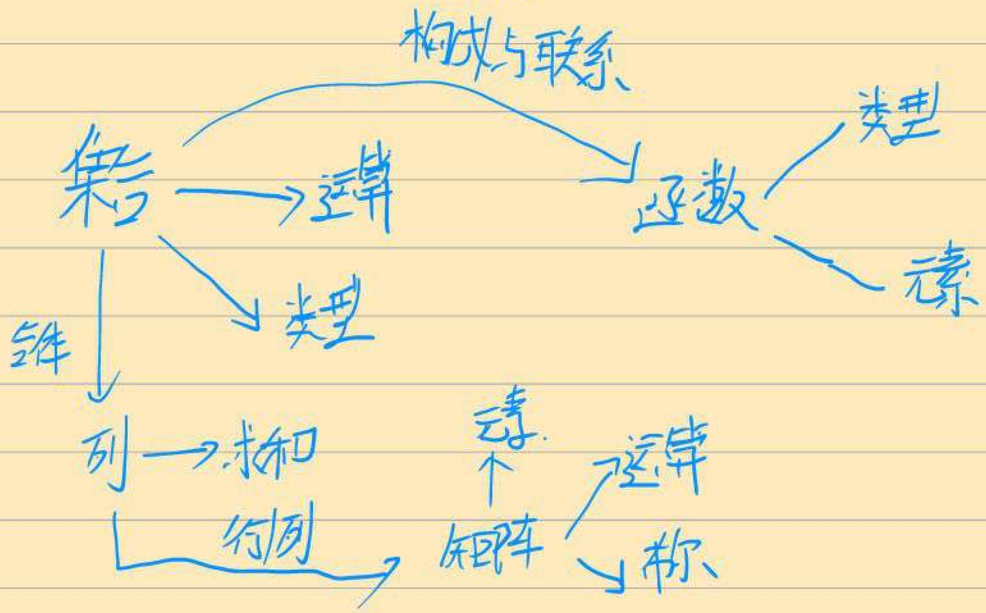
$$A=[a_{11}, a_{12}] \xrightarrow[\text{互换}]{i/j} A^T=\begin{bmatrix} a_{11} \\ \vdots \\ a_{21} \end{bmatrix}$$

12.2.2 并(join) 对0-1矩阵按位操作

交(meet)

12.2.3 布尔积 $\odot$ (boolean product) $A \odot B$ 正常乘，但元素 $\times \to \wedge$
$+ \to \vee$

12.3 幂 $A^{[r]} = \underbrace{A \odot A \odot A \cdots}_{r个A}$



六 计数

1. 基础计数原则

1.1 乘积法则(product rule)和加法规则(sum rule)

1.1.1 先后 情况按乘法计算总情况

1.1.2 左右 情况按加法计算总情况

1.2 减法法则(subtration)(容斥)

即 $|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$

1.3 除法法则(division)

即对一个事件求出所有的基础类别后组合成实际单位事件

若: 4个人坐在一张圆桌边, 有多少种坐法

基础类别有 $4\times3\times2\times1=24$, 一个实际单位事件有4个类别

故有坐法 $n=24/4=6$

1.4 树图(tree diagrams)

2. 鸽巢原理(pigeonhole principle)

若 $\geq k+1$ 个物体放入 k 个盒子, 则至少有一个盒子装了2个或更多的物体

3. 排列组合(permutations and combinations)

前者在于顺序而非分布, 后者在于分布而非顺序

n个元素组合为r的, 有 $C(n+r-1, r)$ 种方式

留待补充

4. 二项式定理 (binomial theorem)

$(x+y)^n = C_n^i x^{n-i} y^i \quad (0 \le i \le n)$

4.1 $C_{n+1}^k = C_n^{k-1} + C_n^k$

8 高级计数技术

1. 递推 (recurrence)

我们使用递推关系和初始条件构建模型求解

1.1 斐波那契数

1.2 汉诺塔 (hanoi tower)

2. 动态规划 (   )

3. 求解线性递推关系 (linear recurrence)

3.1 常系数 (homogeneous)

$r^n = c_1 r^{n-1} + c_2 r^{n-2} + \cdots + c_k r^{n-k}$，有k个解

特征方程 (characteristic equation) $r^k - c_1 r^{k-1} - c_2 r^{k-2} - \cdots - c_k = 0$

特征根 (characteristic rooting) 序列 $\{a_n = r^k\}$

3.1.1 两根不同

对递推关系 $a_n = c_1 a_{n-1} + c_2 a_{n-2}$

有特征方程 $r^2 - c_1 r - c_2 = 0$，有根 r_1, r_2

则有 $a_n = c' r_1^n + c'' r_2^n$

根据原始条件，代入n，求出 c', c''，得到 a_n

3.1.2 两根相同

则式变为 $a_n = a_1 r_1^n + a_2 \cdot n \cdot r_1^n$

3.1.3 多根不同

若有特征方程 $r^k - c_1 r^{k-1} - \cdots - c_k = 0$

则 $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \cdots \alpha_k r_k^n$

同样步骤，代入求解

3.1.4 k个根只有t个不相异，对每个根 t_i 有其重数 m_i

$$r^k - c_1 r^{k-1} - \cdots - c_k = 0$$

$$a_n = (a_{1,0} + a_{1,1} n + \cdots + a_{1,m_1-1} n^{m_1-1}) r_1^n$$
$$+ (a_{2,0} + a_{2,1} n + \cdots + a_{2,m_2-1} n^{m_2-1}) r_2^n$$
$$\cdots$$

