《离散数学》复习

Chapter 7: Relations

一、关系的定义

1. 集合 A、B 上的二元关系：A×B 的子集

   A ‥ A×A

2. R_1 表示 R 上定义的二元关系，对于 aRb，后者除以前者，商为整数
3. 具有 n 个元素的集合上共有 2^{n^2} 个二元关系
4. 关系的定义域与值域

   定义域 $Dom(R)=\{x\in A \mid \exists y\in B, s.t. (x,y)\in R\}$

   值域 $Ran(R)=\{y\in B \mid \exists x\in A, s.t. (x,y)\in R\}$

5. n 元关系：$A_1\times A_2\times\cdots\times A_n$ 的子集，称 n 为关系的度
6. 复合关系：设 $R:A\to B$，$S:B\to C$，则 $S\circ R: A\to C$
7. 反关系 (Inverse)：$R^{-1}=\{(x,y) \mid (y,x)\in R\}$

二、关系的性质

1. 自反性 Reflexive

   ① R 是 Reflexive $\Leftrightarrow \forall x\in A, (x,x)\in R$

   Irreflexive $\Leftrightarrow \forall x\in A, (x,x)\notin R$

   ② n 个元素的集合共有 2^{n^2-n} 个自反的关系

   ③ R 是自反的 $\Leftrightarrow R_= \subseteq R$

2. 对称性 Symmetric

   ① R 是 Symmetric $\Leftrightarrow \forall x,y\in A, (x,y)\in R \Rightarrow (y,x)\in R \Leftrightarrow R^{-1}=R$

   · R 是 Antisymmetric $\Leftrightarrow \forall x,y\in A, (x,y)\in R$ 且 $(y,x)\in R \Rightarrow x=y \Leftrightarrow R\cap R^{-1}=R_=$

   ② $R_=$ 既是对称的，又是反对称的

   ③ n 个元素的集合共有 $2^n\cdot 2^{\frac{n^2-n}{2}}$ 个对称的关系、$2^n\cdot 3^{\frac{n^2-n}{2}}$ 个反对称的关系

3. 传递性 Transitive

① R_1 具有传递性

② R 具有传递性 $\Leftrightarrow R^n \subseteq R$，$n=2,3,\cdots$

三、关系的表示

1. 矩阵表示

① 自反关系：主对角线上元素全为1

② 对称关系：$m_{ij}=m_{ji}$

③ 传递性：若 M_R^2 中的 $C_{ij}\neq 0$，则 M_R 中的 $C_{ij}\neq 0$（利用 $R^2\subseteq R$）

④ 复合关系：$M_{S\circ R}=M_R\odot M_S$

2. 有向图表示

① 自反性：每个节点上都存在环(Loop) 指形如(a,a)的边

② 对称性：每条边都与一条与之方向相反的边成对出现

四、关系的闭包

1. 定义

① 关系的闭包是满足条件的最小集合

② 自反闭包：$r(R)$，对称闭包：$s(R)$，传递闭包：$t(R)$

2. 传递闭包的计算

① 存在从a到b，长度为n的路径 $\Rightarrow (a,b)\in R^n$

② 联通关系(Connectivity Relation)

$$R^*=\{(a,b)\mid \text{存在从a到b的路径}\}=\bigcup_{n=1}^{\infty}R^n$$

$t(r)=R^*$，此方法复杂度 $O(n^4)$

③ Warshall 算法计算传递闭包，此方法复杂度 $O(n^3)$

$W_0=M_R$，$W_n=M_{R^*}$，W_i 表示只以 $v_1,v_2,\cdots,v_i$ 作为中间节点时两节点间存在的路径

$a \equiv b \pmod n$ 表示 $a\%n = b\%n$ 年 月 日

可转化为 $n \mid (a-b)$

五、等价关系和等价类

1. 等价关系：满足自反性、对称性、传递性
2. 令R是A上的等价关系，对A上的任意元素a，其等价类是集合：

$$[a]_R = \{x \in A \mid (x,a) \in R\}$$ ，a为代表元

3. 集合A在关系R上的商集为R的所有等价类组成的集合

$$A/R = \{[a]_R \mid a \in A\}$$

4. 划分 Partitions：不相交、全覆盖地将集合划分为一组子集

集合在某个关系上的等价类构成它的一组划分

六、偏序 Partial Ordering

1. 定义：关系 $R_{\le}: S \leftrightarrow S$ 称为偏序关系，若它满足自反性、反对称性、传递性

由于在S集合上定义了偏序关系，所以称S为偏序集（Poset），用 $(S, \le)$ 表示

2. 可比性：在偏序集中，若元素a、b满足 $a \le b$ 或 $b \le a$ 其中之一，则称a、b可比（Comparable）
3. 全序集：若 $(S, \le)$ 中任意两元素都可比，称S为 Totally Ordered Set 或 Linearly Ordered Set 或 Chain，称 $\le$ 为 Totally Order 或 Linear Order
4. 字典序（Lexicographic Order）：一种从多个偏序集构造新偏序集的方法

