|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sets | | : {0,1,2,3,...}. : {...,-1,0,1,...}. : {1/2,-23,9.6}  : real nums. : complex nums | | | | | | | | 0 neither negative nor positive. 0 is even  : iff | | | |
| Mathematical Statements | | | | Statement/proposition is true or false but not both  Universal conditional statement (universal + conditional): can be rewritten to make them purely universal/conditional  Universal existential statement, existential universal statement, or any other combination | | | | | | | | | |
| Basic properties of Integers | | | x,y,z : Closure under addition & multiplication: x+y & xy  Commutativity: x + y = y + x and xy = yx  Associativity: x+y+z = (x+y)+z = x+(y+z) & xyz = (xy)z = x(yz)  Distributivity: x(y+z) = xy + xz and (y+z)x = yx + zx  Trichotomy: x=y or x<y or x>y (only 1 can be true) | | | | | | | | n is even an integer k s.t. n = 2k  n is odd an integer k s.t. n = 2k + 1  d is divisor/factor of n (d|n) k s.t. n = dk  (only for CS1231S) n is colorful if k s.t. n = 3k | | |
| n is prime: (n>1) r, s , (n=rs (r=1 s=n) (r=n s=1) | | | | | | | | n is composite: r,s (n=rs (1<r<n) (1<s<n)) | | |
| Rational nums | | | | r is rational a,b s.t. r = a/b and b≠0 | | | a/b is in lowest terms if the largest int that divides both a and b is 1 | | | | | | |
| Proofs | Axiom/Postulate: statement assumed to be true w/o proof  Corollary: simple deduction from theorem  Conjecture: statement believed to be true; has no proof | | | | | | | Theorem: statement proved using rigorous mathematical reasoning (major result)  Lemma: small theorem (minor result; purpose to help in proving theorem) | | | | | |
| Direct proof | | | | | 1. Let a and b be 2 consecutive odd nums ...  2. Product of 2 consecutive odd nums is always odd | | | | | | | |
| Disproof by counterexample | | | | | Proof by mathematical induction | | | | | | | Combinatorial proof |
| Proof by exhaustion | | | | | (suitable when num of cases if small)  irrational nums p and q s.t. pq is rational | | | | | | | |
| Proof by deduction  (type of direct proof; when num of cases is infinite) | | | | | 1. Let nums be n and n+1 1.1 (n+1)2-n2 = n2+2n+1-n2 = 2n+1 (By algebra)  1.2 2n+1 is odd (by defn of odd nums) 2. Diff of any 2 consecutives squares is odd | | | | | | | |
| Proof by contradiction | | | | | 1. Suppose not, i.e.... Contradiction 2. Assumption is false, i.e.... | | | | | | | |
| Proof by contraposition | | | | | Proof ~q ~p. Conclude p q | | | | | | | |
| Theorem 4.2.1 | | | | | Every integer is a rational number | | | | | | | | |
| Theorem 4.2.2 | | | | | Sum of any two rational nums is rational | | | | | | | | |
| Corollary 4.2.3 | | | | | Double of a rational number is rational | | | | | | | | |
| Theorem 4.3.1 | | | | | For all positive integers a and b, if a|b, then a ≤ b | | | | | | | | |
| Theorem 4.3.2 | | | | | The only divisors of 1 are 1 and -1 | | | | | | | | |
| Theorem 4.3.3 | | | | | Transitivity of Divisibility: For all integers a,b and c, if a|b and b|c, then a|c | | | | | | | | |
| Theorem 4.4.1 | | | | | Quotient-Remainder Theorem: Given any int n and positive int d, unique ints q and r s.t. n = dq + r and 0 ≤ r < d | | | | | | | | |
| Lemma 4.4.4 | | | | | For any r , =|r| ≤ r ≤ |r| | | | | | | | | |
| Theorem 4.4.6 | | | | | Triangle Inequality: For any x, y , |x+y| ≤ |x|+|y| | | | | | | | | |
| Theorem 4.6.1 | | | | | There is no greatest integer | | | | | | | | |
| Proposition 4.6.4 | | | | | integers n, if n2 is even, then n is even | | | | | | | | |
| Theorem 4.7.1 | | | | | √2 is irrational | | | | | | | | |
| T1Q10 | | | | | Let n be an . Then n2 is odd iff n is odd | | | | T2Q10 | | | Let a, b . If n = ab, then a ≤ or b ≤ | |

Properties of real nums (Appendix A)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F1. Commutative Laws | | | real nums a,b: a+b = b+a and ab = ba | | | F2. Associative Laws | | real nums a,b,c: (a+b)+c = a+(b+c) and (ab)c = a(bc) | | | |
| F3. Distributive Laws | | | real nums a,b,c: a(b+c) = ab+ac and (b+c)a = ba + ca | | | F4. Existence of Identity Elements | | 2 distinct real nums, 0 and 1, s.t. for every real num a: 0+a = a+0 = a and 1\*a = a\*1 = a | | | |
| F5. Existence of Additive Inverses | | | For every real num a, a real num -a of a, s.t. a + (-a) = (-a) + a = 0 | | | F6. Existence of Reciprocals | | For every real num a ≠ 0, a real num 1/a or a-1 s.t. a\*(1/a) = (1/a)\*a = 1 | | | |
| T1. Cancellation Law for Addition | | | If a+b = a+c, then b = c | | | T2. Possibility of Subtraction | | Given a and b, there is exactly one x s.t. a+x = b. This x = b-a. | | | |
| T3. | | | b-a = b+(-a) | | | T4. | | -(-a) = a | | | |
| T5. | | | a(b-c) = ab-ac | | | T6. | | 0\*a = a\*0 = 0 | | | |
| T7. Cancellation Law for Multiplication | | | If ab = ac and a ≠ 0, then b = c | | | T8. Possibility of Division | | Given a and b w a ≠ 0, there is exactly one x s.t. ax = b. This x = b/a and is the quotient of b and a | | | |
| T9. | | | If a ≠ 0, then b/a = ba-1 | | | T10. | | If a ≠ 0, then (a-1)-1 = a | | | |
| T11. Zero Product Property | | | | If ab = 0, then a = 0 or b = 0 | | T12. Rule for Multiplication w Negative Signs | | | | (-a)b = a(-c) = -(ab), (-a)(-b) = ab, | |
| T13. Equivalent Fractions Property | | | | , if b ≠ 0 and c ≠ 0 | | T14. Rule for Addition of Fractions | | | | , if b ≠ 0 and c ≠ 0 | |
| T15. Rule for Multiplication of Fractions | | | | , if b ≠ 0 and d ≠ 0 | | T16. Rule for Division of Fractions | | | | , if b ≠ 0, c ≠ 0 and d ≠ 0 | |
| Positive real nums satisfy Ord1-3 | | Ord1. For any real nums a and b, if a and b are positive, so are a+b and ab | | | | Ord2. For every real num a ≠ 0, either a is +ve or -a is +ve, but not both | | | | Ord3. 0 is not positive | |
| T17. Trichotomy Law | | For any real nums a and b, exactly 1 of the 3 relations a < b, b < a or a = b holds | | | | T18. Transitive Law | | | | If a < b and b < c, then a < c | |
| T19. | If a < b, then a+c < b+c | | | | T20. | If a < b and c > 0, then ac < bc | | | T21. | | If a ≠ 0, then a2 > 0 |
| T22. | 1 > 0 | | | | | T23. | If a < b and c < 0, then ac > bc | | | | |
| T24. | If a < b, then -a > -b. In particular, if a < 0, then -a > 0 | | | | | T25. | If ab > 0, then both a and b are positive or both are negative | | | | |
| T26. | If a < c and b < d, then a+b < c+d | | | | | T27. | If 0 < a < c and 0 < b < d, then 0 < ab < cd | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Compound Statements | ~: not/negation (others use ¬)  : and/conjunction | | | | | | | | | | | | : inclusive or/disjunction (A or B or both) unlike exclusive or (A or B but not both; XOR) | | | |
| Order of ops | | ~: performed first | | | | | | | , are coequal (use parentheses to disambiguate order of ops) | | | | | | | , are coequal: performed last |
| Statement form | Expression made up of statement vars and logical connectives that becomes a statement when actual statements are sub for the component statements vars (e.g. 3+n=9,, 2x = x2) | | | | | | | | | | | | | | | |
| Logical Equivalence | 2 statements forms are logically equivalent iff they have identical truth values for each possible sub of statements for their statement vars. P Q  Prove not equivalent: 1) construct truth table OR 2) find counter example | | | | | | | | | | | | | Tautology: statement form that is always true  Contradiction: statement form that is always false | | |
| **Theorem 2.1.1** | Commutative laws | | | | | | | | | pq qp | | | pq qp | | | |
| Associative laws | | | | | | | | | pqr (pq)r p(qr) | | | pqr (pq)r p(qr) | | | |
| Distributive laws | | | | | | | | | p(qr) (pq) (pr) | | | p(qr) (pq) (pr) | | | |
| Identity laws | | | | | | | | | ptrue p | | | pfalse p | | | |
| Negation laws | | | | | | | | | p~p true | | | p~p false | | | |
| Double negative laws | | | | | | | | | ~(~p) p | | |  | | | |
| Idempotent laws | | | | | | | | | pp p | | | pp p | | | |
| Universal bound laws | | | | | | | | | ptrue true | | | pfalse false | | | |
| De Morgan's Laws | | | | | | | | | ~(pq) ~p ~q | | | ~(pq) ~p~q (not and not OR neither nor) | | | |
| Absorption laws | | | | | | | | | p(pq) p | | | p(pq) p | | | |
| Negation of true and false | | | | | | | | | ~true false | | | ~false true | | | |
| Conditional Statements | If p then q OR p implies q OR q if p OR p q. It is false when p is true and q is false; otherwise it is true  p is the hypothesis/antecedent of the conditional and q the conclusion/consequent  Conditional statement is vacuously true/true by default (i.e. if p is false, statement as a whole is true by default) | | | | | | | | | | | | | | | |