3.1.5 非齐次 (nonhomogeneous)

对 $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n)$

解为 a_n = 齐次解 + 特解 $= a_n^{(p)} + a_n^{(h)}$

$$F(n) = (b_t n^t + b_{t-1} n^{t-1} + \cdots + b_1 n + b_0) s^n$$

① 若 s 非相伴的齐次解，则特解有形式

$$a_n^p = (p_t n^t + p_{t-1} n^{t-1} + \cdots + p_1 n + p_0) s^n$$

代入原式求出各常数的值.

② 若 s 是，且重数为 m，则

$$a_n^p = n^m (p_t n^t + p_{t-1} n^{t-1} + \cdots + p_1 n + p_0) s^n$$

如 $a_n = 3a_{n-1} + 2n$

① $a_n = 3a_{n-1}$ ② $a_n = 3a_{n-1} + 2n$，代入 $a_n = cn + d$

$$\underbrace{a_n^h = a 3^n \qquad a_n^p = -n - \tfrac{3}{2}}_{a_n = -n - \frac{3}{2} + a \cdot 3^n}$$

4. 分治 (divide and conquer)

分析时间复杂度

① $f(n)$ 为增函数，n 被 b 整除. $f(n) = af(n/b) + C$

- 若 $a=1$ 则 $f(n) = O(\log n)$
- 若 $a>1$ 则 $f(n) = O(n^{\log_b a})$
- 若 $n=b^k$，$a \neq 1$ 时 $f(n) = C_1 n^{\log_b a} + C_2$
  - $C_1 = f(1) + C/(a-1)$
  - $C_2 = -C/(a-1)$

① [illegible] ② [illegible]

② $f(n)$ 为增函数，$n=b^k$，b 非负

$$f(n) = af(n/b) + cn^d$$

$$f(n) \text{ 为 } \begin{cases} O(n^d) & a < b^d \\ O(n^d \log n) & a = b^d \\ O(n^{\log_b a}) & a > b^d \end{cases}$$

5 生成函数 (generating functions)

5.1

对一个序列 $\{a_n\}$ ($0 \le n$) 有其生成函数 $G(x) = a_0 + a_1 x + \cdots + a_n x^n$

求和

$$f(x) = \sum_{k=0}^{\infty} a_k x^k, \quad g(x) = \sum_{k=0}^{\infty} b_k x^k$$

$$f(x) + g(x) = \sum_{k=0}^{\infty} (a_k + b_k) x^k$$

$$f(x) g(x) = \sum_{k=0}^{\infty} \left(\sum_{j=0}^{k} a_j b_{k-j} \right) x^k$$

在一个区间内收敛才有效

5.2

二项式 $\binom{u}{k} = \dfrac{\overbrace{u(u-1)(u-2)\cdots}^{k\text{项}}}{k!}$

$$(1+x)^u = \sum_{k=0}^{\infty} \binom{u}{k} x^k$$

应用

① 化为生成函数：对一问题，若需将情况列出并方程组，求解出特殊组合的数量，就可以使用生成函数

如 $x+y=3$ 　$2\le x\le 4$，$-1\le y\le 2$，x 与 y 为整数

则 x 的所有情况为 $-\infty$ 到 $+\infty$，加上范围为 2, 3, 4，生成函数

$G_x = x^2+x^3+x^4$，同理 $G_y = x^{-1}+1+x+x^2$

我们将 G_x 与 G_y 相乘 $(G_x \cdot G_y)$ 得到 x^3 那项的系数即为答案

② 求解递推关系 P58

6 容斥 (inclusion-exclusion)

即两集合之间可能出现的重合部分

$|A\cup B| = |A|+|B|-|A\cap B|$

$$|A_1\cup A_2\cup\cdots\cup A_n| = \underbrace{\sum_{i=1}^{n}|A_i|}_{\text{一项}} - \underbrace{\sum_{i=1}^{n}\sum_{j=1}^{i}|A_i\cap A_j|}_{\text{二项}} + \underbrace{\sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{k=1}^{j}|A_i\cap A_j\cap A_k|}_{\text{三项}} - (-1)^{n+1}\cdots$$

如此递推

齐次

求解递推 (a_n=?a_{n-1})

非齐次

干扰项

计数 → 容斥 → 概率

生成函数

无穷级数

留待补充

九.关系

1.关系 (relation) ARB

1.1 性质

① A上的(on)的关系 ARA

② 自反(reflexive) $a\in A$ 有$(a,a)\in R$

③ 对称(symmetric) 若(a,b),则(b,a)

