~~§1 Logic and proof.~~

§1 Logic and proof.

proposition (true/false)

converse: $q \to p$
contrapositive: $\neg q \to \neg p$
inverse: $\neg p \to \neg q$

Logical connectives:

Negation. Conjunction ($\wedge$) Disjunction ($\vee$) Exclusive or ($\oplus$) Implication ($\to$)

异或，相同为0，相异为1 　 ($T \to F$: F) ($p \to q$)

Biconditional ($\leftrightarrow$)

Tautology (永真) Contradiction Contingency

Logically equivalent : $p \leftrightarrow q$ is a tautology.
denoted by $p \equiv q$ or $p \Leftrightarrow q$

De Morgan's Laws.

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

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

~~Identity Laws~~

~~Associative Laws~~

~~$(p \vee q) \wedge r \equiv p$~~ Distribution laws.

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

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

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

propositional Logic

predicate Logic

Constant : "1", "Sustech"
Variable : x, y
predicate: ~~S~~ student(x)

Quantifiers:
Universal: $\forall$
Existential: $\exists$
最高优先级

$\forall x (At(x, SUSTech) \to Smart(x))$

$\forall x (At(x, SUSTech) \wedge Smart(x))$

$\neg \exists x P(x) \equiv \forall x \neg P(x)$

$\neg \forall x P(x) \equiv \exists x \neg P(x)$

$\forall x \exists y L(x,y) \not\equiv \exists y \forall x L(x,y)$

Everybody loves somebody 　 There is someone who is loved by everyone

若quantifiers相同则不影响.

Rules of Inference:

$$\frac{p \to q \quad p}{\therefore q}$$ 肯定前件式 modus ponens

$$\frac{p \to q \quad \neg q}{\therefore \neg p}$$ 否定后件式 modus tollen

$$\frac{p \to q \quad q \to r}{\therefore p \to r}$$ 假言三段论 hypothetical syllogism

$$\frac{p \vee q \quad \neg p}{\therefore q}$$ 选言三段论 disjunctive syllogism

$$\frac{p}{\therefore p \vee q}$$

$$\frac{p \wedge q}{\therefore q}$$

$$\frac{q}{\therefore p \wedge q}$$

$$\frac{\neg p \vee r \quad p \vee q}{\therefore q \vee r}$$

$\forall e$: UI 　 $\forall i$: UG 　 $\exists e$: EI 　 $\exists i$: EG

Methods to prove

- contrapositive
- Contradiction
- by cases
- equivalence.

§2 Set, Functions, Sequence, Sum and Matrices

Set: listing the elements. (unordered collection of objects)

$A \subseteq B$: $\forall x (x \in A \rightarrow x \in B)$

Cardinality : the number of elements in S
infinite set

powerset : $P(S)$. The set of all subset of set S.
$|S| = n$, then $|P(S)| = 2^n$

Tuple: $(a_1, a_2, \cdots a_n)$ ordered

Cartesian Product: $A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$
$A_1 \times A_2 \times \cdots \times A_n = \{(a_1, a_2, \cdots a_n) \mid a_i \in A_i \text{ for } i = 1, 2, \cdots, n\}$
$|A \times B| = |A| \times |B|$

Set Operation : $A \cap B$, $A \cup B$. $\bar{A}$ (A^c), $A - B$
disjoint : $A \cap B = \phi$ $\quad |A \cup B| = |A| + |B| - |A \cap B|$
$\overline{A \cap B} = \bar{A} \cup \bar{B}$ (De Morgan's laws)

Function: $f: A \rightarrow B$ exactly one element of B to each element of A
domain codomain

$f(a) = b$ $\quad$ range of f : the set of all images of elements of A
preimage image

Injective (1-1) $\quad$ Surjective (onto) $\quad$ bijective



即 $f(x) = f(y)$ implies $x = y$ $\quad$ for every $b \in B$, there is $a \in A$ s.t. $f(a) = b$

