Review1. Logic

2019年5月27日 18:37

Proof examples

- Proof methods: direct proof, proof by contrapositive, proof by contradiction.
- Euler Circuit: if and only if a connected multigraph with at least 2 vertices has each of its vertex has even degree.

Logic

- Proposition logic: declarative statement either true or false.
  - *Also called atomic (elementary) proposition.*
  - *Propositional logic refer to objects and their properties and relations.*
- Logical connectives: connect to form compound propositions（复合命题）.
  - Negation, conjunction（合取，并）, disjunction（析取，或）, exclusive or（异或）, implication（蕴含式，→）, biconditional（等值，↔）.
    - Implication: $p \rightarrow q$, p is hypothesis, q is conclusion. Equals p is sufficient for q, or q is necessary for p.
      - Converse, contrapositive, inverse.
    - Biconditional proposition: p iff q.
- Translation: use logic symbol to present language.

Truth table

- Constructing the truth table: 2^n entities for n variables proposition.
- Bit string: sequence of 0 or more bits.

Tautology and contradiction

- Tautology: always true compound proposition.
  - Contradiction: always false compound proposition.
  - Contingency: neither tautology nor contradiction.
- Equivalent: two propositions always have the same truth table.
  - Logically equivalent: a pair of biconditional propositions is tautology, $p \equiv q$ *or* $p \Leftrightarrow q$.

表 1-15 逻辑等价

等价关系	名称
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	恒等律
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	支配律
$p \vee p \equiv p$ $p \wedge p \equiv p$	幂等律
$\neg(\neg p) \equiv p$	双非律
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	交换律
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	结合律
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	分配律
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	德摩根定律
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	吸收律
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	否定律

表 1-16 涉及条件语句的逻辑等价

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

表 1-17 涉及双条件的逻辑等价

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

Quantifiers and predicate

- Quantifiers: universal quantifier (all), existential quantifier (at least one).
- Predicate logic: constant, variable, and predicate.
  - Universe (domain), truth set, truth value.
- De Morgan Law for quantifiers:

Negation	Equivalent Statement
$\neg \exists x \, P(x)$	$\forall x \, \neg P(x)$
$\neg \forall x \, P(x)$	$\exists x \, \neg P(x)$

- Nested quantifiers: more than one quantifiers in a predicate logic.
  - Order of quantifiers, negation nested quantifier.

Review2. Mathematical Proofs

2019年5月27日　18:37

Theorems and proofs

- Axiom (postulate): considered to be true.
  - Theorem: proved to be true.
  - Lemma: proved to be true, used to prove other theorems.
- Proof based on logical equivalences.

推理规则	永真式	名称
p $p \to q$ $\therefore q$	$[p \land (p \to q)] \to q$	假言推理
$\neg q$ $p \to q$ $\therefore \neg p$	$[\neg q \land (p \to q)] \to \neg p$	取拒式
$p \to q$ $q \to r$ $\therefore p \to r$	$[(p \to q) \land (q \to r)] \to (p \to r)$	假言三段论
$p \lor q$ $\neg p$ $\therefore q$	$[(p \lor q) \land \neg p] \to q$	析取三段论
p $\therefore p \lor q$	$p \to (p \lor q)$	附加
$p \land q$ $\therefore p$	$(p \land q) \to p$	化简
p q $\therefore p \land q$	$[(p) \land (q)] \to (p \land q)$	合取
$p \lor q$ $\neg p \lor r$ $\therefore q \lor r$	$[(p \lor q) \land (\neg p \lor r)] \to (q \lor r)$	消解

- Proof methods: direct proof, by contrapositive, by contradiction, by cases, proof of equivalence.
  - Vacuous proof: prove $\boldsymbol{p} \to \boldsymbol{q}$ by prove p is always false.
    - Trivial proof: prove q is always true.