④ 反对称(antisymmetric) (a,b)且(b,a) 则 $a=b$(即仅允许同为)

⑤ 传递(transitive) $(a,b)+(b,c)$ 则 (a,c)

⑥ 关系本质上是集合,不过一般是有序2元组

注: $\neg R$ 表示取R的矩阵中非的部分 R^{-1} 则是其矩阵的转置

1.2 组合 (combining)

① 支持一般集合的 $\cup\ \cap\ -$ 运算

② $A \xrightarrow{R} B \xrightarrow{S} C$ 组成,注意符号顺序

$$\begin{tikzcd} A \arrow[r, "R"] \arrow[rr, bend right, "S\circ R"'] & B \arrow[r, "S"] & C \end{tikzcd}$$

③ 幂 $R^n = R^{n-1}\circ R$ $R^1=R$

④ 传递 $\Longleftrightarrow$ $R^n\subseteq R$

2. n元关系 (n-ary)

2.1 从n个集合中得到的关系R,称为n元关系,n称为阶(degree),集合称为域(domain)

2.2 数据库

① 所有被数据库包括的所有数据,称为外延(extension)

② 外延的属性称为内涵(intension)

③ 主键(primary key) 由单个域标识

④ 复合键(composite key) 由多个域的笛卡儿积得到

⑤ 域(fields),由n个集合组成,标识了一种数据

2.3 运算 (SQL)

① S_C 在n元关系R中满足n元关系C的集合

② 投影(projection) $P_{i_1 i_2 \cdots i_m}$，删除了关系R中除了 $i_1 \sim i_m$ 外的记录

③ 连接(join) $J_p(R,S)$ 将关系R后面的p项与S前面的p项合并然后得到的 $m+n-p$ 个项的新关系

3 矩阵表示 (matrices)

① 自反 → 矩阵对角线全为1

② 对称 → 若 $a_{ij}=1$ 则 a_{ji} 也为1

③ 反对称 → a_{ij} 与 a_{ji} 不同时为1

④ $M_{S\circ R} = M_R \odot M_S$

4. 图表示 (digraphs)

① 环(loop) 指向自己的路径 圈(cycle/circuit) 回到自己的路径

② 自反：即环

③ 对称：对应的两节点为双向箭头

④ 反对称： ↓ 单向

⑤ 传递

5. 闭包 (closures of relation)

通过添加元素对来使得集合符合某个特性

自反闭包，对称闭包

① 路径(path) a与b的路径仅经过a b点且无重复点

② 若 $(a,b)\in R^n$，则从a到b存在一条长为n的路径

③ 连通性关系 (connectivity relationship) R^* 使得在R中a与b间存在一条长度至少为1的路径

$$R^* = \bigcup_{n=1}^{\infty} R^n$$

5.1 R^*

① 关系R的传递闭包记 R^*

② 若图中存在一条从a到b的长度至少为1的路径，此路径长度不超过 n (n为整数)

若a≠b，则为 n-1

③ 矩阵表示 $M_{R^*} = M_R \vee M_R^{[2]} \vee \cdots \vee M_R^{[n]}$

5.2 求出 R^*

① $M_{R^*} = M_R \vee M_R^{[2]} \vee \cdots \vee M_R^{[n]}$ → $M_R^{[2]} = M_{R^2} = M_R \odot M_R$ (M记法)

② 沃舍尔(warshall)算法

2.1 关系R的初始矩阵 $W_0 = M_R$，求 $W_n = M_R$ (W记法)

：在 W_0 中寻找从i到j的路径，若其中节点在 $\{\phi, w_1, w_2 \ldots w_k\}$ 中，则在 W_k 中 $w_{ij}=1$ 如A={a,b,c,d}，则 $w_1=a$, $w_2=b$ $w_3=c$ $w_4=d$

从 W_0 的图 推导出 W_i

2.2 已知 W_{k-1} 求 W_k

在 W_{k-1} 的图中，寻找 w_{ij} 的路径 { 中间路径在 $\{\phi, w_1 \cdots w_{k-1}\}$ 中；中间路径最多1个 w_k，其余均在其中

⇓ 导出

2.3 $w_{ij}^{[k]} = w_{ij}^{[k-1]} \vee (w_{ik}^{[k-1]} \wedge w_{kj}^{[k-1]})$

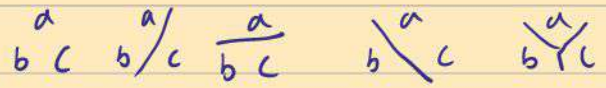
6. 等价关系 (equivalence)

① 定义在集合上的关系R，若有自反+传递+对称，则是等价关系

② 由等价关系联系在一起的所有元素是一个等价类. 多个等价类被等价关系用来划分集合 (partitions)

③ a~b 表示a与b等价

④ 各个等价类彼此不相交

⑤ 等价关系由等价类的划分而得到 

7. 偏序 (partial orderings) (posets)

① 自反+反对称+传递，偏序集(S,R)

② 用符号 ≼ 表示具有以上3个特性的关系，≼ 左边的视为元素

③ 集合S中的元素都可进行比较的 称为全序集(totally ordered) $\preccurlyeq$ 称为全序(total order) →任意两元素有连线,能连成一条直线