Inverse : $f^{-1}(b) = a$ when $f(a) = b$. f should be bijective

Composition: $(f \circ g)(x) = f(g(x))$

Sequence: a function from a subset of integers to a set S

Summation:

~~$a, ar, ar^2 \cdots$: $S = \frac{a(r^{n+1} - 1)}{r - 1}$~~

$\sum_{k=0}^{n} ar^k \, (r \neq 0) = \frac{ar^{n+1} - a}{r - 1}, \; r \neq 1 \quad \sum_{k=1}^{n} k = \frac{n(n+1)}{2} \quad \sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$

$\sum_{k=1}^{n} k^3 = \frac{n^2(n+1)^2}{4} \quad \sum_{k=0}^{\infty} x^k, |x| < 1 = \frac{1}{1-x} \quad \sum_{k=1}^{\infty} kx^{k-1}, |x| < 1 = \frac{1}{(1-x)^2}$

Countable: finite or have same Cardinality as Z^+ $\quad |S| = \aleph_0$
Z, Q

Uncountable:
R, $P(N)$

Schröder-Bernstein Theorem:
$|A| \leq |B|$ and $|B| \leq |A| \Rightarrow |A| = |B|$

§3 Algorithms

Big-O Notation: $f(n) = O(g(n))$, if there exist some positive constants C and x_0 s.t. $|f(n)| \le C|g(n)|$ when $n > x_0$.

e.g. $\cancel{4n^2}$, $\frac{1}{10}n^2$, $100n+10000$ ~~all~~ are all $O(n^2)$ upper bound.

$1, \log n, n, n\log n, n^2, 2^n, n!, \cdots$

Big-Omega Notation: $f(n) = \Omega(g(n))$, if there exist some positive constants C and x_0 s.t. $|f(n)| \ge C|g(n)|$ when $n > x_0$.

$f(x)$ is $\Omega(g(x)) \Longleftrightarrow g(x)$ is $O(f(x))$ lower bound.

Big-Theta Notation: $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $g(n) = O(f(n))$

Time complexity: The number of machine operations needed in an algorithm.

Space complexity: The amount of memory needed

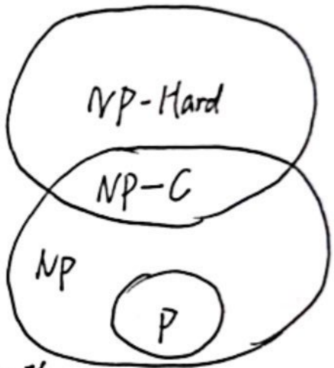
多项式时间: $O(n^k)$

非----: $O(n!), O(2^n) \cdots$

P: 多项式时间复杂度求解

NP: 多项式时间内可验证.

NP-C: 可多项式时间内验证，但不能多项式时间内求解.



P=NP?

~~解决~~

如果找到一算法多项式时间内解决NP，那么所有NP问题也能多项式时间内解决.

§4 Number Theory

Division.. $a|b$ if $b=ac$

prop: (i) if $a|b$ and $a|c$, then $a|(b+c)$

(ii) if $a|b$ then $a|bc$ for integer c

(iii) if $a|b$ and $b|c$, then $a|c$

Congruence: $a \equiv b \pmod m$ if m divides $a-b$ ~~if m divides a-b~~ (def)

if and only if $a \bmod m = b \bmod m$

- if $a \equiv b \pmod m$ and $c \equiv d \pmod m$, then $a+c \equiv b+d \pmod m$ and $ac \equiv bd \pmod m$
- $a \equiv b \pmod m$ if and only if $a = b + km$
- if $a \equiv b \pmod m$, then

  $c \cdot a \equiv c \cdot b \pmod m$

  $c + a \equiv c + b \pmod m$

$(a+b) \bmod m = ((a \bmod m) + (b \bmod m)) \bmod m$

$ab \bmod m = ((a \bmod m)(b \bmod m)) \bmod m$.