| **Implication Law** | | | | p q ~p q (proof using truth table) | | | | | | | | | Negation of conditional statement: ~(p q) p ~q | | | |
| Contrapositive, Converse & Inverse of a conditional statement | | | | | | | Contrapositive of p q is ~q ~p  p q ~q ~p (conditional statement contrapositive)  Note p q q p ~p ~q (conditional inverse or converse) | | | | | | | | Converse of p q is q p  Inverse of p q is ~p ~q  q p ~p ~q (converse inverse) | |
| Only If and Biconditional | | | p **only if** q means if not q then not p,  ~q ~p p q | | | | | | | | | Biconditional of p and q is p **if and only if** (iff) q, p q  p q (p q) (q p) (true if both p,q have same truth values) | | | | |
| Necessary and Sufficient | r is a sufficient condition for s: if r then s OR r s  r is a necessary condtion for s: if not r then not s OR if s then r OR s r (r alone might not imply s occur) | | | | | | | | | | | | r is a necessary and sufficient condition for s: r iff s OR  r s | | | |
| Arguments | Argument form is valid iff whenever statements are substituted that make all the premises true, conclusion is also true  Premises/assumptions/hypothesis: statements except final one  Conclusion: final statement | | | | | | | | | | Testing an argument form for validity:  1. Identify premises and conclusion of the argument form  2. Construct truth table of all the premises and conclusion  3. A row of the truth table in which all the premises are true aka critical row  a) If critical row in which conclusion is false, argument form is invalid  b) If conclusion in every critical row is true argument form is valid | | | | | |
| Modus Ponens & Modus Tollens | | | | | Syllogism: argument form consisting of 2 premises and conclusion (e.g. Modus Ponens or Modus Tollens; both are valid form of argument) | | | | | | | | | | | |
| Fallacies | Error in reasoning, resulting in invalid argument (e.g. using ambiguous premises and treating them as unambiguous, circular reasoning [assuming what is to be proved w/o deriving from premises], jumping to conclusion)  Argument is sound iff it is valid, and all its premises are true (opp is unsound) | | | | | | | | | | | | | | | |
| Converse error | | | | | | | p q, q, p (aka fallacy of affirming the consequence) | | | | | | | | |
| Inverse error | | | | | | | p q, ~p, ~q | | | | | | | | |
| False premise | | | | | | | Argument is valid, but premise is false. E.g. If Singaporean, then must be 2m tall | | | | | | | | |
| Contradiction Rule | | | | | | If an assumption leads to a contradiction, then that assumption must be false (used in proof by contradiction) | | | | | | | | | | |
| **Table 2.3.1**  Rules of Inference (Form of argument that is valid) | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Modus Ponens | if p then q  p  q | Modus Tollens | if p then q  ~q  ~p | Proof by Division into Cases | pq  p r  q r  r | Conjunction | p  q  pq | | Elimination | pq OR pq  ~q ~p  p q | Transitivity | p q  q r  p r | T1Q6d | p r  q r  pq r |  |  | | Generalization | p OR q  p q p q | Specialization | pq OR pq  p q | Contradiction Rule | ~p false  p |  |  | | | | | | | | | | | | | | | | |
| Truth table | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | p | q | ~p | pq | pq | p q | ~q ~p (contrapositive) | q p (converse) | ~p ~q (inverse) | p q (iff) | | T | T | F | T | T | T | T | T | T | T | | T | F | F | F | T | F | F | T | T | F | | F | T | T | F | T | T | T | F | F | F | | F | F | T | F | F | T | T | T | T | T | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Predicate | | Predicate: sentence that contains a finite num of vars and become a statement when specific values are sub for the vars  Domain of a predicate var is the set of all values that may be substituted in place of the var  If P(x) is a predicate and x has domain D, the truth set is the set of all elements of D that make P(x) true when they are substituted for x, i.e. {xD|P(x)}, (| mean s.t. in set theory) | | | | | | | | | | | |
| Quantifier | | Universal quantifier: , for all  Let Q(x) be a predicate and D the domain of x. A universal statement is a statement of the form xD, Q(x)  Statement true iff Q(x) is true for every x in D & Statement false iff Q(x) is false for at least one x in D (counterexample) | | | | | | | | | | | |
| Existential quantifier: , there exist. !: there exists a unique  Let Q(x) be a predicate and D the domain of x. An existential statement is a statement of the form xD s.t. Q(x)  Statement true iff Q(x) is true for at least one x in D & Statement false iff Q(x) is false for all x in D | | | | | | | | | | | |
| Universal Conditional Statement | | | | | | | x (P(x)Q(x)) | | | | | | |
| Equivalent Forms | | Universal: By narrowing U to domain D consisting of all values of x that make P(x) true: xU (P(x)Q(x)) xDQ(x)  Existential: x s.t. (P(x) and Q(x)) xD s.t. Q(x), where D is the set of all x for which P(x) is true | | | | | | | | | | | |
|  | | Factor: an integer that multiplied with another integer gives n, i.e. can be negative int  Prime num: int whose positive integer factors are itself and 1 | | | | | | | | | | | |
| Implicit Quantification | | | | | | Predicate If x > 2, then x2 > 4 is implicit implying real num x, (if x > 2 then x2 > 4) | | | | | | | |
| Negation of quantified statement | | **Thrm 3.2.1:** Negation of universal statement: ~(xD, P(x)) xD s.t. ~P(x)  **Thrm 3.2.2:** Negation of existential statement: ~(xD s.t. P(x)) xD, ~P(x)  Negation of universal conditional statement: ~(x (P(x)Q(x))) xD s.t. ~(P(x)Q(x)) xD s.t. (P(x) ~Q(x)) | | | | | | | | | | | |
| Relation | | xD, P(x) P(x1) P(x2) ... P(xn) | | | | | | | | | xD s.t. P(x) P(x1) P(x2) ... P(xn) | | |
| Vacuous Truth | | x (P(x)Q(x)) is vacuously true/true by default iff P(x) is false for every x in D, i.e.  if its negation xD s.t. (P(x) ~Q(x)) is false, then original statement is true by default  Vacuous truth: true because the hypothesis (antecedent) cannot be satisfied | | | | | | | | | | | |
| Variants of Universal Conditional Statement | | | | | | xD (P(x)Q(x)), logically equivalent to contrapositive  Contrapositive: xD (~Q(x)~P(x)) | | | | | Converse: xD (Q(x)P(x))  Inverse: xD (~P(x)~Q(x)) | | |
| Necessary, Sufficient Condn, Only if | | | | x r(x) is a sufficient condition for s(x) means x (r(x)s(x))  x r(x) is a necessary condition for s(x) means x (~r(x)~s(x)) x (s(x)r(x))  x r(x) only if s(x) means x (~s(x)~r(x)) x (r(x)s(x)) | | | | | | | | | |
| Multiple Quantifiers | xD, yE s.t. P(x,y): every elem x in D, there is a y in E that "works" for x  xD s.t. yE, P(x,y): 1 x in D that "works" no matter which y in E is chosen | | | | | | | | | | | | |
| Negations of Multiple-Quantified Statements | | | | | ~(xD, yE s.t. P(x,y)) xD, yE, ~P(x,y)  ~(xD s.t. yE, P(x,y)) xD, yE s.t. ~P(x,y) | | | | | | | | |
| Order of quantifiers | In a statement containing both and , changing order of quantifiers, changes meaning of statement.  However, if both quantifiers are of same type, then order don't matter | | | | | | | | | | | | |
| Formal Logical Notation | xD, P(x) as x (xD P(x))  xD s.t. P(x) as x (xD P(x))  For this module, follow 1st type of notation | | | | | | | | 2nd type usually used in AI, automata theory, formal languages  Taken tgt, the symbols for quantifiers, vars, predicates and logical connectives is known as language of first-order logic | | | | |
| Universal instantia-tion | If some property is true of everything in the set, then it is true of any particular thing in the set  Fundamental tool of deductive reasoning | | | | | | | | Universal Modus Ponens | | | x (P(x) Q(x))  P(a) for a particular a  Q(a) | |
| Universal Modus Tollens | x (P(x) Q(x))  ~Q(a) for a particular a  ~P(a)  Used in proof of contradiction | | | | | | | | Universal Transitivity | | | x (P(x) Q(x))  x (Q(x) R(x))  x (P(x) R(x)) | |
| Validity of args w Quantified statement | Same for args w compound statements  Arg valid iff truth of its conclusion follows necessarily from truth of its premises.  **Defn 3.4.1:** arg form is valid means no matter what particular predicates are substituted for the predicate symbols in its premises, if the resulting premise statements are all true, then the conclusion is also true  Arg is valid iff its form is valid. Can use venn diagrams to show validity/invalidity of args | | | | | | | | | | | | |
| Converse, Inverse error | Converse error: x (P(x) Q(x))  Q(a) for a particular a  P(a) | | | | | | | | | | Inverse error: x (P(x) Q(x))  ~P(a) for a particular a  ~Q(a) | | |
| Rules of Inference for Quantified Statement | | | Universal instantiation | | | | | xD, P(x)  P(a) if aD | | Existential instantiation | | | xD, P(x)  P(a) for some aD |