④ 若$(S,\preccurlyeq)$是全序,且对$\forall a\in S$的任意非空子集,必有$b\in$它对a,有 $a\preccurlyeq b$,则是良序(well-ordered)

⑤ 良序归纳: $\forall y\in S$,若$x\in S$且$x\prec y$ 则$P(y)$为真,则$P(x)$对所有$x\in S$为真

x为最小元素　　即由哈塞图大小

7.1 字典序(lexicographic order)

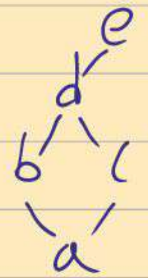
7.2 哈塞图(hasse diagrams)

① 用连线表示偏序关系.

② 顺序为从下到上,从小到大

③ 覆盖(covers)

若$a\preccurlyeq b$ 则a放在下面,称为元素



有 ① $a<(b,c)$

② $d>(b,c)$

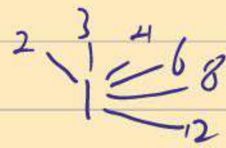
③ $e>d$

7.2.1 构建哈塞图

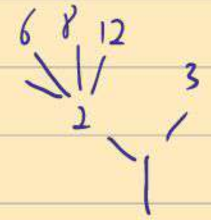
{1, 2, 3, 4, 6, 8, 12}　a整除b

① 从小到大,从下到上

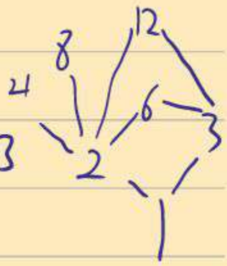
对1



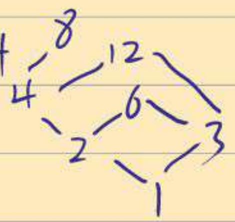
对2



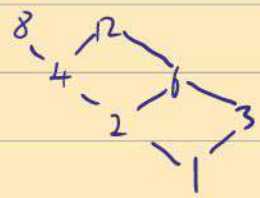
对3



对4



对6



对8 对12 不变

即注重顺序和沿线的继承

7.2.2 极大元(maximal),极小元(minimal) → 不唯一

b再无新元素 b使得 $a<b$,则a为maximal;没有新元素b使得 $b<a$

7.2.3 最大元(greatest element) 最小元(least element) 唯一,或没有最值

7.2.4 上界(upper bound)，下界(lower bound)

最大下界(greatest lower bound) 最小上界(least upper bound)

给出一子集，图中所有大于等于子集中最大元素构成的集合为上界，上界的的最小元素是最小上界，上界构成上界集合，上界中可以存在子集中的元素

7.3 格(lattices)

格：一种特殊的哈塞图，若偏序中任意两个元素都有最小上界与最大下界

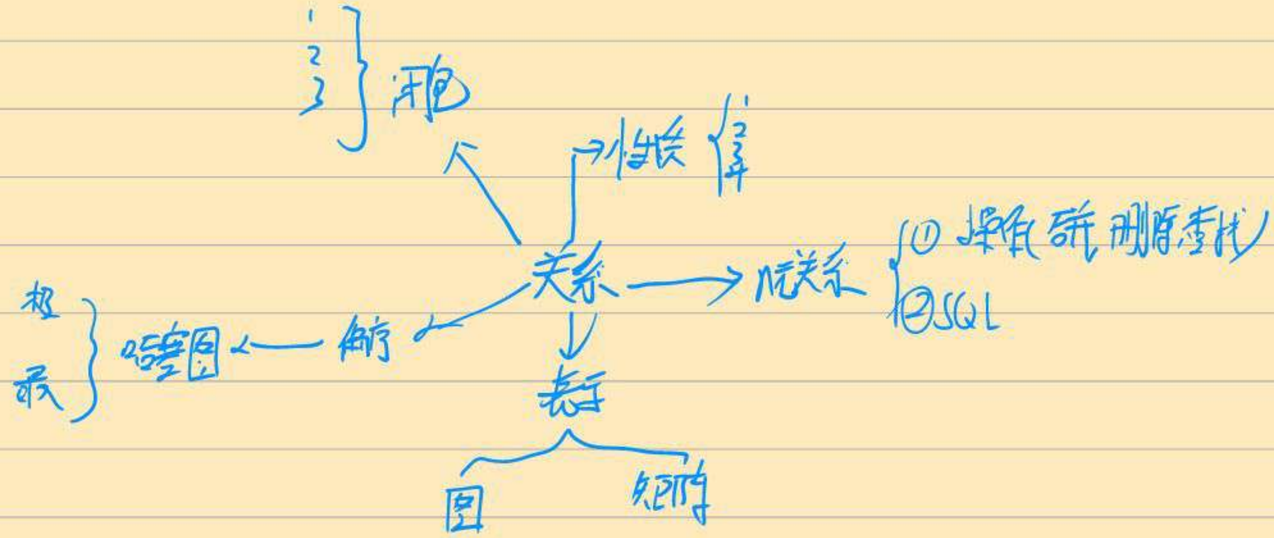
完全格：对任意一个节点子集有格的性质

7.4 拓扑排序(topological sorting)

