By CC98@想拿满绩

浙江大学
ZHE JIANG UNIVERSITY

Relations

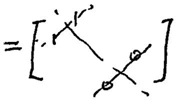
2^{n^2-n} reflexive: $\forall a\in A, (a,a)\in R$ $\quad \forall x(x\in A \rightarrow (x,x)\in R)$

M_R 对角线必全为1
digraph: 每个ohe之有个loop

irreflexive: $\forall x\,(x\in A \rightarrow (x,x)\notin R)$. ~~digraph~~ ~~a loop~~

M_R - - ——— - - 0

$\frac{n}{2}\times 2^{\frac{n^2-n}{2}}$ symmetric: $\forall x \forall y\,((x,y)\in R \rightarrow (y,x)\in R)$

$M_R =$ 

digraph ~~arc~~ 有arc(x,y 则有arc(y,

$\frac{n}{2}\times 3^{\frac{n^2-n}{2}}$ antisymmetric: $\forall x \forall y\,((x,y)\in R \wedge (y,x)\in R \rightarrow x=y)$

~~transtive~~
transitive: $\forall x \forall y \forall z\,((x,y)\in R \wedge (y,z)\in R \rightarrow (x,z)\in R)$

$\Updownarrow$ $\quad \rightarrow \overline{r_{ij}\wedge r_{jk}} \vee r_{ik} = 1$

$R^n \subseteq R$ for $n=1,2,3,\cdots$

~~$(a,b)\in R$ $(b,c)\in S$~~ ~~R和S复合 $S\circ R$~~ ~~$(a,c)\in S\circ R$~~

关系的表示: $A=\{a_1,\cdots,a_m\}$ $B=\{b_1,\cdots,b_n\}$

$$m_{ij} = \begin{cases} 1 & (a_i,b_j)\in R \\ 0 & (a_i,b_j)\notin R \end{cases} \rightarrow \text{connection matrix}$$

$(a_i,b_j)\in R$ $\quad a_i \rightarrow b_j$ $\quad$ digraph / directed graph

---

symmetric, transitive $\not\Rightarrow$ reflexive

$M_{R_1\cup R_2} = M_{R_1} \vee M_{R_2}$ $\quad M_{R_1} = (c_{ij})$

$M_{R_1\cap R_2} = M_{R_1} \wedge M_{R_2}$ $\quad M_{R_2} = (d_{ij})$

$M_{\overline{R_1}} = (\overline{c_{ij}})$

$M_{R_1-R_2} = M_{R_1\cap \overline{R_2}} = (c_{ij} \wedge \overline{d_{ij}})$

composition

$R = \{(a,b) \mid a\in A, b\in B, aRb\}$, $S=\{(b,c) \mid b\in B, c\in C, bSc\}$

R和S复合: $S\circ R = \{(a,c) \mid a\in A \wedge c\in C \wedge \exists b\,(b\in B \wedge aRb \wedge bSc\}$

$M_{S\circ R} = M_R \cdot M_S$

inverse relation.

$R = \{(a,b) \mid a \in A, b \in B, aRb\}$

$R^{-1} = \{(b,a) \mid (a,b) \in R, a \in A, b \in B\}$

也可表示为 R^c

properties: R, S. $A \to B$ $T: B \to C$

$(R \cup S)^{-1} = R^{-1} \cup S^{-1}$

$(R \cap S)^{-1} = R^{-1} \cap S^{-1}$

$(\overline{R})^{-1} = \overline{R^{-1}}$

$(R - S)^{-1} = R^{-1} - S^{-1}$

$(A \times B)^{-1} = B \times A$

$\overline{R} = A \times B - R$.

$(T \circ S)^{-1} = S^{-1} \circ T^{-1}$

Closures of Relations

reflexive closure: $r(R) = R \cup I_A$

symmetric closure: $s(R) = R \cup R^{-1}$ $M_{s(R)} = M_R \vee M_R^T$

transitive closure: $t(R) = R^*$

术语: a path of length n in a digraph G: $(x_0, x_1), (x_1, x_2), \cdots, (x_{n-1}, x_n)$

notation: $x_0, x_1, \cdots, x_n$.

cycle or circuit: 若 $x_0 = x_n$

a, b 间有长为 n 的 path $\Leftrightarrow$ $(a,b) \in R^n$

the connectivity relation $R^* = \bigcup_{n=1}^{\infty} R^n$.