Review3. Sets and Functions

2019年5月27日 18:37

Sets

- Set: unordered collection of objects (elements, members).
  - *Built with sets: combinations, relations, graphs.*
  - *Axiomatic set theory: avoid Russell's paradox.*
- Venn diagram: visualize sets.
- Proper subset: belongs to but not equal.
  - Two sets are equal if and only if each is a subset of the other.
- Cardinality: number of distinct elements in a finite set.
- Power set: the set of all subsets of set S, denoted by P(S).
- Tuples: ordered n-tuple with n elements and in order.
- Cartesian product: $\boldsymbol{A \times B}$, all ordered pairs $\boldsymbol{(a, b)}$ of elements in A, B.
  - Relation A from B: a subset of the Cartesian product.
- Union, intersection, complement, difference.
  - Union of a collection of sets and intersection of a collection of sets.
  - Disjoint: intersection is empty set.
  - Inclusion and exclusion: $|\boldsymbol{A \cup B}| = |\boldsymbol{A}| + |\boldsymbol{B}| - |\boldsymbol{A \cap B}|$
- Set identities:

等　式	名　称
$A\cup\varnothing=A$ $A\cap U=A$	恒等律
$A\cup U=U$ $A\cap\varnothing=\varnothing$	支配律
$A\cup A=A$ $A\cap A=A$	幂等律
$\overline{(\overline{A})}=A$	补集律
$A\cup B=B\cup A$ $A\cap B=B\cap A$	交换律
$A\cup(B\cup C)=(A\cup B)\cup C$ $A\cap(B\cap C)=(A\cap B)\cap C$	结合律
$A\cap(B\cup C)=(A\cap B)\cup(A\cap C)$ $A\cup(B\cap C)=(A\cup B)\cap(A\cup C)$	分配律
$\overline{A\cup B}=\overline{A}\cap\overline{B}$ $\overline{A\cap B}=\overline{A}\cup\overline{B}$	德摩根定律
$A\cup(A\cap B)=A$ $A\cap(A\cup B)=A$	吸收律
$A\cup\overline{A}=U$ $A\cap\overline{A}=\varnothing$	补律

  - Prove set identities using membership tables.
- Represent sets in computer: bit string to universal set and set in-set-element to 1.

$U = \{1, 2, 3, 4, 5\}$

$A = \{2, 5\}$ – $A = 01001$

$B = \{1, 5\}$ – $B = 10001$

Functions

- Function from A to B: exactly one element of B to each element of A.
  - *A is domain, B is codomain.*
  - *For* $f(a) = b$, *b is image and a is preimage.*
    - *Range of function: set of images of elements of A.*
- Injective (one-to-one) function: $f(x) = f(y)$ ***implies*** $x = y$
- Surjective (onto) function: for every element in B there is an element in A such that $f(a) = b$
  - Bijection: both one-to-one and onto.
- Inverse function: only for bijective function.
- Composition of functions: $(f \circ g)(x) = f(g(x))$
  - Identity function maps element to itself.
- Floor function, ceiling function.

Sequence

- A function from subset of integers to a set S $\{a_n\}$.

和	闭 形 式	和	闭 形 式
$\sum_{k=0}^{n} ar^k (r \neq 0)$	$\frac{ar^{n+1} - a}{r - 1}, r \neq 1$	$\sum_{k=1}^{n} k^3$	$\frac{n^2 (n+1)^2}{4}$
$\sum_{k=1}^{n} k$	$\frac{n(n+1)}{2}$	$\sum_{k=0}^{\infty} x^k, \lvert x \rvert < 1$	$\frac{1}{1-x}$
$\sum_{k=1}^{n} k^2$	$\frac{n(n+1)(2n+1)}{6}$	$\sum_{k=1}^{\infty} kx^{k-1}, \lvert x \rvert < 1$	$\frac{1}{(1-x)^2}$

  - Arithmetic progression (initial term and common difference), geometric progression (initial term and comm ratio).
    - Sum of geometric progression: $S = \Sigma(ar^j) = \frac{a(r^{n+1} - 1)}{r-1}$
  - Recursively defined sequences: previous elements and initial element.
- Countable set: finite or has the same cardinality as Z^+.
  - To prove same cardinality: find a bijective function.