$+_m$: $a +_m b = (a+b) \bmod m$ $\quad Z_m: \{0,1,2,\cdots,m-1\}$

$\cdot_m$: $a \cdot_m b = ab \bmod m$ $\quad$ 满足交换律.结合律.分配律

$(\underline{101}\,\underline{011}\,\underline{111})_2 = (537)_8$

Primes: devidible only be 1 and itself

composite (合数): greater than 1 and not prime.

Fundamental Theorem of Arithmetic: Every integer greater than 1 can be uniquely as a prime or as the product of two or more primes.

- If n is composite, then n has a prime divisor less than or equal to $\sqrt{n}$

$a = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n}$, $b = p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n}$, $\gcd(a,b) = p_1^{\min(a_1,b_1)} p_2^{\min(a_2,b_2)} \cdots p_n^{\min(a_n,b_n)}$

$\operatorname{lcm}(a,b) = p_1^{\max(a_1,b_1)} p_2^{\max(a_2,b_2)} \cdots p_n^{\min(a_n,b_n)}$

Euclidean Algorithm: Let $a = bq + r$, then $\gcd(a,b) = \gcd(b,r)$

Bezout's Theorem: a,b positive integers, $\exists\, s,t$ s.t. $\gcd(a,b) = sa + tb$

Example:

$$
\begin{aligned}
503 &= 1\cdot 286 + 217 & \qquad 1 &= 10 - 1\cdot 9 \\
286 &= 1\cdot 217 + 69 & &= 7\cdot 10 - 1\cdot 69 \\
217 &= 3\cdot 69 + 10 & &= 7\cdot 217 - 22\cdot 69 \\
69 &= 6\cdot 10 + 9 & &= 29\cdot 217 - 22\cdot 286 \\
10 &= 1\cdot 9 + 1_{(\gcd)} & &= 29\cdot 503 - 51\cdot 286
\end{aligned}
$$

Cor. $\gcd(a,b)=1$ and $a|bc$, then $a|c$

if p is prime and $p|a_1a_2\cdots a_n$, then $p|a_i$ for some i

If $ac \equiv bc \pmod m$ and $\gcd(c,m)=1$, then $a \equiv b \pmod m$

Solving Linear Congruences.

$ax \equiv b \pmod{m}$

$\bar{a}a \equiv 1 \pmod{m}$: inverse of a modulo m

~~for~~ $ax \equiv b \pmod{m}$

$\rightarrow \bar{a}ax \equiv \bar{a}b \pmod{m}$

$\rightarrow x \equiv \bar{a}b \pmod{m}$

If $\gcd(a,m)=1$, then $sa + tm = 1 \pmod{m}$

Since $tm \equiv 0 \pmod{m}$, then $sa \equiv 1 \pmod{m}$

s is the inverse.

The Chinese Remainder Theorem.

Let $m_1, m_2, \cdots, m_n$ be pairwise relatively prime positive integers greater than 1

$x \equiv a_1 \pmod{m_1}$

$x \equiv a_2 \pmod{m_2}$

$\vdots$

$x \equiv a_n \pmod{m_n}$

has a unique solution modulo $m = m_1 m_2 \cdots m_n$

let $M_k = m/m_k$, since $\gcd(m_k, M_k) = 1$, there is an integer y_k s.t. $M_k y_k \equiv 1 \pmod{m_k}$

let $x = a_1 M_1 y_1 + a_2 M_2 y_2 + \cdots + a_n M_n y_n$, then x is a solution

Example:

$x \equiv 2 \pmod{3}$

$x \equiv 3 \pmod{5}$

$x \equiv 5 \pmod{7}$

$m = 3 \cdot 5 \cdot 7 = 105$, $M_1 = m/3 = 35$, $M_2 = m/5 = 21$, $M_3 = m/7 = 15$

$35 \cdot 2 \equiv 1 \pmod{3}$ $y_1 = 2$

$21 \equiv 1 \pmod{5}$ $y_2 = 1$

$15 \equiv 1 \pmod{7}$ $y_3 = 1$

$x = 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 \equiv 233 \equiv 23 \pmod{105}$

Application:

Pseudorandom number generators: $x_{n+1} = (ax_n + c) \pmod{m}$

Hash function.