$t(R) = R^*$ 中关于最小性证明:

假设 S transitive, $S \supseteq R$. 我们下证 $S \supseteq R^*$.

$\because S^n \subseteq S \quad \therefore S^* \subseteq S \qquad \because R \subseteq S \quad \therefore R^* \subseteq S^* \subseteq S$ □.

事实上，只需考虑长不超过 n 的 path

if $|A| > n$, then any path of length $> n$ must contain a cycle.

pf: Pigeon Hole Principle.

计算 $t(R)$ Warshall's Algorithm.

~~定义~~ 若 $a, x_1, x_2, \cdots, x_{m-1}, b$ 为一条 path, its interior vertices are $x_1, x_2, \cdots, x_{m-1}$

Warshall's Algorithm 基于一系列 $W_0, W_1, \cdots, W_n$. ($x_1 \sim x_{m-1}$ 中可含 a, b)

$W_0 = M_R \qquad W_k = (w_{ij}^{(k)})$

$$w_{ij}^{(k)} = \begin{cases} 1 & \text{if there is a path from } v_i \text{ to } v_j \text{ s.t. all the interior vertices of this path are in the set } \{v_1, v_2, \cdots, v_k\} \\ 0 & \text{otherwise} \end{cases}$$

$W_n = M_{t(R)}$

$$w_{ij}^{(k)} = w_{ij}^{(k-1)} \vee \left(w_{ik}^{(k-1)} \wedge w_{kj}^{(k-1)} \right)$$

$\downarrow$ 所有内点在 $\{v_1 \cdots v_{k-1}\}$ 中

$\downarrow$ v_k 在内点集中.

Algorithm: $O(n^3)$.

```
W = (w_ij)nxn ;
for (k=1; k<=n; k++)
  for (i=1; i<=n; i++)
    for (j=1; j<=n; j++)
      w_ij = w_ij ∨ (w_ik ∧ w_kj)
```

R is an equivalence relation if R is

- reflexive
- symmetric
- transitive

equivalence class $[x]_R$

集合的一个 partition 与一个等价关系 R 对应.

若 R_1, R_2 均为等价关系(A上的)

则 $R_1 \cap R_2$ 也为等价关系(A上的)
(reflexive 的交 仍 reflexive
symmetric 的交 仍 symmetric
transitive 的交 仍 transitive)

$R_1 \cup R_2$ reflexive, symmetric

---

partial orderings

R is a partial ordering or partial order if R is reflexive, antisymmetric, transitive.

A set S together with a partial ordering R is called a partially ordered set, ~~or~~ poset, and is denoted by (S, R)

---

comparable / incomparable

偏序集 $(S, \preceq)$ 中的元素 a, b 称为 comparable ~~either $a \preceq b$ or $b \preceq$~~ $a \preceq b$ or $b \preceq a$

否则 称二者 incomparable

If $(S, \preceq)$ is a poset and S 中任意两元素可 comparable,

S: totally ordered or linearly ordered

此时 $(S, \preceq)$ 被称为 a chain

Lexicographic order

Given $(A_1, \preceq_1)$, $(A_2, \preceq_2)$, we construct an induced partial order R on $A_1 \times A_2$: $(x_1, y_1) \prec (x_2, y_2)$ either if $x_1 \prec_1 x_2$ or $x_1 = x_2$ and $y_1 \prec_2 y_2$

Hasse Diagrams —— A method used to represent a partial ordering

---

$(A, \preceq)$ is a poset $B \subseteq A$ 若 $(B, \preceq)$ 是全序集, B 称为 a chain of $(A, \preceq)$

$B \subseteq A$ 若 对 $\forall a, b \in B$ $(a \neq b)$, $(a, b) \notin R$, $(b, a) \notin R$; B 称为 a antichain of $(A, \preceq)$

Maximal and Minimal Elements.