Review4. Complexity of Algorithms

2019年5月27日 18:37

Algorithms

- Algorithm: finite sequence of precise instructions for performing a computation or for solving a problem.
- Big-O notation: $f(n)=O(g(n))$ when for some positive constant c and $n>n_0$ there is $|f(n)\leqslant c|g(n)/$
  - $(f_1+f_2)(x)=O(\max(|g_1(x)|,|g_2(x)|))$
  - $(f_1f_2)(x)=O(g_1(x)g_2(x))$
- Big-Omega notation: $f(n)=\Omega(g(n))$ for some positive integer c and $n>n_0$ there is $/f(n)/\geqslant c|g(n)|$
  - Big-O is an upper bound while big-Ω is a lower bound.
- $f(n)=O(g(n))$ *and* $g(n)=O(f(n))\rightarrow f(n)=\Theta(g(n))$
- Compute time complexity: give steps and find time complexity for each step.
  - *Best case, worst case, and average case.*
- NP-Complete problem: any one of NP-Complete problems has an efficient solution then all the them have efficient solutions.
  - Input size: the minimum number of bits needed to encode the input of the problem.
  - Class P: all decision problem solvable in polynomial time.
  - Certificate: corresponding to a yes-input. NP problem can be verified certificate in polynomial time.
  - Problems belongs to NP: composite, D subset sum, SAT problem.
- Boolean formula satisfiable: assign truth values to acquire final result 1.

Review5. Number Theory

2019年5月27日 18:37

Number theory

- Division: $a \,/\, b$ *if exists interger c has* $b = ac$*, or* $\frac{b}{a}$ *is integer*
  - Note a is factor of b and b is multiple of a.
  - *If* $a|b$ *and* $a|c$ *then* $a/(b+c)$
  - *If* $a|b$ *then* $a|bc$ *for all integers c*
  - *If* $a|b$ *and* $b|c$ *then* a/c
  - $a = dq + r$, q is quotient and r is remainder.
- Congruence: a is congruent to b modulo m if $m/(a-b)$, denoted $a \equiv b\,(mod\ m)$
  - Integers a and b are congruent modulo m if and only if there is an integer k such that $a = b + km$, or $a\ mod\ m = b\ mod\ m$
  - *If* $a \equiv b(mod\ m)$ *and* $c \equiv d(mod\ m)$ *then* $a + c \equiv b + d(mod\ m)$ *and* $ac \equiv bd\,(mod\ m)$
- Arithmetic modulo m:
  - $a +_m b = (a + b)\ mod\ m,\ a \cdot_m b = ab\ mod\ m$
  - Closure, associativity, identity elements 0 and 1, additive inverses, commutativity, distributivity.

I'll skip number systems part, as it's basic in previous course.

Prime

- Positive integer greater than 1 and divisible only by 1 and itself is prime.
  - *Composite: not prime.*
- Fundamental theorem of arithmetic: every integer greater than 1 can be written uniquely as a prime or as the product of two or more primes.
- GCD (greatest common divisor) and LCM (least common multiple).
  - Relatively prime: $\gcd(a, b) = 1$
  - Find gcd: factorization (can also find lcm), Euclidean algorithm.
    - $a = bq + r$ *then* $\gcd(a, b) = \gcd(b, r)$
  - Bezout's identity: $\gcd(a, b) = sa + tb$
    - *If* $\gcd(a, b) = 1$ *and* a/bc *then* a/c
    - *If* $ac \equiv bc(mod\ m)$ *and* $\gcd(c, m) = 1$ *then* $a \equiv b\,(mod\ m)$
- Mersenne primes: prime form $2^p - 1$, and p is prime.
- Goldbach's conjecture: integer $n > 2$ is the sum of two primes.

Linear congruence

- Congruence form $ax \equiv b\,(mod\ m)$
- Inverse of a modulo m: $\bar{a}a \equiv 1\,(mod\ m)$
  - Solution to linear congruence: $x \equiv \bar{a}b\,(mod\ m)$
    - $ax \equiv b(mod\ n) \rightarrow d = \gcd(a, n)$ *and* $d/b \rightarrow ra + sn = d$ *and* $x_0 = \frac{rb}{d} \rightarrow x \equiv \frac{rb}{d}\,(mod\,\frac{n}{d})$
- Chinese reminder theorem:

$$x \equiv a_1 \pmod{m_1}$$
$$x \equiv a_2 \pmod{m_2}$$
$$\cdots$$
$$x \equiv a_n \pmod{m_n}$$

- $m = m_1 m_2 \cdots m_n$ and $M_k = \frac{m}{m_k}$ and $\gcd(m_k, M_k) = 1 \rightarrow y_k M_k \equiv 1 (mod\ m_k) \rightarrow x_0 = a_1 M_1 y_1 + a_2 M_2 y_2 + \quad + a_n M_n y_n \rightarrow x \equiv x_0\ (mod\ m)$
- Another solution:

  $2x \equiv 2 \pmod 6$
  $3x \equiv 2 \pmod 7$
  $2x \equiv 4 \pmod 8$

  首先求解第一个方程，得到$x \equiv 1 \pmod 3$，于是令$x = 3k + 1$，第二个方程就变为：

  $9k \equiv -1 \pmod 7$

  解得$k \equiv 3 \pmod 7$。于是，再令$k = 7l + 3$，第三个方程就可以化为：

  $42l \equiv -16 \pmod 8$

  解出：$l \equiv 0 \pmod 4$，即 $l = 4m$。代入原来的表达式就有 $x = 21(4m) + 10 = 84m + 10$，即解为：

  $x \equiv 10 \pmod{84}$

- *Modular arithmetic and congruencies: pseudorandom number generators, hash functions, cryptography.*
  - Pseudorandom number generator: modulus m, multiplier a, increment c, seed x_0, $x_{n+1} = (ax_n + c)\ (mod\ m)$
  - *Cryptography: symmetric cryptography, asymmetric cryptography, RSA cryptography, DLP and EI Gamal cryptography, Diffie-Hellman key exchange protocol, and cryptocurrency.*
- Fermat's little theorem: p is prime, $x \nmid p \rightarrow x^{p-1} \equiv 1\ (mod\ p)$
- Euler' totient function Φ: $x^{\Phi(n)} \equiv 1\ (mod\ n)$
- Cryptography: kryptos (secret) and graphos (writing).
  - RSA public key cryptosystem: large primes p and q, $n = pq$ and $\Phi(n) = (p-1)(q-1) \rightarrow \gcd(e, \Phi(n)) = 1$ and $ed \equiv 1(mod\ \Phi(n)) \rightarrow C = M^e\ mod\ n$ and $M = C^d\ mod\ n$
    - C is encryption and M is decryption.
  - *I'll skip other methods of cryptography and if you are interested, you can join cryptography course next semester.*

Review6. Mathematical Induction

2019年5月27日 18:37

Mathematical induction

- Start from small examples, then suppose for k case the proposition establishes, finally prove k+1 is also establish.
  - Basic step, inductive hypothesis, and inductive step.
  - Strong principle and weak principle.
- Well-ordering principle: every set of non-negative integers has a smallest element.

Review7. Recursion

2019年5月27日 18:37

Recursion

- Inductive analysis (prove correctness), towers of Hanoi.
- Recurrences: function defined on the set of n-1 values.
    - Initial condition(s), base case(s).
- Iterating a recurrence: bottom-up, top-down.
    - $T(n) = rT(n-1) + g(n)$ *and initial condition* $T = a \rightarrow T(n)$ $= r^n a + \sum_{i=1}^{n} r^{n-i} g(i)$

Divide and conquer

- For given form: $T(n) = rT(n-1) + 1$ *with initial condition* $T = b \rightarrow T(n) = r^n b + \frac{a(1-r^n)}{1-r}$
- Divide and conquer formula: $T(n) = rT\left(\frac{n}{m}\right) + a$
    - *Example: binary search.*

Linear recurrence relation

- Degree k relation with constant coefficients: $a_n = c_1 a_{n-1} + c_2 a_{(n-2)} + \quad + c_k a_{n-k}$
    - Linear, homogeneous (all terms are multiples of a_j's), degree k, constant coefficients.
    - CE (characteristic equation): $r^k - \sum_{i=1}^{k} c_i r^{k-1} = 0$
        - If CE has k distinct roots r_i, then $a_n = \sum_{i=1}^{k} a_i r_i^n$
- *You are an adult and you should solve these questions by yourself.*
- Degenerate roots in general: t roots $r_1, \cdots, r_t$ with multiplicities $m_1, \cdots, m_t$
    - $a_n = \sum_{i=1}^{t} \left(\sum_{j=0}^{m_i - 1} a_{i,j} n^j \right) r_i^n$

Generating function

- *Used to characterize sequences.*
- $G(x) = a_0 + a_1 x + \quad + a_k x^k + \quad = \sum_{k=0}^{\infty} a_k x^k$
    - Generate function of $a_k = \binom{m}{k}$: $G(x) = (1+x)^m$
- Addition and multiplication of generating function.