§5 Induction and Recursion.

Weak Principle of Mathematical Induction

(a) If the statement $P(b)$ is true — Basic Step, Inductive Hypothesis

(b) the statement $\underline{P(n-1) \to P(n)}$ is true for all $n > b$, then $P(n)$ is true for all integers $n \geq b$ — Inductive Step, Inductive Conclusion.

$\downarrow$

$P(b) \wedge P(b+1) \wedge \cdots \wedge P(n-1) \to P(n)$

(Strong)

Recursion: $\to$ inductive analysis

~~Recurrences~~: or use recurrences (递推)

Recurrence:

- $T(n) = rT(n-1) + a$, $T(0) = b$

  $T(n) = r^n b + a \sum_{i=0}^{n-1} r^i = r^n b + a \frac{1-r^n}{1-r}$ (by induction)

First-Order Linear Recurrences:

$\boxed{T(n) = f(n)T(n-1) + g(n)}$ < first-order because only depend on $T(n-1)$; linear because $T(n-1)$ only appears to the first power

When $f(n) = r$, $T(n) = r^n T(0) + \sum_{i=1}^{n} r^{n-i} g(i)$ (by induction)

Th: $\sum_{i=1}^{n} i x^i = \frac{n x^{n+2} - (n+1) x^{n+1} + x}{(1-x)^2}$

Divide and conquer algorithms.

$$T(n) = \begin{cases} \text{something given.} & \text{if } n \leq n_0 \\ r \cdot T(n/m) + a & \text{if } n > n_0 \end{cases}$$

Binary search example: $T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$ (assume n is a power of 2)

Iterating Recurrences: (迭代递推)

examples: $T(n) = \begin{cases} 1 \\ T(n/2) + 1 \end{cases}$

$T(n) = T(\frac{n}{2}) + 1$

$= T(\frac{n}{2^2}) + 2$

$\cdots$

$= T(\frac{n}{2^{\log_2 n}}) + \log_2 n = 1 + \log_2 n$

$T(n) = \begin{cases} 1 \\ T(n/2) + n \end{cases}$

$T(n) = T(\frac{n}{2}) + n$

$= T(\frac{n}{2^2}) + \frac{n}{2} + n$

$= \cdots$

$= T(\frac{n}{2^{\log_2 n}}) + \frac{n}{2^{\log_2 n - 1}} + \cdots + \frac{n}{2} + n$

$= 1 + 2 + \cdots + \frac{n}{2} + n = \Theta(n)$

$T(n) = \begin{cases} 1 \\ 3T(n/3) + n \end{cases}$

$T(n) = 3T(\frac{n}{3}) + n$

$= 3^2 T(\frac{n}{3^2}) + 2n$

$\cdots$

$= 3^{\log_3 n} T(\frac{n}{3^{\log_3 n}}) + n\log_3 n = n + n\log_3 n$

Th. $T(n) = aT(n/2) + n$, a is positive integer, $T(1)$ is nonnegative

1. If $a < 2$, then $T(n) = \Theta(n)$
2. If $a = 2$, then $T(n) = \Theta(n \log n)$
3. If $a > 2$, then $T(n) = \Theta(n^{\log_2 a})$

☆ Th. The Master Theorem.

$T(n) = aT(n/b) + cn^d$

If $a < b^d$, then $T(n) = \Theta(n^d)$

If $a = b^d$, then $T(n) = \Theta(n^d \log n)$

If $a > b^d$, then $T(n) = \Theta(n^{\log_b a})$

§6 Counting.