| Universal generalization | | | | | P(a) for every aD  xD, P(x) | | Existential generalization | | | P(a) for some aD  xD, P(x) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Set Theory | Set: unordered collection of objects  Objects: members or elems of set  - Order and duplicates do not matter  If S is a set, x S means x is an elem of S  Cardinality of set S: |S| is the size of the set, i.e. num of elems | | | | | | Set Roster Notation: {1,2,3}, {1,2,3,...,100}  Set-builder notation: Set of all elems x U s.t. P(x) is true is denoted {x U:P(x)} or {x U|P(x)}  Replacement notation: Set of all objs of the form t(x) where x ranges over elems of A is denoted {t(x): x A} or {t(x)|x A} | | | | | |
| Subsets, Proper Subsets, Empty Set and Singleton | | | | | A is subset of B, i.e. A B iff every elem of A is also an elem of B or A B iff x (x A x B) | | | | | | | |
| A is a proper subset of B, A B iff A B and A ≠ B.  A B x (xA xB) | | | Empty set: Set w no elem, {}, denoted as  Singleton: set w exactly 1 elem | | | | |
| Ordered Pairs | | | | Ordered pair is an expression of the from (x,y) | | | | | | (a,b) = (c,d) (a=c) (b=d) | | |
| Cartesian Products | | | | Given sets A and B, Cartesian product of A and B, denoted AB (read as A cross B), is set of all ordered pairs (a,b) where a is in A and b is in B  AB = {(a,b): aA bB} | | | | | | is set of all ordered pairs (x,y) where x, y , i.e. Cartesian plane | | |
| Set Equality | | | | A = B iff every element of A is in B and every elem of B is in A | | | | | | A = B A B B A OR x (x A x B) | | |
| Venn Diagrams | | | | | Note {x} ≠ {x, } | | | | | | | |
| Operations on Sets | | | Universal set: e.g. certain mathematical situations all sets considered are sets of real nums  Let A and B be subsets of a universal set U  1. A B: set of all elements that are in at least 1 of A or B  2. A B: set of all elems that are common to both A and B  3. B - A or B\A (diff of B minus A/relative complement of A in B): set of all elems in B and not in A  4. (complement of A, AC): set of all elems in U not in A | | | | | | | | | (Elements mtd)  1. A B = {xU: xA xB}  2. A B = {xU: xA xB}  3. B\A = {xU: xA xB}  4. = {xU: xA} |
| = A0 A1 ... An | | | | | | = A0 A1 ... An | | | |
| Partition of Sets | | | Sets can be divided into nonoverlapping (disjoint) pieces. Such a division is called a partition, e.g. {A1, A2, A3}: partition of A  2 sets are disjoint iff they have no elems in common, i.e. A and B are disjoint iff A B =  Sets A1, A2,... are mutually disjoint (pairwise disjoint) iff no two sets Ai and Aj w distinct subscripts have any elems in common, i.e. for all i,j=1,2,..., i.e. Ai Aj = whenever i ≠ j | | | | | | | | | |
| Power Sets | Power set of A, P(A) is the set of all subsets of A | | | | | E.g. A = {x,y}. Then P(A) = {, {x}, {y}, {x,y}}  If A has n elems, then its power set P(A) has 2n elems | | | | | | |
| Ordered n-tuples | | Let n and let x1, x2,...,xn be (not necessarily distinct) elems. Ordered n-tuple is of the form (x1,x2,...,xn)  Order pair is an ordered 2-tuple; ordered triple is an ordered 3-tuple  Cartesian product of A1, A2, ..., An denoted A1 A2  ... An is set of all ordered n-tuples (a1,a2,...,an) where ai Ai, i = 1,...,n  i.e. A1 A2  ... An = {(a1,a2,...,an): a1 A1 a2 A2... an An}  If A is a set, then An = AA...A (n times) | | | | | | | | | | |
| Properties of Sets | | | | 1. Inclusion of Intersection: For all sets A and B, (a) A B A, (b) A B B  2. Inclusion in Union: for all sets A and B, (a) A A B, (b) B A B  3. Transitive property of Subsets: For all sets A,B and C, A B B C A C | | | | | | | | |
| Procedural Versions of Set Defn | | | | 1. a X Y (a X) (a Y)  2. a X Y (a X) (a Y)  3. a X – Y (a X) (a Y) | | | | | | | 4. a a X  5. (a,b) XY (a X) (b Y) | |
| **Theorem 6.2.1** | | | | 1. Inclusion of Intersection: For all sets A and B, (a) A B A, (b) A B B  2. Inclusion in Union: for all sets A and B, (a) A A B, (b) B A B  3. Transitive property of Subsets: For all sets A,B and C, A B B C A C | | | | | | | | |
| **Theorem 6.2.2**  Set Identities | | | | Let all set below be subsets of a universal set U   |  |  |  |  | | --- | --- | --- | --- | | 1. Commutative Laws | A B = B A  A B = B A | 7. Idempotent Laws | A A = A  A A = A | | 2. Associative Laws | (A B) C = A (B C)  (A B) C = A (B C) | 8. Universal Bound Laws | A U = U  A = | | 3. Distributive Laws | A (B C) = (A B) (A C)  A (B C) = (A B) (A C) | 9. De Morgan's Laws | =  = | | 4. Identity Laws | A = A  A U = A | 10. Absorption Laws | A (A B) = A  A (A B) = A | | 5. Complement Laws | A = U  A = | 11. Complements of U and | =  = U | | 6. Double Complement Law | (AC)C = A | 12. Set Difference Law | A\B = A | | | | | | | | | |
| **Theorem 6.2.4** | | | | An empty set is a subset of every set, i.e. A for all sets A | | | | | | | | |
| **Theorem 6.3.1** | | | | Suppose A is a finite set w n elems, then P(A) has 2n elems. i.e. |P(A)| = 2|A| | | | | | | | | |
| **T3Q8** | | | | A B iff A B = B | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Relations on Sets | Let A and B be sets. A (binary) relation from A to B is a subset of A B  Given an ordered pair(x,y) in A B, x is related to y by R, or x is R-related to y, written x R y, iff (x,y) R  x R y means (x,y) R. x [R with slash across] y means (x,y) R | | | | | | | | | | | | | | | E.g. Let A = {0,1}, B = {1,2}. Define R s.t xRy iff x < y  Then 0R1, 0R2, 1R2  R = {(0,1), (0,2), (1,2)} | | | | |
| Let A and B be sets and R be a relation from A to B  Domain of R, Dom(R) is the set{aA: aRb for some bB}  Co-domain of R, coDom(R), is the set B  Range of R, Range(R) is the set{bB: aRb for some aA} | | | | | | | | | | | | | Let A = {1,2,3}, B = {2,4,9}. Define relation R from A to B as: (x,y) AB, (x,y) R x2 = y  Dom(R) = {2,3}, coDom(R) = {2,4,9}  Range(R) = {4,9} | | | | | | |
| R from A to B can also be depicted as an arrow diagram:  - Represent elems of A as pts in 1 region and elems of B as pts in another region  - For each x A and y B, draw an arrow from x to y iff xRy  E.g. Let A = {1,2,3}, B = {1,3,5}. Define relation S as: (x,y) AB, (x,y) S x < y | | | | | | | | | | | | | | | | | | | Diagram  Description automatically generated |
| Inverse of a Relation | | | | | Let R be a relation from A to B. Define the inverse relation R-1 from B to A as: R-1 = {(y,x) BA: (x,y) R}  OR x A, y B ((y,x) R-1 (x,y) R) | | | | | | | | | | | | | | | |
| Directed Graph of a Relation | | | | | A relation on set A is a relation from A to A. (i.e. a subset of AA)  AA = A2. In general An is AA (n times)  Arrow diagram of such a relation can be modified so that it becomes a directed graph.  - Represent A only once, instead of 2 separate sets of pts, and draw an arrow from each pt of A to its related pt.  - If pt is related to itself, draw loop that extends out from the pt and goes back to it | | | | | | | | | | | | | | | |
| Composition of Relations | | | | | Let A, B and C be sets. Let R AB be a relation. Let S BC be a relation. The composition of R with S, denoted SR, is the relation from A to C s.t. x A, z C (x SR z (y B (xRy ySz)))  i.e. There is some "path" from x to z via some intermediate elem y in B in arrow diagram | | | | | | | | | | | | | | | |
| Proposition: Composition is Associative. Let A,B,C,D be sets. Let R AB, S BC and T CD be relations.  T(SR) = (TS)R = TSR | | | | | | | | | | | | | | | |
| Proposition: Inverse of a composition. Let A,B and C be sets. Let R AB and S BC be relations  (SR)-1 = R-1S-1 | | | | | | | | | | | | | | | |
| N-ary Relations and Relational Databases | | | | | | | | Given n sets A1,A2,...,An, an n-ary relation R on A1A2...An is a subset of A1A2...An. The special cases of 2-ary, 3-ary and 4-ary are called binary, ternary and quaternary relations | | | | | | | | | | | | |
| Reflexivity, Symmetry & Transitivity | | | Let R be a relation on a set A  1. R is reflexive iff x A (xRx) (arrow to itself)  2. R is symmetric iff x,y A (xRy yRx) (arrow both ways)  3. R is transitive iff x,y,z A (xRy yRz xRz) (arrow from 1 to 2, 2 to 3 and 1 to 3)  **T4Q2**: R is symmetric x,y A (xRy yRx) R = R-1 | | | | | | | | | | | | | | | | Define relation R on as: x,y (xRy 3|(x-y)) aka congruence modulo 3  R is reflexive, symmetric & transitive | |
| Asymmetry & Irreflexive | | | R is asymmetric iff x,y A (xRy y NOT Rx, i.e. (y,x) R)  Relation on A is irreflexive iff x A, (x x) | | | | | | | | | | | | | | | Asymmetric Antisymmetric (vacuously true) | | |
| Strict Partial Order | | | | | | | A relation is a strict partial order iff it is irreflexive, antisymmetric and transitive | | | | | | | | | | | | | |