极大、极小元 $\quad a$ 极大 $\Leftrightarrow \nexists b.\ a \prec b$

Greatest and Least Element

最大、最小元 $\quad a$ 最大 $\quad \forall b \quad b \preceq a$

Upper and Lower Bounds

$(S, \preceq) \quad A \subseteq S.$ 若 $\exists a \in S$. s.t. $\forall b \in A \quad b \preceq a$ ~~称A为~~

称 a 为 A 的一个 upper bound

Least Upper and Greatest Lower Bound.

lub(A) $\qquad$ glb(A)

Well-ordered Sets : every nonempty subset of A has a least element

注：良序集是全序集.

Lattice

A poset is called a lattice if every pair of elements has a lub and a glb.

$\left.\begin{array}{l} \text{全序集} \\ (\mathbb{Z}^+, |) \\ (P(S), \subseteq) \end{array}\right\}$ 均为 Lattice.

Topological Sorting

~~We ~~sort~~ impose a total ordering $\preceq$~~

Def A total ordering $\preceq$ is said to be compatible with the partial ordering R if $a \preceq b$ whenever aRb.

Constructing a compatible total ordering from a partial ordering is called topological sorting.

Lemma 每个有限非空~~集合~~偏序集 $(S, \preceq)$ 有极小元.

Pf: ~~设 a_0~~ 任取 a_0, 若 a_0 不是, $\exists\, a_1 \preceq a_0$

若 a_1 不是 $\exists\, a_2 \preceq a_1$, - - - .

Algorithm: To sort a poset (S, R).

- Select a (any) minimal element and put it in the list. Delete it from S.
- Continue until all elements appear in the list (and S is void).

~~simple graph~~

Graphs

		Multiple edges allowed?	Loops allowed?
Simple graph	Undirected	×	×
Multigraph	Undirected	✓	×
Pseudograph	Undirected	✓	✓
Simple directed graph	Directed	×	×
Directed multigraph	Directed	✓	×
Mixed graph	Mixed	✓	✓

Basic Terminology

- Vertex, Edge
- adjacent vertices (or neighbors)
- An edge e connecting u and v is called incident with vertices u and v, or is said to connect u and v.
- u, v are called endpoints of edge $\{u, v\}$.
- loop
- ~~the~~ degree of a vertex (a loop at a vertex contributes twice)

  $\deg(v) = 0 \Rightarrow v$: isolated

  $\deg(v) = 1 \Rightarrow v$: pendant

$$\sum_{v \in V} \deg(v) = 2e$$

Some ~~special~~ special simple graphs.

1) Complete Graphs $- K_n$
2) Cycle $- C_n$
3) Wheels $- W_n$ (C_n 填加一个顶点 构成 W_n)
4) n-Cubes $- Q_n$

Bipartite Graphs

complete bipartite graph $K_{m,n}$

<u>Thm</u> A simple graph is bipartite

$\Longleftrightarrow$ ~~It is possible to assign~~
~~可对其~~ 可用2种颜色对图染色，使相邻结点不同色.

Regular ~~gr~~ Graph

simple graph + 每个结点度相同

n-regular : 每个结点度为 n.

~~边~~ K_n : $(n-1)$-regular

$G=(V,E)$ $H=(W,F)$

H is a subgraph of G if $W \subseteq V$. $F \subseteq E$.

proper subgraph if $H \neq G$.

spanning subgraph if $W=V$, $F \subseteq E$

$G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$

Representing Graphs

- Graphs
- Adjacency lists — lists that specify all the vertices that are adjacent to each vertex.
- Adjacency matrix $a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G \\ 0 & \text{otherwise} \end{cases}$ (基于顶点排序方式)
  (multigraph or pseudograph 则 是非0-1矩阵)
  (有向图 $a_{ij}=1$ 当 $v_i \to v_j$ 有边)
- ~~Incidence matrix~~

~~sum~~ the sum of the entries in a row of the adjacency matrix for an undirected graph?

