CS1231 DISCRETE STRUCTURES

Cheatsheet

Properties of Tian Xiao

Logical Statements

Basic Operators: and (∧), or (∨), not (~)

Laws of Logical Equivalence

01. Commutative Laws: $p \wedge\!\!\!\!\vee q \equiv q \wedge\!\!\!\!\vee p$
02. Associative Laws: $(p \wedge\!\!\!\!\vee q) \wedge\!\!\!\!\vee r \equiv p \wedge\!\!\!\!\vee (q \wedge\!\!\!\!\vee r)$
03. Distribution Laws: $p \wedge (q \vee r)$; $p \vee (q \wedge r)$
04. Identity Laws: $p \wedge$ **True** $\equiv p$; $p \vee$ **False** $\equiv p$
05. Negation Laws: $p \wedge \sim p \equiv$ **False;** $p \vee \sim p \equiv$ **True**
06. Double Negative Laws: $\sim(\sim p) \equiv p$
07. Idempotent Laws: $p \wedge p \equiv p$; $p \vee p \equiv p$
08. Universal Bound Laws: $p \wedge$ **False** $\equiv$ **False**; $p \vee$ **True** $\equiv$ **True**
09. De Morgan's Laws: $\sim(p \wedge q) \equiv \sim p \vee \sim q$; $\sim(p \vee q) \equiv \sim p \wedge \sim q$
10. Absorption Laws: $p \vee (p \wedge q) \equiv p$; $p \wedge (p \vee q) \equiv p$
11. Negation of True/False: ~**True** ≡ **False**; ~**False** ≡ **True**

Conditional Statements

01. Implication Law: $p \to q \equiv \sim p \vee q$
02. Contrapositive: $\sim q \to \sim p \equiv p \to q$
03. Converse: $q \to p$
04. Inverse: $\sim p \to \sim q$

Rules of Inference

Modus Ponens

$p \to q$
p
$\bullet\ q$

Modus Tollens

$p \to q$
$\sim q$
$\bullet \sim p$

Generalisation

p
$\bullet\ p \vee q$

Conjunction

p
q
$\bullet\ p \wedge q$

Elimination

$p \vee q$
$\sim q$
$\bullet\ p$

Specialisation

$p \wedge q$
$\bullet\ p$

Transitivity

$p \to q$
$q \to r$
$\bullet\ p \to r$

Division into Cases

$p \vee q$
$p \to r$
$q \to r$
$\bullet\ r$

Contradiction

$\sim p \to$ **False**
$\bullet\ p$

Quantitative Operators: ∃! (there exists one and only one)

Quantitative Statements

01. Negation: $\forall \to \exists$; $P(x) \to \sim P(x)$
02. Contrapositive: $\forall x \in D, \sim Q(x) \to \sim P(x)$
03. Converse: $\forall x \in D, Q(x) \to P(x)$
04. Inverse: $\forall x \in D, \sim P(x) \to \sim Q(x)$

Universal Instantiation

$\forall x \in D, P(x)$
$a \in D$
$\bullet\ P(a)$

Rule + Universal Instantiation = Universal Rule

Definition of Numbers

01. Even: $\exists k \in \mathbf{Z}$ such that $x = 2k$
02. Odd: $\exists k \in \mathbf{Z}$ such that $x = 2k + 1$
03. Prime: $\forall r, s \in \mathbf{Z}+$, $n = rs \Rightarrow (r = 1, s = n)$ or $(r = n, s = 1)$
04. Composite: $\exists r, s \in \mathbf{Z}+$, $(n = rs)$ and $(1 < r, s < n)$
05. Rational: $\exists p, q \in \mathbf{Z}$, $r = \frac{p}{q}$ and $q \neq 0$
06. Divisible: $\exists k \in \mathbf{Z}$, $n = dk$

Proof by Contradiction

The contrapositive of $P(x) \to Q(x)$ is $\sim Q(x) \to \sim P(x)$.

1. Prove the contrapositive statement through a direct proof.
   1.1. Suppose $x \in \mathbf{D}$ such that $Q(x)$ is **False**.
   1.2. Show that $P(x)$ is **False**.
2. Therefore, the original statement $P(x) \to Q(x)$ is **True**.