给定两个偏序集 $(A, <_A)$、$(B, <_B)$，则 $A \times B$ 上的字典序定义为

$$(a_1, b_1) < (a_2, b_2) \iff \begin{cases} a_1 <_A a_2 & \text{或} \\ a_1 = a_2 \text{ 且 } b_1 <_B b_2 \end{cases}$$

5. Hasse图：表示有限集上的偏序关系（见电子笔记）
6. 良序集 Well-Ordered Set：$(S, \le)$ 是良序集，若 ① $\le$ 是全序 ② S的每个非空子集都存在最小元
7. 格 Lattices：一个偏序集称为 Lattice，若其中每一对元素组成的集合都存在

8. 极值、最值、界、确界

令 $(A,\leq)$ 是偏序集，$B\subseteq A$

① Maximum Element 极大元　　极小：Minimum

a 是 B 的极大元：$a\in B$，在可比范畴内，B 中没有比 a 更大的元素（局部概念，可能存在多个）

② Greatest Element 最大元　　最小：Least

a 是 B 的最大元：$a\in B$，a 与 B 内其它元素都可比，且最大

③ Upper Bound 上界　　下界：Lower

a 是 B 的上界：$a\in A$，且 $\forall x\in B$，有 $x\prec a$

④ Least Upper Bound 上确界　　下确界：Greatest Lower Bound

a 是 B 的上确界：$a\in A$，且 $\forall x\in B$，有 $x\prec a$，且对 B 的其它上界 y 都有 $x\prec y$

Chapter 8: Graphs

一、图的类型和术语

1. 图的类型

① Simple Graph：无向图，不含环，两节点间最多1条边

② Multi-Graph：〃　〃　〃　可有多条边

③ Pseudo-Graph：〃　可以含环　〃　可有多条边

④ Directed-Graph：有向图，可以含环　〃　最多1条边

⑤ Directed Multi-Graph：〃　〃　〃　可有多条边

2. 节点的度 Degree

① 对于无向图：

Ⅰ. 度表示与该节点相连的边数量，记作 $deg(v)$，一个环贡献2个度

Ⅱ. 度为0的节点称为 Isolated Vertex，度为1的节点称为 Pendant Vertex

Ⅲ. The Handshaking Theorem：所有节点的度之和 = 边数×2

② 对于有向图：

Ⅰ. In-Degree：以它为到达节点的边数，$deg^-(v)$

Out-Degree：〃 出发 〃 ，$deg^+(v)$

Ⅱ. $\sum_{i=1}^{n} deg^-(v_i) = \sum_{i=1}^{n} deg^+(v_i) = |E|$

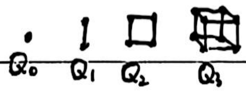
3. 一些特殊的简单图

① 完全图 Complete Graph K_n：n个节点，两两之间均存在边

② 环图 Cycles C_n：n个节点首尾相连组成的图

③ 车轮图 Wheels W_n：由 C_n 添加一个新顶点组成，这个新顶点和其它每个顶点间都存在一条边

④ 体图 Cube Q_n：有 2^n 个节点，每个节点对应一个 n bit string，若两节点 bit string 仅一位不同，则这两个节点之间存在边



⑤ 二分图 Bipartite Graph：顶点被分为两个集合，每个集合内部的任意顶点间不存在边

⑥ 完全二分图 Complete Bipartite Graph：$K_{m,n}$：顶点集 V_1、V_2 的大小分别为 m、n，且 V_1 的每个节点都连到 V_2 的每个节点，反之亦然

4. 子图

设 $G=(V,E)$，$H=(W,F)$

① 若 $W \subseteq V$，$F \subseteq E$，则 H 是 G 的子图 (subgraph)

② 若 $W = V$，$F \subseteq E$，〃 生成子图 (spanning subgraph)

二、图的表示和同构

1. Adjacency Matrix：简单图的邻接矩阵是对称的，且主对角线元素全0

2. Incidence Matrix：

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array}\begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

只能表示无向图
(e_1：环)

3. Isomorphism 同构：① 两个简单图 G_1, G_2 的顶点间存在一一对应关系，使得所有边得到保持

② 可通过比较顶点数、边数、各顶点度数、是否存在特定长度的简单回路，确定两图并非同构

三、图的连通性 Connectivity

1. 定义：① path：路径 ② circuit：回路

③ simple circuit：简单回路——每个边最多经过一次

2. 无向图连通性

① 每对节点间都存在路径，即为联通

② 若一个非连通图是多个连通子图的并，则每个子图称为 connected components.

③ 若移除一个顶点和与之相连的所有边会造成 connected components 数目的增加，则该节点为 cut vertex.