The product Rule: A count decomposes into a sequence of <u>dependent</u> counts.

$$n = n_1 \cdot n_2 \cdots\cdots n_k$$

The Sum Rule: A count decomposes into a set of <u>independent</u> counts.

$$n = n_1 + n_2 + \cdots + n_k .$$

use tree diagrams

Pigeonhole principle (鸽巢原理): N objects into k bins, there at least one bin contain ~~contain~~ at least $\lceil N/k \rceil$ objects

Inclusion-Exclusion Principle: $\left|\bigcup_{i=1}^{n} E_i\right| = \sum_{k=1}^{n} (-1)^{k+1} \sum_{1 \le i_1 < i_2 < \cdots < i_k \le n} |E_{i_1} \cap E_{i_2} \cap \cdots \cap E_{i_k}|$ (proof: P556)

Permutation: $P(n,k) = n(n-1)(n-2)\cdots(n-k+1) = \frac{n!}{(n-k)!}$

$$C(n,k) = \frac{n!}{k!(n-k)!} = \binom{n}{k} \qquad \binom{n}{k} = \binom{n}{n-k}$$

$$\sum_{i=0}^{n} \binom{n}{i} = 2^n \qquad \binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

The Binomial Theorem:

$$(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^{n-i} y^i$$

§8 Advanced Counting Techniques

Solving linear Recurrence Relations.

Def: $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$, $c_1, \cdots, c_k$ are $\in \mathbb{R}$, $c_k \neq 0$

linear, degree k, all terms are multiples of a_j's (homogeneous), constant coefficients

Consider Degree 2:

$a_n = c_1 a_{n-1} + c_2 a_{n-2}$

Characteristic equation (CE): $r^2 - c_1 r - c_2 = 0$

If it has 2 roots $r_1 \neq r_2$, then ~~the~~ $\underline{a_n = \alpha_1 r_1^n + \alpha_2 r_2^n}$ is a solution of the recurrence relation

If it has only 1 root, then $\underline{a_n = \alpha_1 r_0^n + \alpha_2 n r_0^n}$

$a_n = \sum_{i=1}^{k} c_i a_{n-i}$

CE: $r^k - \sum_{i=1}^{k} c_i r^{k-i} = 0$

If it has k distinct roots, then $\underline{a_n = \sum_{i=1}^{k} \alpha_i r_i^n}$.

If it has t roots $r_1, r_2, \cdots r_t$ with multiplicities $m_1, \cdots, m_t$ 根的 重数, $m_1 + \cdots + m_t = k$

then $\underline{a_n = \sum_{i=1}^{t} \left(\sum_{j=0}^{m_i - 1} \alpha_{i,j} n^j \right) r_i^n}$.

Linear Nonhomogeneous Recurrence Relations.

$$a_n = \underbrace{c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}}_{\text{associated} \sim \text{relation.}} + \underbrace{F(n)}_{\text{depend only on } n}$$

solution $a_n = p(n) + \underbrace{h(n)}_{\text{to this}}$

calculate $p(n)$, 代入原式, 只考虑和 n 有关的项及常数项

Generating Function:

~~G~~ $G(x) = a_0 + a_1 x + \cdots + a_n x^n$

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

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

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

in Counting

$(x^2 + x^3 + x^4)^3 \leftarrow$ coefficient of x^8

$(1+x)^n = \sum_{k=0}^{n} C(n,k) x^k$

$(1+ax)^n = \sum_{k=0}^{n} C(n,k) a^k x^k$

$(1+x^r)^n = \sum_{k=0}^{n} C(n,k) x^{rk}$

$\frac{1-x^{n+1}}{1-x} = \sum_{k=0}^{n} x^k = 1 + x + x^2 + \cdots + x^n$

$\frac{1}{1-x} = \sum_{k=0}^{\infty} x^k = 1 + x + x^2 + \cdots$

$\frac{1}{1\cancel{+}-ax} = \sum_{k=0}^{\infty} a^k x^k = 1 + ax + a^2 x^2 + \cdots$