Sets and Functions

Set Concepts

01. Equal Sets: $A = B \Leftrightarrow x \in A \leftrightarrow x \in B$
02. Subset: $A \subseteq B \Leftrightarrow \forall x,\ x \in A \to x \in B$
03. Finite Set: $|S| = n$, where n is called cardinality
04. Power Set: $P(A)$ is the set of all subsets of A
05. Cartesian Product: $A \times B = \{(a,\ b) \mid a \in A,\ b \in B\}$

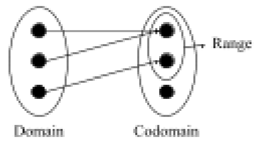
Set Operations

01. Union: $A \cup B = \{x \mid (x \in A) \vee (x \in B)$
02. Intersection: $A \cap B = \{x \mid (x \in A) \wedge (x \in B)$
03. Compliment: $B - A = B \backslash A = B \cap \overline{A}$
04. Complement: $\overline{A} = U - A$
05. Disjoint: $A \cap B = \emptyset$

Set Identities

01. Commutative Laws: $A \cap\!\!\!\!\cup B \equiv B \cap\!\!\!\!\cup A$
02. Associative Laws: $(A \cap\!\!\!\!\cup B) \cap\!\!\!\!\cup C \equiv A \cap\!\!\!\!\cup (B \cap\!\!\!\!\cup C)$
03. Distribution Laws: $A \cap (B \cup C)$; $A \cup (B \cap C)$
04. Identity Laws: $A \cap U = A$; $A \cup \emptyset = A$
05. Negation Laws: $A \cap \overline{A} = \emptyset$; $A \cup \overline{A} = U$
06. Double Negative Laws: $(\overline{\overline{A}}) = A$
07. Idempotent Laws: $A \cap A = A$; $A \cup A = A$
08. Universal Bound Laws: $A \cap \emptyset = \emptyset$; $A \cup U = U$
09. De Morgan's Laws: $\overline{A \cap B} = \overline{A} \cup \overline{B}$; $\overline{A \cup B} = \overline{A} \cap \overline{B}$
10. Absorption Laws: $A \cup (A \cap B) = A$; $A \cap (A \cup B) = A$
11. Negation of True/False: $\overline{U} = \emptyset$; $\overline{\emptyset} = U$

Function Concepts



01. $f : X \to Y$ is injective iff $\forall a,\ b \in X,\ f(a) = f(b) \Rightarrow a = b$
02. $f : X \to Y$ is surjective iff $\forall y \in Y,\ \forall x \in X\ (f(x) = y)$
03. Bijective: 1-1 + onto
04. Inverse Functions: Let $f : X \to Y$ be a bijection. Then its inverse $g : Y \to X$: $\forall y \in Y,\ g(y) = x \Leftrightarrow f(x) = y$
05. Image: $f(X) = \{f(x) \mid x \in X\}$
06. Preimage: $f^{-1}(Y) = \{x \in X \mid f(x) \in Y\}$

Induction

Mathematical Induction

1. For each $n \in D$, let $P(n)$ be the proposition <XXX>.
2. (Base step) $P(1)$ is true because <RRR>.
3. (Induction step)
   3.1. Let $k \in D$ such that $P(k)$ is true, i.e. <XXX>.
   3.2. <YYY>
   …
   3.n. Thus $P(k + 1)$ is true.
4. Hence $\forall n \in D,\ P(n)$ is true by Mathematical Induction.

Strong Induction

1. For each $n \in D$, let $P(n)$ be the proposition <XXX>.
2. (Base step) $P(1)$ is true because <RRR>.

3. (Induction step)
   3.1. Let $k \in D$ such that $P(1)$, …, $P(k)$ is true, i.e. <XXX>.
   3.2. <YYY>
   …
   3.n. Thus $P(k + 1)$ is true.
4. Hence $\forall n \in D,\ P(n)$ is true by Strong Induction.

Integers

Definition - Divisibility
Let $n,\ d \in Z$ and $d \neq 0$. $\exists k \in Z\ (n = dk) \Rightarrow d \mid n$.