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

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

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

$$\frac{1-x^{n+1}}{1-x} = \sum_{k=0}^{n} x^k = 1 + x + x^2 + \cdots + x^n$$
$$\frac{1}{1-x} = \sum_{k=0}^{\infty} x^k = 1 + x + x^2 + \cdots$$
$$\frac{1}{1-ax} = \sum_{k=0}^{\infty} a^k x^k = 1 + ax + a^2x^2 + \cdots$$
$$\frac{1}{1-x^r} = \sum_{k=0}^{\infty} x^{rk} = 1 + x^r + x^{2r} + \cdots$$
$$\frac{1}{(1-x)^2} = \sum_{k=0}^{\infty}(k+1)x^k = 1 + 2x + 3x^2 + \cdots$$
$$\frac{1}{(1-x)^n} = \sum_{k=0}^{\infty} C(n+k-1,k)x^k$$
$$\frac{1}{(1+x)^n} = \sum_{k=0}^{\infty} C(n+k-1,k)(-1)^k x^k$$
$$\frac{1}{(1-ax)^n} = \sum_{k=0}^{\infty} C(n+k-1,k)a^k x^k$$
$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$
$$\ln(1+x) = \sum_{k=0}^{\infty} \frac{(-1)^{k+1}x^k}{k} = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots$$

Review8. Counting

2019年5月27日　　18:37

Counting

- *Determine the number of these objects.*
    - The product rule and the sum rule or combination.
    - Inclusion-exclusion principle: $|A \cup B| = |A| + |B| - |A \cap B|$

      $$|\cup_{i=1}^{n} E_i| = \sum_{k=1}^{n} (-1)^{k+1} \sum_{1 \le i_1 < i_2 < \cdots < i_k \le n} |E_{i_1} \cap E_{i_2} \cap \cdots \cap E_{i_k}|$$

        - *Determine the number of onto functions, the appears on tree leaves.*
- Pigeonhole principle: a set of objects stored in a set of bins.
    - If there are k+1 objects and k bins, then there is at least one bin with two or more objects.
    - N objects in k bins, at least one bin containing $\left\lceil \frac{N}{k} \right\rceil$ objects.
    - Example: bijective functions number, counting triangles, counting pairs.
- The bijection principle: two sets have same size if and only if there is a one-to-one function from one set onto the other.

Binomial coefficient

- K-element permutation of N: a list of k distinct elements chosen from a set N.
- Unorder set formula:

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

    - $\sum_{i=0}^{n} \binom{n}{i} = 2^n$
- **Pascal's triangle**: each entry in Pascal's triangle is the sum of two entries directly above it.
    - Pascal's identity: $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$
- Algebraic proof, combinatorial proof, bijective proof, binomial proof.
    - Binomial theorem: $(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^{n-i} y^i$
        - $\sum_{i=0}^{n} \binom{n}{i} = 2^n$
    - Trinomial coefficient: $\binom{n}{k_1\ k_2\ k_3} = \frac{n!}{k_1!k_2!k_3!}$

Review9. Relations

2019年5月27日 18:37

Binary relation

- Binary relation from A to B is a subset of a Cartesian product $\boldsymbol{A \times B}$
  - Set R belongs to relation: a R b means $(\boldsymbol{a}, \boldsymbol{b}) \in \boldsymbol{R}$, noted as $\boldsymbol{a \rightarrow b}$
  - Use table to represent binary relation.
- Relation: one to many relationships between elements in A and B.
- Number of binary relations:
  - On a set A: $\boldsymbol{2^{n^2}}$
  - Reflexive: $\boldsymbol{2^{n(n-1)}}$
  - Total preorder: $\sum_{k=0}^{n} k!\,S(n, k)$
  - Total order: $n!$
  - Equivalence relation: $\sum_{k=0}^{n} S(n, k)$
- Reflexive relation: $(a, a) \in R\ for\ every\ a \in A$
  - Irreflexive relation: $(a, a) \notin R\ for\ every\ a \in A$
- Symmetric relation: $(b, a) \in R\ and\ (a, b) \in R\ for\ all\ a, b \in A$
  - Antisymmetric relation: $(b, a) \in R\ and\ (a, b) \in R\ implies\ a = b\ for\ all\ a, b \in A$
- Transitive relation: $(a, b) \in R\ and\ (b, c) \in R\ implies\ (a, c) \in R\ for\ all\ a, b, c \in A$
- Combining relations: union, intersection, difference.
- Composite of relations: composite of R and S notes $S \circ R$
  - $(a, b) \in R\ and\ (b, c) \in S \rightarrow (a, c) \in S \circ R$
  - Powers R^n defined: $R^1 = R\ and\ R^{n+1} \circ R$
  - Transitive relation if and only if $R^n \subseteq R\ for\ n = 1, 2, 3 \cdots$