The number of edges incident to the vertex i, which is the same as degree of i minus the number of loops at i. 对于directed? $\deg^+(v_i)$

ZHE JIANG UNIVERSITY

- Incidence matrix

  $G=(V,E)$ $\quad v_1,\cdots,v_n$

  $\qquad\qquad e_1,\cdots,e_m$

  the incidence matrix with respect to this ordering of V and E is $n\times m$ matrix $M=(m_{ij})_{n\times m}$, where

  $$m_{ij}=\begin{cases}1 & \text{when edge } e_j \text{ is incident with } v_i\\ 0 & \text{otherwise}\end{cases}$$

  Note: Incidence matrices of undirected graphs:

  ~~每~~列 2个1 : 对于连接不同顶点的边

  ~~每~~列 1个1 : 对于 loops.

Isomorphism of Graphs

$G_1=(V_1,E_1)$ $\quad G_2=(V_2,E_2)$

1-1 and onto $f: V_1\to V_2$

$\forall a.b\in V_1$ $\quad$ a.b 在 G_1 中相邻 $\Leftrightarrow$ $f(a)\cdot f(b)$ 在 G_2 中相邻.

判断是否同构: 使用不变量.

使用 adjacency matrix

---

paths

a path of length n in a simple graph 是一列顶点 $v_0, v_1, \cdots, v_n$ 使相邻边存在

circuit : a path, $v_0=v_n$, $n>0$ $\qquad$ notation : vertex sequence

称 path is simple 若其未将某条边包含多于1次、

a path of length zero consists of a single vertex

有向图中的 path 类似定义.

Number of different paths of length r from v_i to v_j is equal to $(A^r)_{ij}$ (A为图的 adjacency matrix)

(注: 这里是标准的矩阵幂, 不是 Boolean product.

connected: 每对不同顶点之间有个 path.

a connected undirected path. 每对不同顶点之间必有 simple path.

the maximally connected subgraphs of G are called the connected components or just the components

---

cut vertex (or articulation point): if removing it and all edges incident with it results in more connected components than in the original graph.

cut edge (or bridge) 类似定义.

Connectedness in directed graphs.

A directed graph is <u>strongly connected</u> if there is a path from a to b and from b to a ($\forall a, b$).

___ ___ ___ <u>weakly connected</u> if the underlying undirected graph is connected.

Strongly connected components (or strong components)

(path, connected components, circuit of length n ... 也可作为不变量判断图的同构)

Euler ~~path~~ Path: A simple path containing every edge of G.

Euler Circuit: - - - - circuit - - - -

Euler Graph: A graph contains an Euler circuit.

Thm A connected multigraph has an Euler circuit

⟺ 每个~~结点度~~结点度为偶数.

Thm __ __ __ ________ ____ path but not an Euler circuit

⟺ 恰有两个结点度为奇数.

Thm A directed multigraph having no isolated vertices has an Euler circuit

⟺ - weakly connected
  - in-degree and out-degree of each vertex are equal

has an Euler path but not an Euler circuit

⟺ - __ ________ __
  - 每点入度出度相同. 除去1个 入度=出度+1
    1个 出度=入度+1.

~~Thm~~ Hamilton Path : visits every vertex exactly once.
Hamilton circuit
Hamilton graph

Dirac's Thm

~~~~ G n个顶点 简单图 n ≥ 3. 每个结点度 $\geq \frac{n}{2}$.

$\Rightarrow$ G ~~has a~~ 有 Hamilton circuit.

Ore's Thm

G n个顶点 简单图. $\forall u, v$ 不相邻. $deg(u) + deg(v) \geq n$

$\Rightarrow$ G 有 Hamilton circuit

An Important necessary condition: (对于 H circuit)

For any nonempty subset S of set V, the number of connected components in G−S <= |S|

G−S 是 G 的子图.

设 C 是 G 的一条 H circuit. 对任一 V 的非空子集 S.

C−S 中的 连通分支 <= |S|

G−S 中的 连通分支 <= C−S 中的连通分支.

Weighted graph $G = (V, E, W)$

Dijkstra's Algorithm (undirected graph with positive weights)

Step 1 $L_0(a) = 0$. $L_0(v) = \infty$ ($\forall v \neq a$) $S_0 = \phi$

Step 2. $S_k = S_{k-1} \cup \{$ ~~a vel~~ a vertex u with the smallest label $\}$

Then update the label $L_k(v) = \min\{L_{k-1}(v), L_{k-1}(u) + w(u,v)\}$

$O(n^2)$.
~~~~

~~Def~~ A graph is called planar if it can be drawn in the plane without any edges crossing.