Theorem - Properties of Division
01. If $a \mid b,\ b \mid c$, then $a \mid c$ (Transitive Property).
02. $\forall m,\ n \ni Z\ (a \mid b \land a \mid c \Rightarrow a \mid mb + nc)$

Theorem - Division Algorithm
Let $n \in Z$ and $d \in Z^+$. Then there are unique integers q and r, with $0 \leq r < d$ such that $n = dq + r$.

Definition - Modular Arithmetic
Let $a,\ b \in Z,\ m \in Z^+$. $a \equiv b\ (mod\ n)$ if $n \mid (a - b)$.

Theorem (Chpt. 4 Pg. 3)
01. $a \equiv b\ (mod\ m)$ iff $a\ mod\ m\ =\ b\ mod\ m$
02. $a \equiv b\ (mod\ m)$ iff $\exists k \in Z\ (a = b + km)$
03. If $a \equiv b\ (mod\ m),\ c \equiv d\ (mod\ m)$, then
    $a + b = c + d\ (mod\ m),\ ac = bd\ (mod\ m)$.

Definition - Prime Number
A positive number is:
(1) prime, if it has exactly 2 divisors, 1 and itself.
(2) composite, if it has more than 2 divisors.

Theorem (Chpt. 4 Pg. 5)
Every positive integer n greater than 1 has at least 1 prime divisor.

Theorem - Prime Factorisation Theorem (Fundamental Theorem of Arithmetic)
Every positive integer greater than 1 can be written uniquely as a product of primes where the prime factors are written in order of nondecreasing size.

Theorem (Chpt. 4 Pg. 6)
If n is composite, then it has a divisor d with $1 \leq d \leq \sqrt{n}$.

Theorem (Chpt. 4 Pg. 6)
There are infinitely many primes.

Definition - Greatest Common Divisor (GCD)
Let a and b be integers, not both zero. GCD of a and b is the greatest integer d such that $d \mid a$ and $d \mid b$. GCD can be computed using prime factorisation.

Definition - Relatively Prime
Integers a and b are relatively prime if $gcd(a, b) = 1$.

Theorem - Base *b* Expansion of Integers
Base b expansion of any integer is unique.

Algorithm for Base *b* Expansion
procedure base b expansion of $n \in Z^+$
q := n
k := 0
while q ≠ 0:
begin
    a_k := q **mod** b
    q := q // b
    k := k + 1
end base b expansion of n is $(a_{k-1}...a_1a_0)_n$.

Theorem - Euclidean Algorithm
$a\ mod\ b = r \Rightarrow gcd(a,\ b) = gcd(b,\ r)$

Euclidean Algorithm
x := a
y := b
while y ≠ 0:
begin
    r := x **mod** y
    x := y
    y := r
end {gcd(a, b) = x}

Theorem - Result of Euclidean Algorithm
Let $a,\ b \in Z^+,\ r = gcd(a,\ b).\ \exists m, n \in Z\ (r = am + bn)$.

Theorem (Chpt. 4 Pg. 10)
$a,\ b,\ c \in Z^+.\ (gcd(a,\ b) = 1 \land a \mid bc) \Rightarrow a \mid c$.

Theorem (Chpt. 4 Pg. 11)
$Prime(p) \land p \mid a_0a_1...a_n \Rightarrow p \mid a_k$ for some k

Theorem - Cancellation
If $ac \equiv bc\ (mod\ m)\ \land\ gcd(c,\ m)\ =\ 1$, then $a \equiv b\ (mod\ m)$.

Definition - Multiplicative Inverse
$\overline{a}a \equiv 1\ (mod\ m)$. This exists iff gcd(a, m) = 1.

Theorem (Chpt. 4 Pg. 12)
$ax \equiv b\ (mod\ n) \Rightarrow x \equiv \overline{a}b\ (mod\ n)$

Theorem - Fermat's Little Theorem
If p is a prime and gcd(a, p) = 1, then $a^{p-1} \equiv 1\ (mod\ p)$

Relations

Definition - Relation
Let A and B be sets. A relation R is a subset of $A \times B$. We have xRy iff $(x,\ y) \in R$. $R^{-1} = \{(y,\ x) \in B \times A \mid xRy\}$.