| Chain | | | | | | | Let be a strict partial order on a set A.  A subset C of A is a chain iff each pair of distinct elements in C is comparable, i.e. a,b C (a ≠ b) (a b b a)  A maximal chain is a chain M s.t. t M M {t} is not a chain | | | | | | | | | | | | | |
| Transitive Closure of a Relation | | Generally, a relation fails to be transitive as it fails to contain certain ordered pairs. E.g. (1,3), (3,4) in R, then (1,4) must also be in R  Relation obtained by adding the least num of ordered pairs to ensure transitivity is called the transitive closure of the relation  Transitive Closure: Let A be a set and R a relation on A. Transitive closure of R is the relation Rt on A that satisfies:  1) Rt is transitive. 2) R Rt. 3) If S is any other transitive relation that contains R, then Rt S | | | | | | | | | | | | | | | | | | |
| Relation Induced by a Partition | | | | | A **partition** of a set A is a finite or infinite collection of nonempty, mutually disjoint subsets whose union is A  is a partition of a set A if (1) is a set of which all elems are non-empty subsets of A, i.e. ≠ S A for all S  (2) Every elem of A is in exactly 1 elem of , i.e. x A S (xS) & x A S1,S2 (xS1 xS2 S1 = S2)  OR A partition of a set A is a set of non-empty subsets of A s.t. x A !S (xS) [!: there exists a unique]  Elems of a partition are called components of the partition | | | | | | | | | | | | | | | |
| Partitions as relations | | | | We may view a partition as a "is in the same components as" relation  Given a partitionof a set A, the relation R induced by the partition is: x,y A, xRy a component S of s.t. x,yS  **Thrm 8.3.1** Let A be a set w a partition and let R be the relation induced by the partition. Then R is reflexive, symmetric, and transitive (vacuously true) | | | | | | | | | | Text, letter  Description automatically generated | | | | | | |
| Equivalence Relation | | | | | Let A be a set and R a relation on A. R is an equivalence relation iff R is reflexive, symmetric and transitive. (~ to denote equivalence relation) | | | | | | | | | | | Must prove all 3 properties to prove equivalence relation | | | | |
| Equivalence Classes of an Equivalence Relation | | | | | Suppose A is a set and ~ is an equivalence relation on A. For each aA, the equivalence class of a, denoted [a] (aka class of a), is the set of all elements xA s.t. a is ~-related to x  = {xA: a~x}, OR x A (x a~x) | | | | | | | | | | | | | | | |
| E.g. Let A = {0,1,2,3,4} and define relation R on A as: R = {(0,0), (0,4), (1,1), (1,3), (2,2), (3,1), (3,3), (4,0), (4,4)}  [0] = {0,4}, [1] = {1,3}, [2] = {2}, [3] = {1,3}, [4] = {0,4}  Since [0] = [4] and [1] = [3]. The distinct equivalence class of the relation are {0,4}, {1,3} and {2} | | | | | | | | | | | | | | | |
| **Lemma Rel.1** Equivalence Classes | | | | | | | Let ~ be an equivalence relation on a set A.  The following are equivalent for all x,y A. (i) x~y. (ii) [x] = [y]. (iii) [x] [y] ≠ | | | | | | | | | | | | | |
| **Thrm 8.3.4** | | | | Partition Induced by an Equivalence Relation: If A is a set and R is an equivalence relation on A, then the distinct equivalence classes of R form a partition of A; i.e. union of the equivalence classes is all of A, and the intersection of any 2 distinct classes is empty | | | | | | | | | | | | | | | | |
| Congruence | | | | Divisibility: Let n, d . Then d|n n = dk for some k  Congruence: Let a,b , n . Then a is congruent to b modulo n iff a-b = nk for some k . i.e. n|(a--b). ab (mod n) | | | | | | | | | | | | | | | | |
| Proposition | | | | Congruence-mod n is an equivalence relation on for every n  Note [x] = [x+n] for equivalence classes of congruence-mod-n | | | | | | | | | | | | | | | | |
| Dividing a Set by an Equivalence Relation | | | | | | Let A be a set and ~ be an equivalence relation on A. Denote by A/~ the set of all equivalence classes w.r.t ~, i.e. A/~ = {: xA} (A/~: quotient of A by ~) | | | | | | | | | | | | | | |
| **Thrm Rel.2** Equivalence classes form a partition: Let ~ be an equivalence relation on a set A. Then A/~ is a partition of A | | | | | | | | | | | | | | |
| Summary | A relation on set A is a subset of A2  If R is a relation on a set A, then we write x R y for (x,y) R  A partition of a set A is a set of non-empty subsets of A s.t. x A !S (xS)  A relation R on A is an equivalence relation if 1. reflexive: x A (xRx)  2. symmetric: x,y A (xRy yRx)  3. transitive: x,y,z A (xRy yRz xRz)  Let ~ be an equivalence relation on A. Then the set of all equivalence classes is denoted by A/~ = {: xA}, where = {yA: x~y}  Proposition: The same-component relation w.r.t a partition is an equivalence relation  Theorem Rel.2: If ~ is an equivalence relation on A, then A/~ is a partition of A | | | | | | | | | | | | | | | | | | | |
| Antisymmetry | | | | | Let R be a relation on set A. R is antisymmetric iff x,y A (xRy yRx x=y)  OR x,y A (x ≠ y) ((x,y) R) ((y,x) R) | | | | | | | | | | | | Not antisymmetric: x,y A (xRy yRx x≠y)  Not symmetric antisymmetric | | | |
| x,y , aRb a|b is antisymmetric (lecture 6 eg 19a) | | | | | | | x,y , aRb a|b is not antisymmetric (lecture 6 eg 19b) | | | | | | | | |
| Partial Order Relations | | | | | Let R be a relation on set A. Then R is a partial order relation (or partial order) iff R is reflexive, antisymmetric and transitive  A set A is a partially ordered set (poset) w.r.t a partial order relation R on A, denoted by (A, R) | | | | | | | | | | - 2 partial order relations are ≤ relation on a set of real nums & relation on a set of sets  - : general partial order and notation x y is read "x is curly less than or equal to y" | | | | | |
| Hasse Diagrams | | | | | Let be a partial order on a set A. A Hasse diagram of satisfies the following condition distinct x,y,m A:  - If x y and no m A is s.t. x m y, then x is placed below y w a line joining them, else no line joins x and y  1. Remove loops at all vertices  2. Remove arrows whose existence is implied by the transitive property  3. Remove direction indicators on the arrows | | | | | | | | | | | | | | | |
| Comparability & Compatible | | | | Suppose is a partial order relation on a set A, and a, b A  - a, b are comparable iff either a b or b a. Otherwise, a and b are noncomparable  - a, b are compatible if c A s.t. a c or b c  **T5Q10**: In all partially ordered set, any 2 comparable elements are compatible | | | | | | | | | | | | | | | | |
| Maximal / Minimal / Largest / Smallest Element | | | Let set A be partially ordered w.r.t a relation and c A  1. c is a maximal elem of A iff x A, (x c) or (x and c are not comparable). i.e. x A (c x c = x) (nothing is above c)  2. c is a minimal elem of A iff x A, (c x) or (x and c are not comparable). i.e. x A (x c c = x)  3. c is the largest elem of A iff x A (x c) (c is above everything; largest elem need to be comparable w all other elems)  4. c is the smallest elem of A iff x A (c x) | | | | | | | | | | | | | | | | | |
| Consider a partial order on a set A. | | | | | | | | A smallest elem is minimal. (Likewise, any largest elem is maximal) | | | | | | | | | |
| Linearization | | | | Let be a partial order on a set A. A linearization of is a total order on A s.t x,y A (x y x y) | | | | | Linearization of a partial order can be seen as deriving 1 total order (among other possible total orders) from that partial order | | | | | | | | | | | |
| Total Order Relations | | | If R is a partial order relation on a set A, and for any 2 elems x,y in A, either xRy or yRx, then R is a total order relation (or total order) on A. i.e. R is a total order iff R is a partial order and x,y A (xRy yRx) | | | | | | | | | | Hasse diagram of a total order is 1 single line (chain).  Linearization of a total order is the total order itself | | | | | | | |
| Kahn's Algo | | | Input: A finite set A and a partial order on A  Output: A linearization of , for all indices i,j ci cj i ≤ j | | | | | | | 1. Set A0 := A and i := 0  2. Repeat until Ai = {2.1 find a minimal elem ci of Ai w.r.t ;  2.2 set Ai+1 = Ai\{ci}; 2.3 set i := i+1} | | | | | | | | | | |
| Well-Ordered Set | | | Let be a total order on a set A. A is well-ordered iff every non-empty subset of A contains a smallest elem. i.e. S P(A), S = (x S S (x y)) | | | | | | | | | | Lecture 6 example 27  (, ≤) is well-ordered. (, ≤) is not well-ordered | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Function | A (well-defined) fn f from a set X to a set Y, denoted f:X Y, is a relation satisfying  1) x X, y Y s.t. (x,y) f. 2) x X, y1, y2 Y, ((x,y1) f (x,y2) f) y1 = y2  OR f:X Y iff x X, !y Y s.t. (x,y) f (i.e. ea elem of X map to exactly 1 elem of Y) | | | | | | | For arrow diagram, 1) every elem of X has a arrow coming out of it. 2) no elem of X has 2 arrows coming out of it that points to 2 diff elem of Y |