Such a drawing is called a planar representation of the graph.

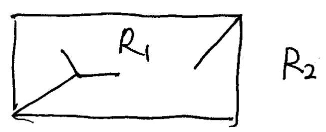
~~Let~~ G a connected planar simple graph.

$$r = e - v + 2$$

regions edges vertices

R: a region of a connected planar simple graph

deg(R): the number of the edges on the boundary of R c



$\deg(R_1) = 12$

$\deg(R_2) = 4$

Cor1 G连通简单平面图. $v \geq 3$, 则 $e \leq 3v - 6$.

Pf: $2e = \sum_{\text{region}} \deg(r_i) \geq 3r$ (每个区域度 ≥ 3).

$\Rightarrow r \leq \frac{2}{3}e \Rightarrow e - v + 2 \leq \frac{2}{3}e \Rightarrow e \leq 3v - 6$ (非连通也可得此结论)

Cor2 G连通平面简单图. $\Rightarrow$ G有个度不超过5的顶点

Pf: $v \geq 3$时. 由Cor1 $\Rightarrow e \leq 3v - 6 \Rightarrow 2e \leq 6v - 12$

若 $\deg v \geq 6$ $(\forall v)$, 则 $2e \geq 6v$. 矛盾!

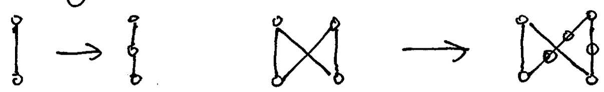
Cor3 G连通平面简单图, $v \geq 3$ 无长为3的circuit $\Rightarrow e \leq 2v - 4$

Pf: 与Cor1类似, 只不过每个区域度 $\deg(r_i) \geq 4$

Homeomorphic

- G_1, G_2 are called homeomorphic if they can be obtained by a sequence of elementary subdivision.

注: elementary subdivision:



Thm A graph is nonplanar
$\Leftrightarrow$ It contains a subgraph homeomorphic to $K_{3,3}$ or K_5

- dual graph of the map. 将区域用点替代. 相邻 (不包含仅~~有一点~~ touch at only one point) 的区域所对应的点连线

~~coloring~~:

coloring: · 没有2个相邻结点是同色的.

chromatic number of a graph: $\chi(G)$. 所需最少颜色数

The chromatic number of a planar graph is no greater than four.

Applications of graph coloring:

1) Scheduling exams.
2) Set up natural habitats of animals in a zoo.

Tree.

~~Tree~~

Def. A tree is a connected undirected graph with no simple circuits.

Forest is an undirected graph with no simple circuits.

Note: ① Any tree must be a simple graph.

② Each connected components of forest is a tree.

Thm An undirected graph is a tree

$\Longleftrightarrow$ there is a unique simple path between any two of its vertices

---

Rooted tree

Once we specify a root, we direct each edge away from the root.
A tree together with its root produces a directed graph called a rooted tree.

Terminology

- Parents: v 不是 root, the parent of v is u 唯一的 (u, v) a directed edge
  Children: v: a child of u when u is a parent of v
- Siblings: vertices with the same parent
- Ancestor: the vertices in the path from the root to this vertex excluding the vertex itself and including the root
- Descendant: those vertices that have v as an ancestor
- Leaf: vertices which have no children
- internal vertices: vertices that have children
- subtrees: If a is a vertex in a tree, the subtree with a as its root is the subgraph of the tree consisting of a and its descendants and all edges incident to these descendants