Definition - Equivalence Relation
Let R be a relation on set A.
01. Reflexive: $\forall x \in A,\ xRx$.
02. Symmetric: $\forall x,\ y \in A,\ xRy \Rightarrow yRx$.
03. Antisymmetric: $\forall x,\ y \in A,\ (xRy \land yRx) \Rightarrow x = y$.
04. Transitive: $\forall x,\ y,\ z \in A,\ (xRy \land yRz) \Rightarrow xRz$.
05. Equivalence: Reflexive + Symmetric + Transitive
06. Partial Order: Reflexive + Antisymmetric + Transitive

Definition - Equivalence Class
The equivalence class of a, $[a]_R = \{x \in a \mid aRx\}$. The set of all equivalent classes, $A/R = \{[a]_R \mid a \in A\}$.

Definition - Partition
A collection of non-empty sets $A_1,\ A_2,\ ...$ forms a partition of set S if:
(1) $A_1 \cup A_2 \cup ... = S$;

(2) $A_1, A_2, \ldots$ are mutually disjoint.

Partial Order Concepts

01. Comparable: $a \preccurlyeq b$ or $b \preccurlyeq a$
02. Maximal: $\sim(\exists c \in A\ (a \prec c))$
03. Minimal: $\sim(\exists c \in A\ (a \succ c))$
04. Largest/Greatest/Maximum: $\forall b \in A,\ b \preccurlyeq a$
05. Smallest/Least/Minimum: $\forall b \in A,\ b \succcurlyeq a$
06. Total Order: $\forall a,\ b \in P,\ a,\ b$ is comparable.
07. Well Order: Every non-empty subset of *P* has a smallest element.

Theorem (Chpt. 5 Pg. 8)

Every finite non-empty poset *S* has a minimal element and a maximal element.

Theorem (Chpt. 5 Pg. 9)

Every non-empty poset *S* has at most one minimum element and at most one maximum element.

Counting and Probability

Multiplication Principle

If there are *m* ways of doing something and *n* ways of doing another thing, then there are *mn* ways of doing both things.

Addition Principle

If we have *m* ways of doing something and *n* ways of doing another thing but we cannot do both things at the same time, then there are *m* + *n* ways of choosing one thing to do.

Pigeonhole Principle

A function from a finite set to a smaller finite set cannot be one-to-one: there must be at least 2 elements in the domain that have the same image in the co-domain.

For any function *f* from a finite set *X* with *n* elements to a finite set *Y* with *m* elements and for any positive integer *k*, if for each $y \in Y$, $f^{-1}(\{y\})$ has at most *k* elements, then *X* has at most *km* elements; in other words, $n \le km$.

Theorem 9.6.1

no. of combinations with repetition allowed = $(n + r - 1)Cr$.

Theorem 9.7.1 - Pascal's Formula

$(n+1)Cr = nCr + nC(r-1)$

Theorem 9.7.2 - Binomial Theorem

$(a+b)^n = nC0a^n + nC1a^{n-1}b + \ldots + nCnb^n$

Theorem 6.3.1

If $|X| = $ n, $|\wp(X)| = 2^n$.

Theorem 9.9.1 - Bayes' Theorem

Suppose that a sample space *S* is a union of mutually disjoint events $B_1, B_2, \ldots, B_n$. Suppose *A* is an event in *S*, and suppose *A* and all the B_i have non-zero probabilities.

If *k* is an integer with $1 \le k \le n$, then

$$P(B_k|A) = \frac{P(A|B_k) \times P(B_k)}{P(A|B_1) \times P(B_1) + P(A|B_2) \times P(B_2) + \ldots + (A|B_n) \times P(B_n)}$$

Definition - Independent Events

A and *B* are independent iff $P(A \cap B) = P(A)P(B)$.

Graphs

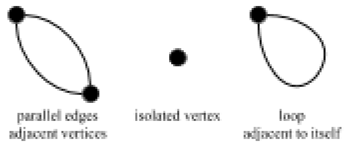
Definition - Undirected Graphs

$G = (V, E)$

$V = \{v_1, v_2, \ldots, v_n\}$ is the non-empty set of vertices (nodes).