| (Setwise) image & preimage | Let f:X Y iff (x,y) f.  So f maps x to y OR x y OR f:x y. x is called the argument of f  f(x) is "f of x" OR output of f for input x OR value of f at x OR image of x under f. And x is a preimage of f(x) | | | | | | If A X, then let f(A) = {f(x): x A}, f(A) is the setwise image of A  If B Y, then let f-1(B) = {x X: f(x) B}, f-1(B) is the setwise preimage of B under f  f-1(B) is NOT an inverse fn. (elem in B might not have preimage) | |
| Domain, co-domain, range | | | Let f:X Y be a fn from set X to set Y  X is domain of f and Y is co-domain of f. | | | Range of f is the (setwise) image of X under f. i.e {y Y: y = f(x) for some x X}  Range co-domain | | |
| Sequence & String | Seq a­0­, a1, ... can be represented by a fn a whose domain is that satisfies a(n) = an n  Fibonacci seq F0, F1,... defined n , F0 = 0, F1 = 1 and Fn+2 = Fn+1 + Fn | | | | | | | |
| Let A be a set. A string/word over A is of the form a0a1...al-1 where l and a0,a1,...,al-1 A  l is aka length of string. Empty string is string of length 0  Let A\* denote set of all strings over A | | | | | | | |
| Equality of Seq: Given 2 seq, defined by fn a(n) = an and b(n) = bn n , 2 seq are equal iff a(n) = b(n) n  Equality of Strings: Given 2 strings s1 = a0a1…al-1 and s2 = b0b1…bl-1 where l , s1 = s2 iff ai = bi n {0,1,2,...,l-1} | | | | | | | |
| Function equality | | | | **Thrm 7.1.1** Fn Equality: 2 fn f: A B and g: C D are equal, f = g iff 1) A=C and B=D, and 2) f(x) = g(x) x A | | | | |
| Injections (One-to-One fn) | | | Fn f: X Y is injective (one-to-one) iff x1,x2 A (f(x1) = f(x2) x1 = x2) OR x1 ≠ x2 f(x1) ≠ f(x2) (contrapositive)  An injective fn is called an injection. (every elem in codomain has ≤1 arrow gg to it) | | | | | |
| Surjections (Onto fn) | | | Fn f: X Y is surjective (onto) iff y Y x X (y=f(x)), i.e. every elem in co-domain has a preimage. So range = co-domain  A surjective fn is called a surjection. (every elem in codomain has ≥1 arrow gg to it) | | | | | |
| Bijection (One-to-One correspondences) | | | | | Fn f: X Y is bijective iff f is injective and surjective, i.e. y Y !x X (y=f(x)).  A bijective fn is called a bijection/one-to-one correspondence. (every elem in codomain has exactly 1 arrow gg to it) | | | |
| Inverse Fn | Let f: X Y. Then g: Y X is an inverse of f iff X y Y (y=f(x) x=g(y))  Proposition: Uniqueness of inverses: If g1 and g2 are inverses of f: X Y, then g1 = g2  **Thrm 7.2.3**: If f: X Y is a bijection, then f-1: Y X is also a bijection. i.e. f: X Y is bijective iff f has an inverse | | | | | | | |
| Composition of Fns | | Let f: X Y and g: Y Z be fns. Define a new fn gf: X Z as (gf)(x) = g(f(x)) X  gf is the composition of f and g (g circle f/g of f of x) | | | | | | |
| Identity Fn | Identity fn on set X, idX, if the fn from X to X defined as idX(x) = x X  **Thrm 7.3.1** Composition w an Identity Fn: If f is a fn from set X to set Y, and idX is the identity fn on X, and idY is the identity fn on Y, then fidX = f and idYf = f  **Thrm 7.3.2** Composition of Fn w its Inverse: If f: X Y is a bijection w inverse fn f-1: Y X, then f-1f = idX and ff-1 = idY | | | | | | | |
| Properties | Thrm Associativity of Fn Composition: Let f: A B, g: B C and h: C D. Then (hg)f = h(gf)  Fn composition is NONcommutative: (gf)(x) ≠ (fg)(x)  **Thrm 7.3.3**: If f: X Y and g: Y Z are both injective, then gf is injective  **Thrm 7.3.4**: If f: X Y and g: Y Z are both surjective, then gf is surjective  **T6Q6**: If gf is injective, f: X Y and g has domain Y, then f is injective  **T6Q7**: If fg is surjective, f: X Y and g has codomain X, then f is surjective | | | | | | | |
|  | Quotient where is the congruence-mod-n relation on , is denoted  E.g. = {{2k: k }, {2k+1: k }}  Define addition + and multiplication on as: whenever [x], [y] , [x] + [y] = [x+y] and [x] [y] = [x\*y]  Proposition: Addition on is well defined. For all n and all [x1],[y1],[x2],[y2] , [x1] = [x2] and [y1] = [y2] [x1] + [y1] = [x2] + [y2]  Proposition: Multiplication on is well defined. For all n and all [x1],[y1],[x2],[y2] , [x1] = [x2] and [y1] = [y2] [x1] [y1] = [x2] [y2] | | | | | | | |
| Order of Bijection | | | | The order of a bijection f: X X is defined to be the least n s.t. ff...f = idA (n times of f) | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sequences | Seq is an ordered set w members called terms. General form: am, am+1, ..., an where m ≤ n  Explicit form: ak = f(k) where f is some fn  Summation: . Expanded form of sum: am + am+1 + ... + an  Summation using recursion: = + an, for all integers n > m | | | | | | | | | | By convention if m > n, then summation = 0  Telescoping sums: convert to partial fractions  If m > n, = 1 | | |
| Common seq | Arithmetic seq/progression iff there is a constant d s.t. ak = ak-1 + d, k . So an = a0 + nd, ints n ≥ 0  Geometric seq/progression iff there is a constant r s.t. ak = rak-1, k . So an = a0rn ints n ≥ 0 | | | | | | | | | | | | Note fn on right is known as closed form |
| Triangle nums: 1,3,6,10,15,21,28,... | | | | Fibonacci nums: 1,1,2,3,5,8,13,21,34,55,... | | | | | | | |
| Lazy Caterer's Seq: 1,2,4,7,11,16,... | | | | Catalan's nums: 1,1,2,5,14,42; | | | | | | | |
| Mathe-matical Induction | Principle of (weak/regular) Mathematical Induction (PMI/1PI) 1. n , let P(n) ...  2. Basis step: Show P(a) is true  3. Inductive step:  3.1. Let k s.t. P(k) is true, i.e.  3.2. Show P(k+1) true  5. n , P(n) true by MI | | | | | Strong Induction (2PI)  1. n , let P(n) ...  2. Basis step: Show P(a) is true, P(a+1),... P(b) true  3. Inductive step:  3.1. Let k s.t. P(a), P(a+1),... P(k + b-a) is true  3.2. Show P(k+1) true  5. n , P(n) true by Strong MI | | | | | | | |
| Well-ordering Principle | | | | Well-Ordering Principle for Integers: Every nonempty subset of has a smallest element. | | | | | | | | | |
| Recursively Defined Sets | | Base clause: Specify certain elements, called founders are in S  Recursion clause: Specify certain constructors under which set S is closed  Minimality clause: Membership for S can always be demonstrated by (infinitely many) successive applications of the clauses above | | | | | | | | E.g. () is in P  a. If E is in P, so is (E); b. If E and F in P, so is EF | | | |
| Structural Induction | To prove x S P(x) is true, suffices to  (basis step) show P(c) is true for every founder c; and  (induction step) show x S (P(x) P(f(x)) is true for every constructor f | | | | | | | x S by base clause  y S by recursion clause w n = ... | | | | | |
| **Thrm 5.1.1** | | | If am, am+1, ... and bm, bm+1,... are seq of real nums and c is any real num, then for any int n ≥ m:  1.  2. c = (generalized distributive law)  3. | | | | | | | | | | |
| **Theorem 5.2.2** | | | Sum of 1st n ints: For all ints n ≥ 1, 1 + 2 + 3 + ... + n = | | | | | | Some fact from tutorial: Product of any 2 consecutive integers is even | | | | |
| **Theorem 5.2.3** | | | Sum of GP: For any real num r ≠ 1, and any int n ≥ 0, | | | | | |
| **Proposition 5.3.1** | | | For all ints n ≥ 0, 22n - 1 is divisible by 3 | | | | **Proposition 5.3.2** | | | | | For all ints n ≥ 3, 2n + 1 < 2n | |
| **T7Q1** | | |  | | | | **T7Q2** | | | | | Let x . n , 1 + nx ≤ | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pigeonhole Principle | | | Let A and B be finite sets. If there is an injection f: A B, then |A| ≤ |B| | | | | | | | |
| Dual Pigeonhole Principle | | | | Let A and B be finite sets. If there is a surjection f: A B, then |A| ≥ |B| | | | | | | |
| Cardinality | Let = {1,2,3,...,n}. Set S is finite iff S is empty or a bijection from S to for some n  Cardinality of a finite set S, |S| is | | | | | | | | | A set S is infinite if it is not finite |
| Theorem: Equality of Cardinality of Finite Sets | | | | Let A and B be any finite sets. |A| = |B| iff there is a bijection f: A B | | | | | |
| Same Cardinality (Cantor): Given any 2 sets A and B. A have same cardinality as B, |A| = |B| iff there is a bijection f: A B | | | | | | | | | |
| **Thrm 7.4.1** Properties of Cardinality: The cardinality is an equivalence relation  For all sets A,B and C: | | | | | | Reflexive: |A| = |A|. Symmetric: |A| = |B| |B| = |A|  Transitive: (|A| = |B|) (|B| = |C|) |A| = |C| | | | |
| Countably Infinite | Cardinal numbers: Define ("aleph"; 1st cardinal number)  A set S is countably infinite (or S has the cardinality of natural numbers) iff |S| = | | | | | | | A set is countable iff it is finite or countably infinite  A set is uncountable if it is not countable | | |