每次弹出哈塞图中的1个极小元，直到弹完

形成 $a_1 < a_2 < a_3 < \cdots$

相容的(compatible)：对一个偏序R，其中若有全序≤，满足只要 aRb 就有 $a \leq b$（不一定 $a \leq b$ 有 aRb），就称为相容的



10 图(Graphs)

1. 图 $G=(V,E)$

  1.1 无向图 (undirected graphs)

    1.1.1 简单图 (simple graph)
    无向,单边,无环

    1.1.2 多重图 (multiple graph)
    无向,多边,无环

    1.1.3 环 (loop) 指向自身的边

    1.1.4 伪图 (pseudographs)
    多边,有环,无向

  1.2 有向图 (directed graph)

    1.2.1 简单有向图 & 有向多重图
    有向,单边,无环　　有向,多边,有环

2. 性质

  2.1 节点 a 的相邻节点 $N(a)$,节点集 A 的相邻节点 $N(A)=\bigcup_{a\in A} N(a)$

  2.2 度(degree)
  与节点连接的边数
  阶→节点数

  ① 无向图-度: $deg(a)$
    - $deg(a)=0$, a 是孤立的(isolated)
    - $deg(a)=1$, a 是悬挂的(pendant)

  ② 入度 $deg^-(a)$　$\to a$

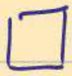
  ③ 出度 $deg^+(a)$　$a\to$

  2.3 握手定理(handshaking)
  无向图中: $|E|=m$, 则 $2m=\sum_{a\in V} deg(a)$, 度数和应是偶数
  有向图中: $2m=\sum_{a\in V} deg^+(a)+\sum_{a\in V} deg^-(a)$

  $$\sum_{a\in V} deg^+(a)=\sum_{a\in V} deg^-(a)=m$$

3. 特殊图

  3.1 完全图 K_n 
  K4
  n为节点数, $\forall x,y$, 均有连线

3.2 圈图 C_n  $n=4$ 圈

3.3 轮图 W_n  $n=4$ 圈中加一点与连线

3.4 n立方体图 Q_n  $n=3$ 3立方体图

4. 二分图 (bipartite graphs)

$G=(V,E)$ 中 V 可以被分为两个不相交的子集 V_1 与 V_2，又在 V_1 与 V_2 间有连线

任意一个回路的长度为偶数（充要条件）

4.1 着色定理判别二分图：一条边连接的两个顶点必为不同颜色

4.2 完全二分图 (complete bipartite graphs)

V_1 与 V_2 中的每个顶点之间都有连线

$K_{m,n}$ m与n个节点

4.3 匹配 (matching)

匹配M是边E的子集，M中每条边对应的节点在整个M中只出现一次，

最大匹配 (maximum matching) 最大子集M

完全匹配 (complete matching) ① 对划分 V_1 与 V_2，若 V_1 中所有的顶点都在M中边的端点集中，则有 V_1 到 V_2 的完全匹配

② 对 V_1 的所有子集A，若 $|N(A)| \geq |A|$

4.4 子图 (subgraph) $H \subset G=(V,E)$ 真子图 (proper graph)

4.5 操作
- 删除 (removing) 只删边
- 增加 (adding) 只增边
- 收缩 (contractions) 删边，两个端点合并
- 其他删除 删点和对应的边

并图

5. 图的表示

5.1 邻接表 (adjacency lists) 记录相连的点

5.2 邻接矩阵 (adjacency matrices) 用 |V|×|V| 的矩阵表示边的数目

5.3 稀疏图 (sparse) 稠密图 (dense)

5.4 关联矩阵 (incidence matrices) 用 |V|×|E| 的矩阵表示关系

无向图 / 1, 0

有向图 入度记 -1 / 出度记 1 / 0

5.5 图的同构 (isomorphism) 对两个图 G_1, G_2，G_1 的 V_1 映射到 G_2 的 V_2，则 E_1 的映射到 E_2 不变

6. 连通性 (connectivity)

6.1 通路 (path) (两点间边的序列)

6.2 回路 (circuit) 一点间边的序列

} 若有重边或点，则是非简单的 路/边

6.3 连通性

6.3.1 无向图

① 任意两点间有通路

② 割边 (cut edge / bridge) 边

③ 割点 (cut vertices / articulation points) 删除此点及其边后，图不连通

④ 边割集 (edge cut) 割边使得图不连通

⑤ 点连通度 $\kappa(G)$ (vertex connectivity)

割点集中最小顶点数 → 使图变为不连通图的最小点数

不可分割图 (nonseparable graphs) 无割点

k 连通 (k-connected) $\kappa(G) \geq k$

双连通 (biconnected) 不可分割 + $|V| \geq 3$

⑥ 边连通度 (edge connectivity) $\lambda(G)$ 同上

6.3.2 有向图

① 强连通性(strongly connected) 有向图任意两节点有通路
弱连通性(weakly connected) 有向图的无向图是连通的

② 强连通分支(strong components): 是极大的子图,即任意加入一个节点都会破坏其连通性

方法: 任找一个节点,若此节点与其他节点连通,则加入此节点,直到找不到新节点,则此节点就是一个强连通分支

6.4 同构(Isomorphism)