$E = \{e_1, e_2, \ldots, e_n\}$ is the set of edges.

$e_1 = \{v_1, v_2\}$ is the undirected edge connecting v_1 and v_2 (endpoints). e_1 is incident on v_1 and v_2.

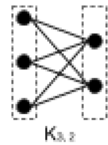


Definition - Directed Graphs

$e_1 = (v_1, v_2)$ is the directed edge connecting v_1 and v_2 (endpoints).

Type of Graphs

01. Simple Graphs: A simple graph is an undirected graph that does not have any parallel edges/loops.
02. Complete Graphs: A complete graph K_n is a simple graph with *n* vertices and exactly one edge connecting each pair of distinct vertices. It has $\frac{n(n-1)}{2}$ edges in total.
03. Bipartite Graphs: A bipartite graph (bigraph) is a simple graph whose vertices can be divided into two disjoint sets *U* and *V* such that each edge connects one vertex in *U* to one in *V*.
04. Bipartite Complete Graph: A bipartite complete graph $K_{m,n}$ is a bipartite graph where every vertex in U is connected to every vertex in V.



05. Subgraph: *H* is a subgraph of *G* if every vertex of *H* is a vertex of *G*, every edge of *H* is an edge of *G* and every edge in *H* has the same endpoints as it has in *G*.

Definition - Degree

$deg(v)$ equals the total number of edges that are incident on *v*, with a loop counted twice. Total degree of a graph equals the sum of degrees of all its vertices.

Basic Concepts of Walks

01. A walk from *v* to *w* is a finite alternating sequence of adjacent vertices and edges of *G* ($v_0e_0v_1e_1v_2 \ldots v_{n-1}e_{n-1}v_n$). The length of this walk is *n*. A walk *v* to *v* consisting of a single *v* is a trivial walk. A close walk is a walk that starts and ends at the same vertex.
02. Trial: A trail is a walk without a repeated edge.
03. Path: A path is a walk without a repeated vertex.
04. Circuit: A circuit (cycle) is a closed walk without a repeated edge. An undirected graph is cyclic if it contains at least one loop or cycle.
05. Simple Circuit: A simple circuit is a circuit that does not contain repeated vertices other than the first and last.

Definition - Connectedness

Two vertices *v* and *w* are connected if and only if there is a walk from *v* to *w*.

A graph is connected if and only if any two vertices are connected.

A connected component of a graph is a connected subgraph of largest possible size.

Theorem 10.1.1 - The Handshake Theorem

total degree of G $= 2 \times$ *no. 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 number of vertices of odd degree.

Lemma 10.2.1
Let G be a graph.

a. If G is connected, then any two distinct vertices of G can be connected by a path.
b. If vertices v and w are part of a circuit in G and one edge is removed from the circuit, then there still exists a trial 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 without disconnecting G.

Definition - Euler Circuit
Let G be a graph. An Euler circuit of G is a circuit that contains every vertex and traverses every edge of G exactly once.

Definition - Eulerian Graph
An Eulerian graph is a graph that contains an Euler circuit.

Theorem 10.2.2
If a graph has an Euler circuit, then every vertex of the graph has a positive even degree. / If any vertex of a graph has a positive odd degree, the graph does not have an Euler circuit. (contrapositive version)

Theorem 10.2.3
If a graph is connected and the degree of every vertex of G is a positive even integer, then G has an Euler circuit.

Theorem 10.2.4
A graph G has an Euler circuit if and only if G is connected and every vertex of G has a positive even degree.

Definition - Euler Trail/Path
Let G be a graph, and let v and w be two distinct vertices of G. An Euler trail/path from v to w is a sequence 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 let v and w be two distinct vertices of G. There is an Euler Trail from v to w if and only if G is connected, v and w have odd degree and all other vertices have positive even degree.

Definition - Hamilton Circuit
Let G be a graph. A Hamilton circuit of G is a simple circuit that includes every vertex of G.

Definition - Hamiltonian Graph
A Hamiltonian graph is a graph that contains a Hamiltonian circuit.