| Eg | is countable. Let f(n): =  is countable. Set F(1) = , F(2) = , F(3) = , F(4) = , Then skip since counted, F(5) =  Every positive rational num appears somewhere in grid, and counting procedure is so every point in grid is reached eventually. Thus F is surjective  Skipping numbers that have already been counted ensures no num is counted twice. F is injective. So F is a bijection from to . So is countably infinite and countable.  Thrm: is countable. Set f: : f(x,y) = | | | | | | | | | A picture containing scatter chart  Description automatically generated |
| Theorems | Cartesian Product: If sets A and B are both countably infinite, then so is  Corollary (General Cartesian Product): Given n ≥ 2 countably infinite sets A1, A2,...,An, is also countably infinite  Thrm (Unions): The union of countably many countable sets is countable. i.e. if A1, A2, ... are all countable sets, then so is | | | | | | | | | |
| Countability via Sequences | | **Proposition 9.1** An infinite set B is countable iff there is a seq b0,b1,... B in which every element of B appears exactly once  **Lemma 9.2** (Countability via Sequence): An infinite set B is countable iff there is a seq b0,b1,... B in which every element of B appears | | | | | | | | |
| Larger Infinities | **Thrm 7.4.2 (Cantor):** The set of real numbers btw 0 and 1 is uncountable  Cantor's Diagonalization Argument (Proof by contradiction):  1. Suppose (0,1) is countable  2. Since it is not finite, it is countably infinite  3. List the elems xi of (0,1) in a seq as follows: x1 = 0.a11a12a13...a1n...,  x2 = 0.a21a22a23...a2n...,  x3 = 0.a31a32a33...a3n..., | | | | | xn = 0.an1an2an3...ann...,    4. Construct a num d = 0.d1d2d3...dn... s.t. dn =  5. Note that , dn ≠ ann. Thus d ≠ xn  6. But d (0,1), hence contradiction. Thus (0,1) uncountable | | | | |
| Thrms | **Thrm 7.4.3:** Any subset of any countable set is countable  **Corollary 7.4.4**: Any set w an uncountable subset is uncountable. Since (0,1) , is uncountable.  **Proposition 9.3**: Every infinite set has a countably infinite subset  **Lemma 9.4** (Union of Countably Infinite Sets): Let A and B be countably infinite sets. Then A B is countable | | | | | | | | | |
| Cardinality of | = |(0,1)|. Let S = (0,1). Imagine picking up S and bending it into a circle  Define F: S as follows: Draw a number line and place S bent into a circle, tangent to the line above point 0  For each point x on the circle representing S, draw a straight line L through the topmost point of the circle and x  Let F(x) be the pt of intersection of L and the number line  Can be seen that F(x) is injective and surjective. Hence S and have same cardinality | | | | | | | | A picture containing shape  Description automatically generated | |
| **T8Q2** | Let B be a countably infinite set and C be a finite set, then B C is countable | | | | | | | | | |
| **T8Q4** | Suppose A1,A2,... are countable sets. Then is countable for any n | | | | | | | | | |
| **T8Q7** | Set B is infinite iff A B s.t. |A| = |B| | | | | | | | | | |
| **T8Q9** | Let A be a countably infinite set. Then (A) is uncountable | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Probability & Counting | | | | Equally Likely Probability Formula: If S is a finite sample space where all outcomes are equally likely and E is an event in S, then the probability of E, P(E) = | | | | | | | |
| **Thrm 9.1.1** Number of elements in a list: If m and n are integers and m ≤ n, then there are n-m+1 integers from m to n inclusive  E.g. how many ints divisible by 5 from 100 to 999: 100 = 5\*20, 995 = 5\*199. So 199-20+1 = 180 such ints | | | | | | | |
| Product/ multiplication rule | | | **Thrm 9.2.1** (Multiplication/Product Rule): If an operation consists of k steps and 1st step can be performed in n1 ways, 2nd step in n2 ways, ... kth step in nk ways (regardless of preceding steps), then entire op can be performed in n1\*n2\*...\*nk ways  **Thrm 5.2.4** (Sets): Suppose A is a finite set. Then | | | | | | | | |
| Addition/ sum rule | | | **Thrm 9.3.1** (Addition/Sum rule): Suppose a finite set A equals the union of k distinct mutually disjoint subsets A1, A2, ..., Ak. Then |A| = |A1| + |A2| + ... + |Ak| | | | | | | | | |
| Permutation | | | | Permutation is an ordering of the objects in a row.  **Thrm 9.2.2** (Permutations): The num of permutations of a set with n (n ≥ 1) elements is n!  **Thrm 9.2.3** (r-permutations from a set of n elements): If n and r are ints and 1 ≤ r ≤ n, then the num of r-permtations of a set of n elems is P(n,r) = n(n-1)(n-2)...(n-r+1) = | | | | | | | |
| Difference rule | **Thrm 9.3.2** (Difference Rule): If A is a finite set and B A, then |A\B| = |A| – |B|  Formula for Probability of the Complement of an Event: If S is a finite sample space and A is an event in S, then P() = 1 – P(A) | | | | | | | | | | |
| Inclusion/ Exclusion Rule | | | | **Thrm 9.3.3** (Inclusion/Exclusion Rule for 2 or 3 sets): If A, B, and C are any finite sets, then |AB| = |A| + |B| – |AB|  and |ABC| = |A| + |B| + |C| – |AB| – |AC| – |BC| + |ABC| | | | | | | | |
| Pigeonhole Principle (PHP) | A function from one finite set to a smaller finite set cannot be one-to-one (injective): There must be at least 2 elements in the domains that have the same image in co-domain | | | | | | | | | | |
| Application to Decimal Expansions of Fractions: Decimal expansion of any rational num either terminates or repeats | | | | | | | | | | |
| **Generalised Pigeonhole Principle**: For any fn f from a finite set X w n elems to a finite set Y w m elems and for any positive int k, if k < n/m, there there is some y Y s.t. y is the image of at least k+1 distinct elems of X | | | | | | | | | | |
| Generalized Pigeonhole Principle (**Contrapositive Form**): For any fn f from a finite set X w n elems to a finite set Y w m elems and for any positive int k, if for each y Y, f-1({y}) has at most k elems, then X has at most km elems; i.e. n ≤ km | | | | | | | | | | |
| Combinations | | | | Let n and r be nonnegative ints w r ≤ n. An r-combination of a set n elems is a subset of r of the n elem. denotes num of subsets of size r that can be chosen from set of n elems  **Thrm 9.5.1** Formula for : , where n and r are nonnegative ints w r ≤ n | | | | | | | r- permutation: Ordered selection  r-combination: unordered selection  P(n,r) = \* r! |
| Repetitions allowed | | **Thrm 9.5.2** Permutations w Sets of Indistinguishable Objs: Suppose a collection consists of n objs of which n1 are of type 1 and are indistinguishable from ea other, n2 are of type 2 and are indistinguishable from ea other,..., nk are of type k and are indistinguishable from ea other and suppose n1 + n2 + ... + nk = n. Then num of distinguishable permutations of the n objs is | | | | | | | | | |
| An r-combination w repetition allowed OR multiset of size r, chosen from a set X of n elems is an unordered selection of elems taken from X w repetition allowed. Note objects are indistinguishable  If X = {x1,x2,...,xn}, multiset of size r is [] where ea is in X and some of the may equal ea other  Thrm 9.6.1 Num of r-combinations w repetition: Num of multisets of size r that can be selected from a set of n elems is  Num of soln to x1 + x2 + ... + xn = r, xi is nonnegative int:  Num of soln to x1 + x2 + x3 = 20, xi is a positive int: equivalent to y1 + y2 + y3 = 17: | | | | | | | | | |
| Summary | |  |  |  | | --- | --- | --- | |  | Order Matters | Order don't matter | | Repetition | nk |  | | No Repetition | P(n,k) |  | | | | | | | | | | Circular Permutation of n objects is (n-1)! | |
| Pascal's Formula | **Thrm 9.7.1** Pascal's Formula: Let n and r be positive ints, r ≤ n. Then  Combinatorial Proof uses counting as basis of proof. Includes bijective proof and proof by double counting (counting num of elems in 2 diff ways to obtain diff expressions in identity)  For 0 ≤ k ≤ n, (don't choose n-r ppl)  For 0 ≤ k ≤ n, k = n (choose k committee & chairperson)  2n = (num of subsets of power set) | | | | | | | | | | |
| Binomial Theorem | | | | | | **Thrm 9.7.2** Binomial Thrm: Given any real nums a and b and any non-negative int n, (a+b)n = | | | | | |
| Probability Axioms | Let S be a sample space. A probability fn P from set of all events in S to set of real nums satisfies the following axioms: For all events A and B in S, | | | | | | | 1. 0 ≤ P(A) ≤ 1  2. P() = 0 and P(S) = 1  3. If A and B are disjoint events (AB=), then P(AB) = P(A) + P(B) | | | |
| Formula | Probability of Complement: If A is any event in sample space S, then P() = 1 – P(A)  Probability of General Union of 2 events: If A and B are any events in sample space S, then P(AB) = P(A) + P(B) – P(AB)  Suppose possible outcomes of an experiment are real nums a1,a2,...,an w prob p1,p2,...,pn. The expected value is  Linearity of Expectation: E | | | | | | | | | | |