N-ary relation

- *Relation on n sets, these sets are domains of R.*
  - Degree of R is n.
  - R is functional in A_i if contains at most one n-tuple $(\cdots, a_i, \cdots)$ for any value a_i within domain A_i.
- *Relational database I'll skip due to you already chosen database principle.*
- Represent relations: explicit list, table, function, zero-one matrix, directed graph.

Closure

- Reflexive closure: contains R, is reflexive, is minimal.
- Closures: with property P of relation R on set A, and S is minimal.
- Transitive closure: find all pairs of elements that are connected with a directed path.

Connectivity relation

- Consists of all pairs (a, b) there is a path between a and b in R.

- $R^* = \bigcup_{k=1}^{\infty} R^k$
- The transitive closure of a relation R equals the connectivity relation R*.

Other relations

- Equivalence relation: reflexive, symmetric, transitive.
  - Equivalence class: the set of all elements related to an element of A, denoted $[a]_R$: $[a]_R = \{b : (a, b) \in R\}$
- Partition of a set S:

$$A_i \cap A_j = \emptyset,\ i \neq j \text{ and } S = \bigcup_{i=1}^{k} A_i$$

- Partial order (poset): reflexive, antisymmetric, transitive.
  - Comparability: elements in $poset\ (S, \preccurlyeq)$ are comparable if either $a \preccurlyeq b\ or\ b \preccurlyeq a$
- Total order (chain): every two elements in poset are comparable.
- Lexicographic ordering: on two posets, $(a_1, a_2) \prec (b_1, b_2) \rightarrow a_1 \prec b_1\ or\ a_1 = b_1\ then\ a_2 \prec b_2$
- Hasse diagram: representation of partial ordering.
  - Maximal and minimal elements: $no\ b \in S\ that\ a \prec b$, so does minimal.
  - Greatest and least: $b \preccurlyeq a\ for\ b \in S$, so does least.
  - Upper bound and lower bound: $a \preccurlyeq u\ for\ all\ a \in A$, so does lower bound.
    - Least upper bound and greatest lower bound.
- Well-ordered set: a poset is total ordering and every nonempty subset of S has a least element.
- Lattices: partial ordered set with every pair of elements has both least upper bound and greatest lower bound.

Review10. Graphs

2019年5月27日 18:37

Graph

- Vertices, edges (endpoints, joins, adjacent), incident (connect).
- Graph: $G = (V, E)$
  - Simple graph, multigraph pseudograph: edges number and loop.
  - Complete graph K_n: all vertices incident to edge.
  - Directed graph, undirected graph.
- Degree of vertex: $\deg(v)$, initial vertex and terminal vertex in directed graph, corresponding in-degree and out-degree.
- Cycle, wheel, and n-dimensional hypercube.
- Bipartite graphs: partitioned into two disjoint subsets.
  - Complete bipartite graph, matching, and maximum matching.
- Union, intersection of graph.
- *Representation: adjacency lists, adjacency matrices, incidence matrices.*
- Isomorphism: a and b are adjacent in G_1 if and only if f(a) and f(b) are adjacent in G_2.
  - Function f is one-to-one and onto.
- Path (circuit, simple), length, connectivity, connected component.
  - Disconnected: cut vertices and cut edges.
- Euler Circuit: the degree of every vertex must be even, or exactly two vertices of odd degree.
- Hamilton path: a simple path passes through all vertices exactly once.
- Shortest path problem, weighted graph.

Planar graph

- Draw in plane without any edges crossing.
- Euler's formula: e edges v vertices then the number of regions in a planar representation is $r = e - v + 2$
- Degree of region: number of edges on the boundary of this region.
- Elementary subdivision: remove an edge {u, v} and add a new vertex w to {u, w}, {w, v}.