① 顶点数相同
② 路径数相同
③ 边对应
④ 边的关系相同

通路数计算

对图的邻接矩阵A,从点v_i到v_j的长度为r的不同通路数 = A^r的第(i,j)项

7. 欧拉通路(euler)和哈密顿通路(Hamilton)

7.1 欧拉

7.1.1 欧拉回路(euler circuit): 包含图全部边且仅包含一次的回路

7.1.2 欧拉通路(euler path): 包含图全部边且仅包含一次的通路

7.1.3 判断:
↓
无向的
不限制简单图

无向图:
- 至少有2个顶点的连通多重图 具有欧拉回路 <=> 度为偶数
- 除有两个度为奇数的顶点外,其他的均为偶数 -> 欧拉通路

有向图:
- ① 首要条件: 图是连通的
- ② 对每个顶点 $deg^{-}=deg^{+}$ ⟹ 回路
- 有两个顶点a,b: $deg^{-}(a)-1=deg^{+}(a)$, $deg^{-}(b)=deg^{+}(b)-1$

7.2 哈密顿

7.2.1 hamilton 回路: 经过每个顶点恰好一次的回路

7.2.2 hamilton 通路: 经过每个顶点恰好一次的通路

7.2.3

判断: 无向图
↓
无向的, 简单图的

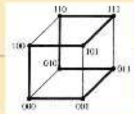
① n个顶点的简单图, $n \geq 3$, $\forall a \in V$, $\deg a \geq \frac{n}{2}$ → 回路 Dirac's

② n个顶点的简单图 $n \geq 3$ 对每一对不相邻的顶点 u和v, 都有 $\deg(u) + \deg(v) \geq n$ → 回路 Ore's

③ 回路 → 通路 (所有点的度都至少为2)

7.2.4 应用 ① 格雷码(gray code)

n位格雷码指每相邻两位串只有1位不同, 而位串有n位, 如00, 01, 11, 10



其hamilton 回路就是n位格雷码, 但顶点的排序还需求化

8 最短路(shortest-path)

通路长度: 通路边权之和

最短通路算法 { $O(n^2)$次运算, 为最短路径 } dijkstra

旅行商问题(traveling salesperson)

以最短路径访问每个节点一次且最后一次回到原点 — hamilton 回路

无准确解, 但有近似解

9 平面图 (planar graphs)

应用到平面上, 线段不能交叉的路, 我们说一个图是平面图是指其可以化为平面图, 又指其是平面图

判断

① 欧拉 (euler)

初始条件: G有e条边和v个顶点的连通平面简单图, r是平面图表示的面数, 则 $r = e - v + 2$

① 若 $v \geq 3$, 则 $e \leq 3v - 6$

② 若G是平面图, 则 $\exists v \in V$, $\deg v \leq 5$

若 $|V| \geq 3$ 且无长度为3的环, 则 $|E| \leq 2|V| - 4$

② 库拉图斯基(kuratowski)

① 一个平面图通过删除边{u,v}而新增点w和边{u,w},{w,v}而构成的新图仍是平面图,称为初等细分(elementary subdivision)
② 一个平面图通过初等细分得到的图是其同胚图(homeomorphic)
③ 非平面图都可以通过$K_{3,3}$或K_5同胚得到
的子图

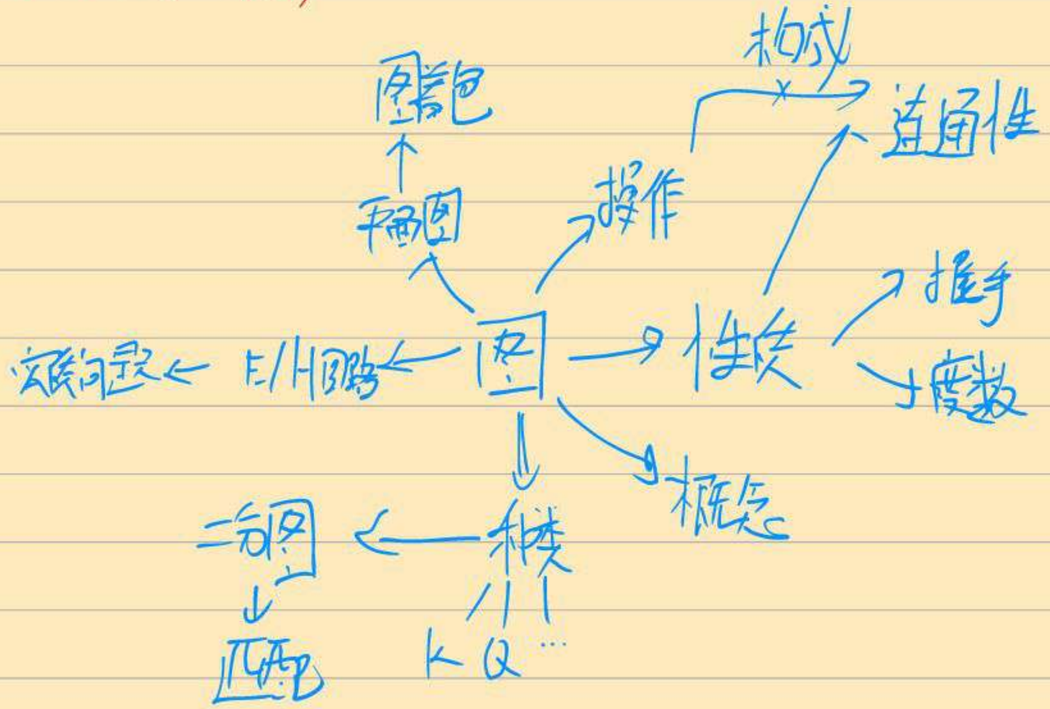
10. 图着色(coloring)