| Conditional Probability | | | | | | | If P(A) ≠ 0, then conditional prob of B given A, P(B|A) = OR = P(B|A)\*P(A) OR P(A) = | | | | |
| Bayes Theorem | **Thrm 9.9.1** Bayes Thrm: Suppose sample space S is a union of mutually disjoint events B1, B2, ..., Bn  Suppose A is an event in S, and suppose A and all the Bi have non-zero prob. If k is an int w 1 ≤ k ≤ n, then  P(Bk|A) = | | | | | | | | | | |
| Independent Events | | | | | If A and B are events in sample space S, then A and B are indep iff P = P(A) \* P(B) | | | | | | |
| Pairwise independent/ Mutually Independent | | | Let A,B and C be events in sample space S. A,B and C are pairwise indep, iff they satisfy conditions 1–3. They are mutually independent iff they satisfy all 4 conditions | | | | | | | | |
| 1. P = P(A)\*P(B) | | | | | | 3. P = P(B) \* P(C) | | |
| 2. P = P(A)\*P(C) | | | | | | 4. P = P(A)\*P(B)\*P(C) | | |
| Events A1,A2,...,An in sample space S are mutually indep iff probability of intersection of any subsets of events is the product of probabilities of the events in the subset | | | | | | | | |
| Binomial Dist | | | X~Binomial(n,p). P(X=x) = | | | | | | |  | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Undirected Graph | | | Undirected graph is denoted by G = (V,E), where V= {v1,...,vn} is set of vertices/nodes in G and E = {e1,...,ek} is set of (undirected) edges in G  An (undirected) edge e connecting vi and vj is denoted as e = {vi,vj} (i can = j)  An undirected graph G consists of 2 finite sets: a nonepty set V of vertices and a set E of edges, where each (undirected) edge is associated w a set consisting of either 1 or 2 vertices called its endpoints  An edge connect its endpoints; vertext that is an endpoint of a loop is aka adjacent to itself; and 2 vertices connected by an edge aka adjacent vertices  An edge is incident on each of its endpoints, and 2 edges incident on the same endpoint are called adjacent edges | | | | |
| Directed Graph | | | Directed graph/digraph G consists of 2 finites sets: a nonempty set V of vertices and a set E of directed edges, where each (directed) edge is associated with an ordered pair of vertices called its endpoints  e = (v,w) for a directed edge e from vertex v to vertex w | | | | |
| Simple Graph | | | Simple graph is an undirected graph that does not have any loops (edge to itself) or parallel edges (both with same set of vertices). (i.e. there is at most 1 edge btw each pair of distinct vertices) | | | | |
| Complete Graph | | A complete graph on n vertices, n > 0, denoted Kn is a simple graph w n vertices and exactly 1 edge connecting each pair of distinct vertices  Number of edges = | | | | | |
| Bipartite Graph | | | Bipartite graph/bigraph is a simple graph whose vertices can be divided into 2 disjoint sets U and V s.t. every edge connects a vertex in U to 1 in V  Complete Bipartite graph is a bipartite graph on 2 disjoint set U and V s.t. every vertex in U connects to every vertex in V. If |U| = m and |V| = n, the complete bigraph is denoted as Km,n | | | Diagram  Description automatically generatedDiagram  Description automatically generatedDiagram  Description automatically generated Complete bipartite | |
| Subgraph of Graph | | | A graph H is a subgraph of graph G iff every vertex in H is also a vertex in G and every edge in H is also an edge in G, and every edge in H has the same endpoints as it has in G | | | | |
| Degree of a Vertex | | | Let G be a undirected graph and v a vertex of G. The degree of v, deg(v) = num of edges that are incident on v, w an edge that is a loop counted twice  The total degree of G = sum of degress of all vertices of G | | | Diagram  Description automatically generated | |
| Thrm for undirected graph? | | | **Thrm 10.1.1** Handshake Theorem: If G is any graph, then the sum of degrees of all vertices = 2\*num of edges of G  **Corollary 10.1.2**: The total degree of a graph is even  **Proposition 10.1.3**: In any graph, there are an even num of vertices of odd degree | | | | |
| Indegree & Outdegree | | | Let G = (V,E) be a directed graph and v a vertex of G. The indegree of v, deg–(v) is num of directed edges that end at v. The outdegree of v, deg+(v) = num of directed edges that originate from v  Note | | | | |
| Travel in a graph | Let G be a graph and v and w be vertices of G.  A **walk** from v to w is a finite alternating sequence of adjacent vertices and edges of G.  A walk has the form v0e1v1e2,...,vn-1envn, where v's are vertices, e are edges, v0 = v, vn = w, and i {1,2,...,n}, vi-1 and vi are the endpoints of ei. The num of edges, n, is the **length** of the walk  The **trivial walk** from v to v consists of the single vertex v  A **trail** from v to w is a walk from v to w that does not contain a repeated edge  A **path** from v to w is a trail that does not contain a repeated vertex  A **closed walk** is a walk that starts and ends at the same vertex  A **circuit/cycle** is a closed walk of length at least 3 that does not contain a repeated edge  A **simple circuit/cycle** is a circuit that does not have any other repeated vertex except the first and last  An undirected grph is **cyclic** if it contains a loop or a cycle; otherwise it is acyclic | | | | | | A picture containing shape  Description automatically generated  u1e1u2e3u5e4u3e5u6e7u5e3u2 is a walk (may repeat edges and/or vertices)  u1e1u2e3u5e4u3e5u6e7u5e6u4 is a trail (cannot repeat edges)  u1e1u2e3u5e4u3e5u6 is a path (cannot repeat vertices and edges)  u5e6u4e2u1e1u2e3u5e7u6e5u3e4u5 is a circuit  u5e6u4e2u1e1u2e3u5 is a simple circuit |
| Connected-ness | | | 2 vertices v and w of graph G = (V,E) are connected iff there is a walk from v to w  The graph G is connected iff vertices v,w V, a walk from v to w  **Lemma 10.2.1** Let G be a graph  a) If G is connected, then any 2 distinct vertices of G can be connected by a path  b) If vertices v and w are part of a circuit in G and 1 edge is removed from the circuit, then there still exists a trail from v to w in g  c) If G is connected and G contains a circuit, then an edge of the circuit can be removed w/o disconnecting G | | | | |
| Connected Component | | | A graph H is a connected component of a graph G iff  1. The graph H is a subgraph of G  2. The graph H is connected  3. No connected subgraph of G has H as a subgraph and contains vertices or edges that are not in H ("largest size/can be expanded") | | | | |
| Euler Circuits | | | Let G be a graph. An Euler circuit for G is a circuit that contains every vertex and traverses every edge of G exactly once  An Eulerian graph is a graph that contains an Euler circuit  **Thrm 10.2.2**: If a graph has an Euler circuit, then every vertex of the graph has positive even degree  Contrapositive of Thrm 10.2.2: if some vertex of a graph has odd degree, then the graph does not have an Euler circuit  **Thrm 10.2.3**: If a graph G is connected and the degree of every vertex of G is a positive even integer, then G has an Euler circuit  **Thrm 10.2.4**: A graph G has an Euler circuit iff G is connected and every vertex of G has positive even degree  **Euler Trail**: Let G be a graph, and v and w be 2 distinct vertices of G. An Euler trail/path from v to w is a seq of adjacent edges and vertices that starts at v, ends at w, passes through every vertex of G at least once, and traverses every edge of G exactly once  **Corollary 10.2.5**: Let G be a graph, and v and w be 2 distinct vertices of G. There is an Euler trail from v to w iff G is connected, v and w have odd degree, and all other vertices of G have positive even degree | | | | |
| Hamiltonian Circuits | | | Given a graph G, a Hamiltonian circuit for G is a simple circuit that includes every vertex of G (i.e. every vertex appears exactly once, except for the first and last which are the same)  A Hamiltonian/Hamilton graph is a graph that contains a Hamiltonian circuit  Euler circuit can visit vertices more than once. Hamiltonian circuit does not need to include all edges  **Proposition 10.2.6**: If a graph G has a Hamiltonian circuit, then G has a subgraph H w the following properties:  1. H contains every vertex of G. 2. H is connected. 3. H has the same num of edges as vertices. 4. Every vertex of H has deg 2  Contrapositive of 10.2.6 says if a graph G does not have a subgraph H w properties (1)-(4), then G does not have a Hamiltonian circuit | | | | |