~~A rooted tree is called m-ary tree~~

- m-ary tree : every internal vertex has no more than m children.
- binary tree : $m=2$
- full m-ary tree : every internal vertex has exactly m children.
- ordered rooted tree : a rooted tree where the children of each internal vertex are ordered

ordered binary tree — left / right child (2 possible children of a vertex, ~~are called~~ if they exist)

the tree rooted at the left / right child is called left / right subtree

Thm A tree with n vertices has $n-1$ edges

Thm A full m-ary tree with i internal vertices contains $n=mi+1$ vertices

Thm A full m-ary tree with

- n vertices has $i=\frac{n-1}{m}$ internal vertices $\left(l=n-\frac{n-1}{m}=\frac{[(m-1)n+1]}{m}\right.$ ~~lea~~ leaves

$n=mi+1$

$n=i+l$

n, i, l 知1个可计算其它

For a full binary tree, $l=i+1$, $e=v-1$

- Level : the level of vertex v in a rooted tree is the ~~height~~ length of the unique path from the root to v
- Height : the height of a rooted tree is the maximum of the levels of its vertices
- Balanced : A rooted m-ary tree of height h is called balanced if all its leaves are at levels h or $h-1$

~~Tree~~ Thm There are at most m^h leaves in an m-ary tree of height h.

Cor. 1) If an m-ary tree of height h has l leaves, then $h \geq \lceil \log_m l \rceil$

2) If the m-ary tree is full and balanced, then $h = \lceil \log_m l \rceil$

Pf: 1) $l \leq m^h$

2) $m^{h-1} < l \leq m^h \Rightarrow h-1 < \log_m l \leq h$

Every tree is a bipartite.

Applications of Trees

- Binary Search Tree
- Decesion Tree
- Prefix codes

binary search tree: vertices are assigned keys so that the key of a vertex is both larger than the keys of all vertices in its left subtree and smaller than the keys of all vertices in its right subtree.

If a binary search tree is balanced, locating or adding an item requires no more than $\lceil \log(n+1) \rceil$ comparisons.

- Decision Trees.
  A rooted tree in which each internal vertex corresponds to a decision, with a subtree at these vertices for each possible outcome of the decision.

- Prefix Codes
  The bit string for a letter never occur as the first part of the bit string for another letter

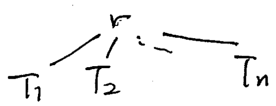
Huffman Coding:
  efficient codes based on the frequencies of occurrences of characters.

object: $\min(\sum l_i w_i)$.

F：是 n个有根树构成的森林，每个有根树仅由一个结点 a_i 构成，权 w_i.

while F not a tree
  按权排序，最小的2个组成一颗新树.（若 $w(T) \geq w(T')$，
  T' 左子树，T 右子树
  边标0，边标1）

Tree Traversal Algorithm



- preorder $r \to T_1 \to T_2 \to \cdots \to T_n$
- inorder $T_1 \to r \to T_2 \to \cdots \to T_n$
- postorder $T_1 \to T_2 \to \cdots \to T_n \to r$

Infix form: the fully parenthesized expression obtained by an inorder traversal of the binary tree.

prefix form (Polish notation)

Postfix form (reverse Polish notation).

spanning tree

G简单图. A spanning tree of G is a subgraph of G that is a ~~the~~ tree containing every vertex of G.

Thm A simple graph is connected $\Longleftrightarrow$ it has a spanning tree

Depth-first search (also called backtracking)

~~Breadth-first search~~

1. 任选一个结点为~~树~~根
2. 不断加边，向从结点引出的path中.
3. 若经过了所有点，则完成。
   否则回溯后继续上述.

Breadth-first search

1. 任选一个结点为根，~~而~~将与之相邻的结点加入构成树.
2. 新结点成为level 1，任意排序
3. 对于level 1中每一结点按顺序访问，并~~其~~添加与之相邻且未被加入的所有结点. 所有新加入的结点位于level 2
4. 重复直至完成.

Backtracking scheme — Applications

- Graph Coloring
- ~~e~~n-Queens Problem
- Sums of Subsets.

~~minima~~

A minimal spanning tree in a connected weighted graph ~~in~~ is a spanning tree that has the smallest possible sum of weights of its edges.

Prim's algorithm　　选与已构成树中的点相~~邻~~连的边权最小的加入.

Kruskal's algorithm　　选权最小的依次加入、

By CC98@想拿满绩