对平面图不同的区域填充颜色,相邻区域颜色不一致,再将区域替换成点,相邻关系替换成线,得到的图称为对偶图(dual graph)

10.1 图的着色数用$\chi(G)$表示,~~平面图的着色数不超过~~4,四色定理 着色数(chromatic)

10.2 图着色的应用
① 课程安排
② ...

N K λ $\chi(G)$



11 树

1.概述

1.1 ①树是没有回路的简单无向图
②森林：树的集合 (forest)

1.2 根(root) → 有根树

孩子(child) 父母(parent) 兄弟(siblings) 祖先(ancestors) 从根到节点的路的所有顶点

后代(descendants) 以节点的孩子作为树的根点.树内所有的节点

树叶(leaf) 内点(internal vertices) 高度(height) (从0开始计

1.3 m叉树(m-ary tree) 满m叉树(full m-ary tree) 每个内点都有m个孩子

完全m叉树 (complete)

1.4 有序根树 (ordered rooted tree)

左子树 右子树

2.性质

2.1 n个顶点的树 含有n-1条边

2.2 有i个内点的满m叉树 有 $n=mi+1$ 个顶点

2.3 一个满m叉树

(1) 有n个顶点，则 ①有 $\frac{(n-1)}{m}$ 个内点
②有 $\frac{[(m-1)n+1]}{m}$ 个树叶

2.4 高度为h的m叉树 最多有 m^h 个树叶

3.平衡m叉树 (balanced)

高度从0起算，一棵高度为h的树的树叶若都在h或h-1层，则是平衡的

4.应用

①二叉搜索树 ②决策树(decision tree) ③前缀码(prefix) — 二叉树
— 哈夫曼编码 (huffman code)

哈夫曼编码

前缀码：对每个字母编码，没有一个字母的编码是另个编码的开头

哈夫曼编码：用字母的出现频率对其进行二叉树的前缀码编码

→ 对一系列字母及其频率 $f_1, f_2 \cdots f_n$，每次取值最小的两个合并为树，树的根节点为频率之和，左子树边权值为0，右子树为1，全部合并后得到前缀码

习惯上将频率大的分配到左子树

博弈树 (game tree)

用树模拟整局游戏的数据结构

树的顶点的值

层数从0开始

树叶的值为此局游戏结束时第一位选手的积分

偶数层内点的值是此内点的孩子的最大值

奇数层内点的值是 ↓ 的最小值

说明

(min max strategy)

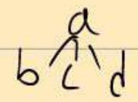
最小最大策略：第一个选手走一个节点的最大值儿子，另一个选手走最小值

但这样走，则此节点的值为第一个选手的积分

11.3 树的遍历 (tree traversal)

通用地址系统：利用完全树的顺序标记节点，但此标记必须加上其父节点的标记作为前缀

遍历



前序：先访问根节点，再从左向右使用前序 a-b-c-d
(preorder)

中序：先对本节点使用中序，左子后再访问根节点、 b-a-c-d
(inorder)

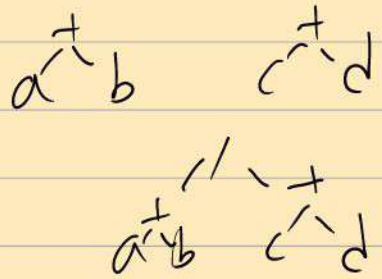
后序 b-c-d-a
(postorder)

有序根树(ordered tree)

中缀 前缀 后缀记法
infix prefix postfix

构建表达树 → 以(a+b)/(c+d)为例，拆为

操作数为树叶
符号为根



对表达树使用不同的遍历，就得到对应的记法

记法计算

前缀：从后往前扫描，若出现 符号-数-数，则计算并替换为结果(数)

后缀：从前往后 数-数-符号

11.4 生成树 (spanning tree) 生成树是图的一种，我们称为准无根(?)