类似地定义 cut edges (bridge)

3. 有向图连通性

① strongly connected：每对节点之间，两个方向上均存在路径

② weakly connected：与之对应的无向图是连通的

4. 节点 v_i, v_j 之间存在长为r的路径的数量 $= M_A{}^r$ 中 a_{ij} 的值

遍历每个节点恰好一次

四、欧拉和哈密顿路径/回路

遍历每条边恰好一次

1. 定理：一个连通的多图存在欧拉回路 ⇔ 每个节点度数均为偶数
2. 定理：一个连通的多图存在欧拉路径但不存在欧拉回路 ⇔ 恰有2个节点度数为奇数
3. 定理：一个节点数为 n ($n \geq 3$) 的连通的简单图，若每个节点的度 $\geq \frac{n}{2}$，则存在哈密顿回路

五、最短路径问题

给定图中两个节点，寻找其间加权长度最短的路径

Dijkstra's Algorithm：简单无向连通图，权值为正.

六、平面图 Planar Graph

1. 若图能在一个平面内画出，不存在相交的边，则为平面图.

   e.g. K_4 是平面图. Q_3, $K_{3,3}$ 不是

2. Euler's Formula

   令 G 为连通的平面简单图，边数 e. 节点数 v. 令 r 为 region 数.

   则 $v-e+r=2$

3. 若 G 为连通的平面简单图 ($v \geq 3$)，则 $e \leq 3v-6$

   若 G 为 〃 且没有长为3的回路，则 $e \leq 2v-4$

4. Kuratowski's Theorem

   ① 图 G 不是平面图 ⇔ G 包含与 K_5 或 $K_{3,3}$ Homeomorphic 的子图

   ② 若两个图能由同一个图经若干次 Elementary Subdivision 得到，则它们是 Homeomorphic 的

   ③ Elementary Subdivision：删掉边 (u,v)，添加一个节点 w 和边 (u,w), (w,v)

七、图的上色问题

1. 将真实的平面地图转化为dual map
2. chromatic number：为图上色，使任意两相邻节点分配到不同颜色所需的最少颜色种类数
3. 平面图的 chromatic number ≤ 4

Chapter 9：Trees

一、基本知识

1. 树：没有简单回路的无向连通图

   （由于没有简单回路，树不能含有环或 Multiple edges）

   森林：没有简单回路的无向图（每个 Connected Components 都是一棵树）
2. 一个无向图是树 $\Leftrightarrow$ 任意两节点间存在唯一的简单路径
3. 树和一个指定的根（root）组成的有向图称为 rooted tree
4. 在有根树中：

   ① 节点 v 的 level $l(v)$ 是从根到 v 的简单路径长

   ② 树的高 $h = \max(l(v))$

   ③ m-ary Tree：每个内部节点的儿子数不超过 m

   ④ full m-ary Tree：　　〃　　恰为 m

   ⑤ ordered rooted tree：每个内部节点的儿子是有序的（左儿子，右儿子…）

   ⑥ balanced tree：设树高为 h，所有叶子分布在高为 h 或 (h−1) 处

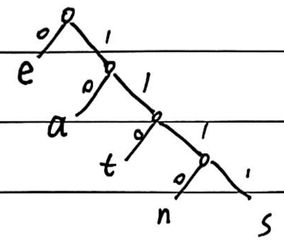
二、树的性质

1. n个节点的树有(n-1)条边
2. 有i个内部节点的 full m-ary Tree，一共有$n=mi+1$个节点
3. 高度为h的 m-ary Tree 最多有m^h个叶子
4. 高度为h的 m-ary Tree 若有l个叶子，则$h \geq \lceil \log_m l \rceil$

   若它还是 full and balanced，则$h=\lceil \log_m l \rceil$

三、树的应用

Prefix codes & Huffman Coding



e: 0
a: 10
t: 110
...

四、树的遍历 Tree Traversal

Preorder：中左右　Inorder：左中右　Postorder：左右中

五、Spanning Tree 生成树

1. 定义：令G是一个简单图，树T是G的生成树，若：

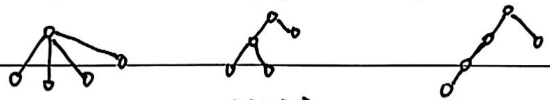
   ① T是G的子图　② T包含G的所有节点

2. 定理：简单图是连通的 ⟺ 它有生成树

   e.g. 找出K5全部非同构的生成树

   K5有5个节点，因此其生成树有4条边，8个度。

   把这8个度分配到5个节点，为保证连通性，每个节点的度必须≥1



11114　11123　11222

3. 生成树的构建： Depth-First / Breadth-First Search
(back tracking)

4. 最小生成树的构建：

① Kruskal's Algorithm：每次添加权重最小、且不会形成回路的边，重复 n-1 次

② Prim's Algorithm：每次添加与已有节点邻接且权重最小、不会形成回路的边，重复 n-1 次