Proposition 10.2.6
A Hamiltonian graph G has a subgraph H with the following properties:

a. H contains every vertex of G.
b. H is connected.
c. H has the same number of edges as vertices.
d. Every vertex of H has degree 2.

Theorem 10.3.2
A^n_{ij} = *no. of walks of length n from* v_i *to* v_j

Definition - Isomorphic Graph
Let $G = (V_G, E_G)$ and $G' = (V_{G'}, E_{G'})$ be two graphs.
$G \simeq G' \Leftrightarrow \exists$ *bijections* $g : V_G \to V_{G'},\ h : E_G \to E_{G'}$
$(\forall v \in V_G,\ e \in E_G\ (v \text{ is an endpoint of } e \Leftrightarrow g(v) \text{ is an endpoint of } h(e)))$

Theorem 10.4.1
Graph isomorphism is an equivalent relation.
Definition - Planar Graph
A planar graph is a graph that can be drawn on a 2D plane without edge crossing.

Kuratowski's Theorem
A finite graph is planar if and only if it does not contain a subgraph that is a subdivision of the complete graph K_5 or the complete bipartite graph $K_{3,3}$.

Euler's Formula
For a connected simple planar graph,
no. of faces = *no. of vertices* + *no. of edges* – 2

Trees

Definition - Tree
A graph is called a tree if and only if it is circuit-free and connected. A trivial tree is a graph that contains a single vertex.
A graph is called a forest if and only if it is circuit-free and not connected.

Lemma 10.5.1
Any non-trivial tree has at least one vertex of degree 1.

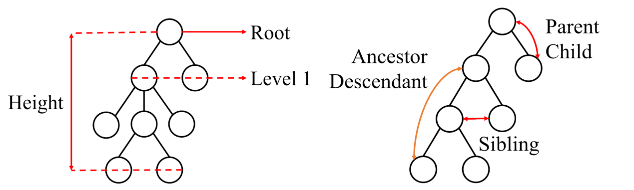
Definition - Terminal Vertex (Leaf) and Internal Vertex
If a tree has one or two vertices, each vertex is called a terminal vertex; otherwise, vertices of degree 1 are called terminal vertex and others are called internal vertex.

Theorem 10.5.2
Any tree with n vertices has n - 1 edges.

Lemma 10.5.3
If G is any connected graph, C is any circuit of G, when one of the edges of C is removed from G, the graph that remains is still connected.

Theorem 10.5.4
If G is a connected graph with n vertices and n - 1 edges, then G is a tree.

Definition - Rooted Tree



Theorem 10.6.1 - Full Binary Tree Theorem

If T is a full binary tree with k internal vertices, then T has a total of $2k + 1$ vertices and $k + 1$ terminal vertices (leaves).

Theorem 10.6.2

$t \leq 2^h$

Depth-First Search

01. Pre-order: entry - left - right
02. In-order: left - entry - right
03. Post-order: left - right - entry

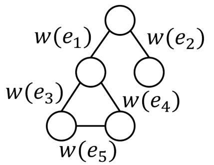
Definition - Spanning Tree

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

01. Every connected graph has a spanning tree.
02. Any two spanning trees for a graph have the same number of edges.

Definition - Weighted Graph



$$w(G) = \sum_{i=1}^{5} w(e_5)$$

Kruskal's Algorithm

Input: G (no. of vertices: n)

01. Initialise T to contain all vertices of G and no edges.
02. Let E be the set of all edges of G. Let $m = 0$.
03. while $m < n - 1$:
    a. Find an edge e in E of the least weight.
    b. Delete e from E.
    c. If e does not produce a circuit in T, add e to T.
    d. m = m + 1
    End while

Output: T

Prim's Algorithm

Input: G (no. of vertices: n)

01. Initialise T to contain one vertex v of G and no edges.
02. Let V be the set of all vertices of G except v.
03. for $i = 1$ to $n - 1$:
    a. Find an edge e in G such that: (1) e connects T to one of the vertices in V and (2) e has the least weight of all edges connecting T to one of the vertices in V. Let w be the endpoint of e that is in V.
    b. Add e and w to T. Delete w from V.

Output: T