$\frac{1}{1-x^r} = \sum_{k=0}^{\infty} x^{rk} = 1 + x^r + x^{2r} + \cdots$

$\frac{1}{(1-x)^2} = \sum_{k=0}^{\infty} (k+1) x^k = 1 + 2x + 3x^2 + \cdots$

$\frac{1}{(1-x)^n} = \sum_{k=0}^{\infty} C(n+k-1, k) x^k$

$\frac{1}{(1+x)^n} = \sum_{k=0}^{\infty} C(n+k-1, k) (-1)^k x^k$

$\frac{1}{(1-ax)^n} = \sum_{k=0}^{\infty} C(n+k-1, k) a^k x^k$

$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \cdots$

$\ln(1+x) = \sum_{k=0}^{\infty} \frac{(-1)^{k+1} x^k}{k} = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots$

§9 Relations

Binary Relation. (from A to B is a subset of a Cartesian Product $A\times B$) ($R\subseteq A\times B$)

aRb : $(a,b)\in R$ $\quad a \not R b$: $(a,b)\notin R$

The number of relation on $|A|=n$ is 2^{n^2}

properties:

Reflexive : $(a,a)\in R$ for ~~all~~ every element $a\in A$

Irreflexive: $(a,a)\notin R$ for every element $a\in A$

Symmetric : $(b,a)\in R$ whenever $(a,b)\in R$ for all $a,b\in A$

Antisymmetric: $(b,a)\in R$ and $(a,b)\in R$ implies $a=b$ for all $a,b\in A$

Transitive: $(a,b)\in R$ and $(b,c)\in R$ implies $(a,c)\in R$ for all $a,b,c\in A$

Composite:

$R=\{(1,0),(1,2),(3,1),(3,2)\}$

$S=\{(0,b),(1,a),(2,b)\}$

$S\circ R=\{(1,b),(3,a),(3,b)\}$

$M_R\odot M_S=\begin{pmatrix}1&1\\1&0\end{pmatrix}$

power R^n:

$R^1=R$, $R^{n+1}=R^n\circ R$

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

The number of reflexive relation on $|A|=n$ is $2^{n(n-1)}$

n-ary Relation:

$R\subseteq A_1\times A_2\times\cdots\times A_n$

Selection : $C: A\to\{\cancel{A}T,F\}$
$(A_1\times A_2\times\cdots\times A_n)$

projection: $P_{\{i_m\}}: A\to A_{i_1}\times\cdots\times A_{i_m}$
$1\le i_k\le n$ for all $1\le k\le m$

reflexive closure
symmetric
transitive

Finding a $\boxed{\text{transitive closure}}$ corresponds to finding all pairs of ~~elements~~ that are connected with a directed path

equal

The $\boxed{\text{connectivity relation}}$ R^* consists of all pairs (a,b), s.t. there is a path between a and b in R. $R^*=\bigcup_{k=1}^{\infty}R^k$

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

Warshall :

① 确定行
② 确定列
③ 变数值

} 循环k次

$W_0=\begin{pmatrix}0&0&0&1\\1&0&1&0\\1&0&0&1\\0&0&1&0\end{pmatrix}$ $\quad W_1 = W_2 = \begin{pmatrix}0&0&0&1\\1&0&1&1\\1&0&0&1\\0&0&1&0\end{pmatrix}$

在计算W_k时 {
第k列中，位为1的行
第k行中，位为1的列
交叉处位为0的变1

$W_3=\begin{pmatrix}0&0&0&1\\1&0&1&1\\1&0&0&1\\1&0&1&1\end{pmatrix}$ $\quad W_4=\begin{pmatrix}1&0&1&1\\1&0&1&1\\1&0&1&1\\1&0&1&1\end{pmatrix}$

```
for k:=1 to n
  for i:=1 to n
    for j:=1 to n
      Wij := Wij V (Wik Λ Wkj)
```

Equivalence Relation.

- reflexsive
- Symmetric
- transitive

The union of all the equivalence classes of R is A.

$$A=\bigcup_{a\in A}[a]_R.$$

Equivalence class

$[a]_R=\{b:(a,b)\in R\}$ R是等价关系

Partial Ordering.

- reflexsive
- antisymmetric.
- transitive

poset (S,R) : partial ordered set.