| Matrix | | | **A**m x n = (aij). **A** = **B** iff A and B have the same size, and aij = bij i = 1,2,...,m and j = 1,2,...,n  Square matrix: matrix w same num of rows and cols  If A is a sq matrix of size n x n, then main diagonal of A consists of entries a11,a22,...,ann  Let G be a directed graph w ordered vertices v1,...,vn. The adjacency matrix of G is the n x n matrix **A** = (aij) over the set of non-negative ints s.t. aij = num of arrows from vi to vj i,j = 1,2,...,n  Let G be an undirected graph w ordered vertices v1,...,vn. The adjacency matrix of G is the n x n matrix **A** = (aij) over the set of non-negative ints s.t. aij = num of edges connecting vi and vj i,j = 1,2,...,n  Note adjacency matrix for undirected graph is symmetric (i.e. aij = aji) | | | | |
| Let A = (aij) be m x k matrix, B = (bij) be k x n matrix. AB is the matrix (cij) where cij = i = 1,...,m and j = 1,...,n  Note matrix multiplication is NOT commutative. Matrix multiplication is associative  For each positive int n, the n x n identity matrix, denoted In = () or just **I**, where , i,j = 1,2,...n  For any n x n matrix A, the powers of A are: A0 = I. An = AAn-1 | | | | |
| **Thrm 10.3.2**: If G is a graph w vertices v1,...vm and **A** is the adjacency matrix of G, then for each positive int n and ints i,j = 1,2,...,m, the ijth entry of **A**n = num of walks of length n from vi to vj | | | | |
| Isomor-phism | Let G = (VG, EG) and G' = be 2 graphs. G is isomorphic to G', G G', iff there exists bijections g: VG and h: EG that preserve the edge-endpoint functions of G and G' in the sense that v VG and e EG, v is an endpoint of e g(v) is an endpoint of h(e)  OR G is isomorphic to G' iff there exists a permutation π: VG s.t. {u,v} EG {π(u),π(v)}  **Thrm 10.4.1** Graph Isomorphism is an Equivalence Relation: Let S be a set of graphs and let be the relation of graph isomorphism on S. Then is an equivalence relation on S | | | | | | |
| Planar Graphs | A planar graph is a graph that can be drawn on a (2D) plane w/o edges crossing  Kuratowski's Thrm: A finite graph is planar iff it does not contain a subgraph that is a subdivision of the complete graph K5 or the complete bipartite graph K3,3  Euler's Formula: For a connected planar simple graph G = (V,E) w e = |E| and v = |V|, if we let f be number of faces/regions, then f = e - v + 2 | | | | A picture containing text, clock, watch  Description automatically generatedGraphical user interface, text, application  Description automatically generatedDiagram, shape, polygon  Description automatically generatedDiagram  Description automatically generated  e = 8, v = 6, f = 8-6+2 = 4 | | |
| Complement Graph | | | | If 𝐺 is a simple graph, the complement of 𝐺, denoted , is obtained as follows: the vertex set of is identical to the vertex set of 𝐺. However, two distinct vertices 𝑣 and 𝑤 of are connected by an edge if and only if 𝑣 and 𝑤 are not connected by an edge in 𝐺. | | | |
| A self-complementary graph is isomorphic w its complement | | | |
| **Lemma 10.5.5** | | | | Let 𝐺 be a simple, undirected graph. Then if there are two distinct paths from a vertex 𝑣 to a different vertex 𝑤, then 𝐺 contains a cycle (and hence 𝐺 is cyclic). | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tree | (graph is assumed to be undirected)  Graph is **circuit-free** iff it has no circuit  Graph is a **tree** iff it is circuit-free and connected  **Trivial tree** is a graph that has only a single vertex | | | | Graph is a **forest** iff circuit free and not connected  Let T be a tree. If T has only 1 or 2 vertices, then each is called a **terminal vertex/leaf**. If T has at least 3 vertices, then a vertex of deg 1 in t is a terminal vertex/leaf, and a vertex of deg > 1 in T is an internal vertex | |
| **Lemma 10.5.1**: Any non-trivial tree has at least 1 vertex of deg 1  **Thrm 10.5.2**: Any tree w n vertices (n > 0) has n - 1 edges (proof by MI)  A non-trivial tree has at least 2 vertices of deg 1  **Lemma 10.5.3**: If G is any connected graph, C is any circuit in G, and one of the edges of C is removed from G, then the graph that remains is still connected.  **Thrm 10.5.4**: If G is a connected graph w n vertices and n-1 edges, then G is a tree | | | | | Proof 10.5.1. Let T be a arbitrarily chosen non-trivial tree  1. Pick a vertex v of T and let e be an edge incident on v  2. While deg(v) > 1, repeat step 2a, 2b and 2c:  2a. Choose e' to be an edge incident on v s.t. e' ≠ e  2b. Let v' be the vertex at other end of e' from v  2c. Let e = e' and v = v'  Algo must eventually terminate as set of vertices of T is finite and T is circuit-free. When it does, a vertex of deg 1 is found |
| Rooted Tree | | | **Rooted Tree** is a tree in which there is one vertex that is distinguished from the others and called the root  **Level** of a vertex is num of edges along the unique path btw it and the root  **Height** of a rooted tree is the max level of any vertex of the tree  Given the root or any internal vertex v of a rooted tree, the **children** of v are all those vertices that are adjacent to v and are one level farther away from the root than v  If w is a child of v, then v is the **parent** of w, and 2 distinct vertices that are both children of the same parent are called **siblings**  Given 2 distinct vertices v and w, if v lies on the unique path btw w and the root, then v is an **ancestor** of w, and w is a **descendant** of v | | | |
| Binary Trees | | | A **binary tree** is a rooted tree in which every parent has at most 2 children. Each child is designated either a left child or a right child (but not both), and every parent has at most 1 left child and 1 right child.  A **full binary tree** is a binary tree in which each parent has exactly 2 children  Given any parent v in a binary tree T, if v has a left child, the the **left subtree** of v is the binary tree whose root is the left child of v, whoses vertices consist of the left child of v and all its descendants, and whose edges consist of all those edges of T that connect the vertices of the left subtree. The **right subtree** of v is defined analogously. | | | |
| **Thrm 10.6.1**: Full Binary Tree Thrm: If T is a full binary tree w k internal vertices, then T has a total of 2k+1 vertices and has k+1 terminal vertices (leaves)  **Thrm 10.6.2**: For non-negative integers h, if T is any binary tree w height h and t terminal vertices (leaves), then t ≤ 2h or equivalently, h ≤ log2t (proof by strong MI) | | | |
| Tree Search | | | Tree search/traversal is process of visiting each node in a tree data structure exactly once in a systematic manner | | | |
| BFS: Starts at root -> visits adjacent vertices -> move to next level | | | |
| |  |  | | --- | --- | | Pre-order | Print current vertex -> Traverse left subtree by recursively calling the pre-order fn -> Traverse right subtree by recursively calling the pre-order fn | | In-order | Traverse left subtree by recursively calling the in-order fn -> Print current vertex -> Traverse right subtree by recursively calling the in-order fn | | Post-order | Traverse left subtree by recursively calling the post-order fn -> Traverse right subtree by recursively calling the post-order fn -> Print current vertex |   DFS: | | | |
| Spanning Tree | | | A **spanning tree**  for a graph G is a subgraph of G that contains every vertex of G and is a tree  **Proposition 10.7.1**: 1. Every connected graph has a spanning tree. 2. Any 2 spanning trees for a graph have the same num of edges | | | |
| A **weighted graph** is a graph for which each edge has an associated positive real num **weight**. The sum of the weights of all the edges is the **total weight** of the graph  A **minimum spanning tree** for a connected weighted graph is a spanning tree that has the least possible total weight compared to all other spanning trees for the graph.  If G is a weighted graph and e is an edge of G, then **w(e)** denotes weight of e and **w(G)** denotes the total weight of G | | | |
| Algos | | Algo 10.7.1 Kruskal  Input: G [a connected weighted graph w n vertices]  1. Initialise T to have all vertices of G and no edges  2. Let E be the set of all edges of G, and let m = 0  3. While (m < n-1):  3a. Find an edge e in E of least weight  3b. Delete e from E  3c. If addition of e to the edge set of T does not produce a  circuit, then add e to edge set of T and set m = m+1  Output: T [T is a MST for G] | | Algo 10.7.2 Prim  Input: G [a connected weighted graph w n vertices]  1. Pick a vertex v of G and let T be the graph w this vertex only  2. Let V be the set of all vertices of G except v  3. For i = 1 to n-1:  3a. Find an edge e of G s.t. (1) e connects T to one of the vertices in V, and  (2) e has the least weight of all edges connecting T to a vertex in V. Let w  be the endpoint of e that is in V  3b. Add e and w to the edge and vertex sets of T, and delete w from V  Output: T [T is a MST for G] | | |