对一个简单图(连通的)，删除一些边后得到的树(兄弟节点不连接，无方向要求)

删除了圈 回路

(DFS) 有时节点的选取不能随意

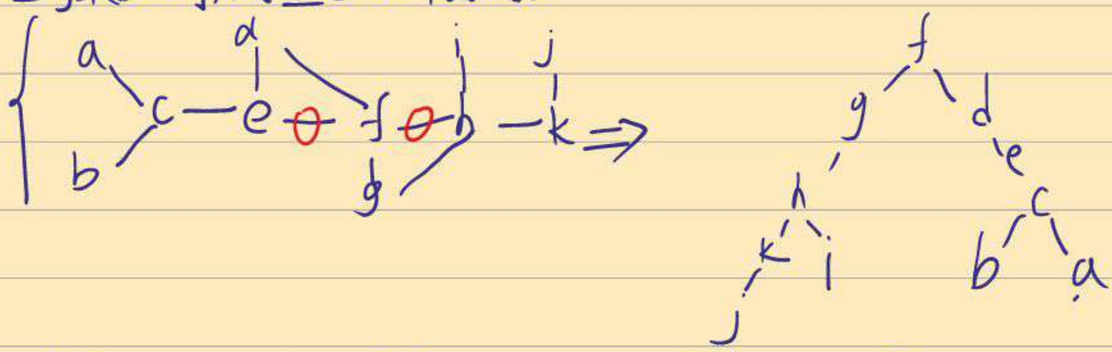
深度优先搜索(用于无权图也叫回溯)

描述：随意选取一个顶点，从此顶点出发，任意选择一个相连的且没有被选中的顶点与对应边加入生成树，继续此步骤直到无相连的顶点，随后按前进的路径回退，再次探索与回退

无向

无根

直到最后回到原点且无未选中过的点



边 {e,f}, {f,h} 称为背边，因为加入它们就会生成圈

宽度优先搜索 (BFS)

我们从图中随机选出一个点作为树的根，再选出与其相连的节点作为第一层，再选出与第一层相连的未加入的节点作第二层，最后构成完整的树

节点应按照邻接表的顺序选取

有向

深度优先搜索

与前一个相似，但 ① 有方向
② 若分支后还有节点，则再创建一个生成树，从而得到森林

11.5 最小生成树 (minimum spanning trees MST)

使边权最小的生成树

11.5.1 prim

选取1个最小边，把这个边和两个顶点一起加入最小生成树，此后，每次都选取权值最小的与目前的最小生成树相连的，不构成回路的边连同端点加入最小生成树

11.5.2 kruskal

选取权值最小的边和顶点，只要不构成回路，就加入MST

最小生成树.

↑

生成树

↑

树 → 属性 → leaf, in-v, v的公式

遍历

二叉树

满 完全

1.2 布尔代数

1. 基础

1.1
- $\bar{0}=1$ $\bar{1}=0$ $0+1=1$ $0+0=0$ $1+1=1$
- $0\cdot1=0$ $0\cdot0=0$ $1\cdot1=1$
- 非 → 补 (complement)
- 布尔和 (boolean sum) 布尔积 (boolean product)

1.2 转换 逻辑等价式 (logical equivalence)

$0\cdot1+0 \Rightarrow (0\wedge1)\vee0$ $(x+y)(x+z) \Rightarrow (p\vee q)\wedge(p\vee r)$

1.3 布尔变元 (boolean variable)

布尔函数，布尔表达式

1.4 n元布尔函数有 2^{2^n} 个，若两个布尔函数输入与输出相等，则称为是等价的 (equivalent)

↳ 有n种输出 → 对应 2 个输出 $=2^n$

1.5 补函数 $F\rightarrow\bar{F}$，其结果取反

1.6 布尔和 $F+G(x_1\cdots x_n)=F(x_1\cdots x_n)+G(x_1\cdots x_n)$

布尔积 $(FG)(x_1\cdots x_n)=F(x_1\cdots x_n)G(x_1\cdots x_n)$

1.7 对偶性 (duality)

对换交换 $\cdot$ 和 $+$，0和1 就产生对偶

$\bar{x}\cdot1+(\bar{y}+z) \rightarrow (\bar{x}+1)(\bar{y}z)$

恒等式两端取对偶，等式仍然成立

$x(x+\cdots)=x$ (恒等式)

2、求表达式与函数展开

2.1 积之和 (sum of products) (析取范式) disjunctive normal

① 从函数真值表构建表达式 (与或式)

② 将函数扩展为展开式

2.2 和之积，取结果为0的项，求和再相乘 (合取范式) conjunctive normal

2.2 函数完备性 (functional completeness)

若所有的函数都可以由一个符号集表示，则符号集是完备的

$\{\cdot, +, \bar{\ }\}$ $\{+, \bar{\ }\}$ $\{\cdot, \bar{\ }\}$ $\{|\}$ $\{\downarrow\}$ 都是完备的

$1|1=1$ $0|1=1|0=0|0=0$ $1\downarrow 1=1\downarrow 0=0\downarrow 1=0$ $0\downarrow 0=0$

3. 电路

4 电路变换

4.1 表达式化简

4.2 卡诺图