Comparability (两个元素之间)

Def: The elemets a and b of a poset $(S,\preceq)$ are comparable if either $a\preceq b$ or $b\preceq a$.

e.g $S=\{1,2,3,4,5,6\}$. R: "$|$"

2.4 comparable, 3.5 incomparable

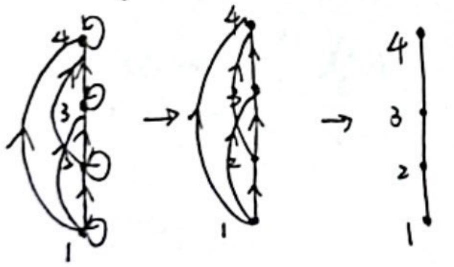
Total Ordering: $(S,\preceq)$ is a poset and every two elements of S are comparable (a chai

Lexicographic Ordering. given $(A_1,\preceq_1),(A_2,\preceq_2)$

on $A_1\times A_2$ i.e. $(a_1,a_2)\preceq(b_1,b_2)$ if $a_1\prec_1 b_1$ or if $a_1=b_1$ and $a_2\preceq_2 b_2$

(字典序)

Hasse Diagram (何塞图): 去除reflexive和transitive得到的边



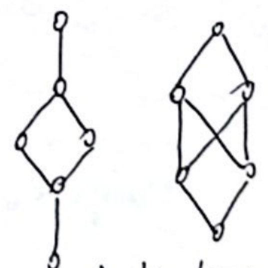
maximal (minimal): a in poset $(S,\preceq)$ if there is no $b\in S$ s.t. $a\prec b$ ($b\prec a$)

upper bound of A : u if $a\preceq u$ for all $a\in A$

least upper bound: less than any upper bound.

Topological Sorting:

Given a partial ordering R, find a total ordering $\preceq$ s.t. $a\preceq b$ whenever aRb, $\preceq$ is said compatible with R



lattice

every pair has least upper bound and greatest lower bound.

§10 Graph

vertices V 顶点

edges E 边

Simple graph: no multiple edges and on loop to itself

Complete graph: K_n

Neighborhood $N(v)$: The set of neighbors of a vertex v.

degree $\deg(v)$: in-degree $\deg^-(v)$ out-degree $\deg^+(v)$

无向图有m条边, $2m = \sum \deg(V)$

有向图 $\sum \deg^-(v) = \sum \deg^+(v)$

cycle C_5 wheel W_4 K_4

Bipartite Graph (二分图): every edge connects a vertex in V_1 and a vertex in V_2 边可涂色

完全二分图:

$K_{2,3}$

最大匹配:最多的边数的匹配方法

Complete matching from V_1 to V_2: $|M| = |V_1|$

Hall's Marriage: has complete matching from V_1 to V_2

$\iff |N(A)| \geq |A|$ for all subsets A of V_1

adjacency list:

a	bce
b	a
c	ade
d	ce

无重边

Adjacency matrices

$$\begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix}$$

Incidence matrices

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

一列表示一条边

Isomorphism 同构 存在双射, a, b 连边 $\iff f(a), f(b)$ 连边.

path: a sequence of edges

circuit: 起终同一点.

simple: 不重复经过顶点

连通分量: G的连通子图,但不是其他连通子图的真子图.

connected 连通

无向图:任意点对间有 path

有向图:

strongly connected: $a \to b$, and $b \to a$.

weakly: 转为无向图后连通

Counting Paths.

A^r (r是长数)

割点,割边:破坏图的连通性

Euler Path (circuit): a simple ~~circuit~~ path (circuit) containing every edge of G $\iff$ 有且刚好2个degree是奇数.

$\iff$ degree 都是偶数

Hamilton Path (circuit): ..... containing every vertex exactly once

G simple graph $n \geq 3$ $\begin{cases} \text{degree}(u) \geq n/2 \\ \text{or} \\ \deg(u) + \deg(v) \geq n \ (u, v \text{ nonadjacent}) \end{cases}$ has a Hamilton circuit.

Dijkstra.

(1) $d(v_0)=0$, $d(v)=\infty$, $S=\phi$

(2) ~~whi~~ while $S\neq V$ $\quad\quad O(n^2)$

let $v\notin S$ be the vertex with the least $d(v)$

$S=S\cup\{v\}$

for each $u\notin S$, $d(u)=\min\{d(u), d(v)+\alpha(u,v)\}$

Planar

Euler's Formula : $r=e-v+2$

The degree of a ~~reig~~ region : number of edges on the boundary of this region.

(割边两次计算两次

Cor1 $\boxed{e\leq 3v-6}$ (在 connected planar simple graph 中,且 $v\geq 3$)

Proof:

$$2e=\sum_{\text{all regions } R}\deg(R)\geq 3r.$$

The degree of regions is at least 3.

Cor2. Connected planar simple graph

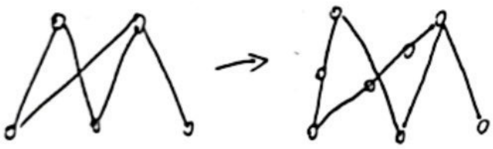
. G has a vertex of degree not exceeding 5.

Cor 3. connected planar simple graph

$v\geq 3$, no circuits of length 3

$\boxed{e\leq 2v-4}$

Kuratowski's Theorem.



Four Color Theorem.

§11 Tree

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

Th. An undirected graph is a tree if and only if there is a unique simple path between any two of its vertices.

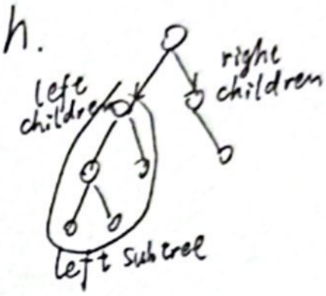
Rooted tree: have root, directed

Th. A full m-ary tree with i internal vertices has $n = mi + 1$ vertices.

level (对一个点，路径长数) height: the maximum of level.

Def. A rooted m-ary tree of height h is balanced if all leaves are at levels h or $h-1$

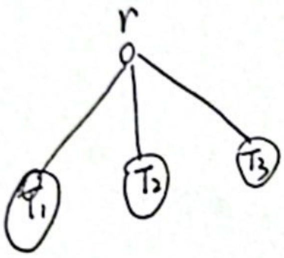
Th. There are at most m^h leaves in an m-ary tree of height h.



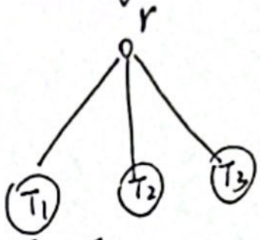
Tree Traversal

preorder:

r, T_1, T_2, T_3



Inorder



T_1, r, T_2, T_3

postorder: T_1, T_2, T_3, r

Polish notation: 先序.
(Prefix)

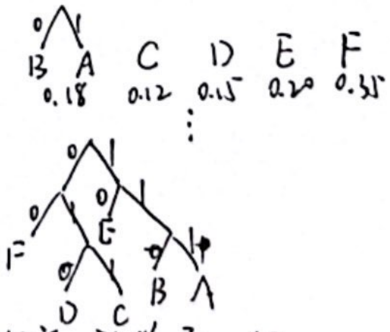
Spanning Tree: contain every vertex DFS BFS
graph connected $\Leftrightarrow$ has a spanning tree.

Prim's Algorithm
每次找一条与已有树相连的最短边. $e\log v$
开始:任一条最短的边

Kruskal's Algorithm.
先给边排序，加入最小边后看是否会生成环 $e\log e$

树的应用:
1. 二叉搜索树.
顶点的关键字大于左子树全部，小于右子树全部
2. 决策树
根树可以建模一系列决策，每个内点对应一次决策，问题的解对应这个根树通向叶子节点的通路.
3. 前缀码
任何一个字符的编码都不能是其他字符编码的前缀

频率: A:0.10 B:0.08 C:0.12 D:0.15 E:0.20 F:0.35



频率大放前面，短.