CS182 Cheatsheet

Logical Equivalences Cheatbox

Identity Laws

$$p \wedge \mathrm{T} \equiv p$$
$$p \vee \mathrm{F} \equiv p$$

Domination Laws

$$p \vee \mathrm{T} \equiv \mathrm{T}$$
$$p \wedge \mathrm{F} \equiv \mathrm{F}$$

Idempotent Laws

$$p \vee p \equiv p$$
$$p \wedge p \equiv p$$

Double Negation

$$\neg(\neg p) \equiv p$$

Commutative Laws

$$p \vee q \equiv q \vee p$$
$$p \wedge q \equiv q \wedge p$$

Associative Laws

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

Distributive Laws

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

De Morgan's Laws

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

Absorption Laws

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

Negation Laws

$$p \vee \neg p \equiv \mathrm{T}$$
$$p \wedge \neg p \equiv \mathrm{F}$$

Implication and Biconditional

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

;

Set Identities Cheat Sheet

Identity Laws:

$$A \cup \emptyset = A$$
$$A \cap U = A$$

Domination Laws:

$$A \cup U = U$$
$$A \cap \emptyset = \emptyset$$

Idempotent Laws:

$$A \cup A = A$$
$$A \cap A = A$$

Complement Laws:

$$A \cup A^c = U$$
$$A \cap A^c = \emptyset$$
$$(A^c)^c = A$$

Commutative Laws:

$$A \cup B = B \cup A$$
$$A \cap B = B \cap A$$

Associative Laws:

$$A \cup (B \cup C) = (A \cup B) \cup C$$
$$A \cap (B \cap C) = (A \cap B) \cap C$$

Distributive Laws:

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$
$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

De Morgan's Laws:

$$(A \cup B)^c = A^c \cap B^c$$
$$(A \cap B)^c = A^c \cup B^c$$

Absorption Laws:

$$A \cup (A \cap B) = A$$
$$A \cap (A \cup B) = A$$

Set Difference:

$$A - B = A \cap B^c$$

Functions

- **Injective -** one-to-one
- **Surjective -** onto
- **Bijective -** both one-to-one and onto

Summation Formulas Cheat Sheet

Basic Summations:

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$$
$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$
$$\sum_{k=1}^{n} k^3 = \left(\frac{n(n+1)}{2}\right)^2$$

Arithmetic Series:

$$\sum_{k=0}^{n}(a+kd) = \frac{n+1}{2}\left[2a+nd\right]$$

Geometric Series:

$$\sum_{k=0}^{n} ar^k = a \cdot \frac{1-r^{n+1}}{1-r} \quad \text{if } r \neq 1$$

Infinite Geometric Series:

$$\sum_{k=0}^{\infty} ar^k = \frac{a}{1-r} \quad \text{if } |r| < 1$$

Telescoping Sum:

$$\sum_{k=1}^{n}\left(\frac{1}{k} - \frac{1}{k+1}\right) = 1 - \frac{1}{n+1}$$

Binomial Theorem:

$$(a+b)^n = \sum_{k=0}^{n}\binom{n}{k}a^{n-k}b^k$$

Graphs

- Handshake Theorem - The sum of the degrees of the vertices equals twice the number of edges $\sum_{v \in V} deg(v) = 2|E|$
- **Dirac's Theorem** If G is a simple graph with n vertices with $n \geq 3$ such that the degree of every vertex in G is at least n/2, then G has a Hamilton circuit.
- **Ore's Theorem** If G is a simple graph with n vertices with $n \geq 3$ such that $deg(u) + deg(v) \geq n$ for every pair of nonadjacent vertices u and v in G , then G has a Hamilton circuit.
- **Euler's Formula** $r = e - v + 2$
- A connected planar graph satisfies $e \leq 3v - 6$
- If G is a connected planar graph with e edges and v vertices with $v \geq 3$ and no circuits of length three, then $e \leq 2v - 4$

Big-O

$n^n > n! > n^k > n > nlog(n) > n > \sqrt{n} > log(n) > k$

Graphs

- Simple graph
- Complete graph
- Cycle
- Wheel
- Hypercube
- Bipartite graph
- Spanning tree

Graphs

- **Euclidian Algorithm** gcd(a, b) = gcd(b, r)
- **Bezout's Theorem** gcd(a, b) = sa + tb
- **Fermat's Little Theorem:**
  $a^p \equiv a(\mod p)$
  $a^{p-1} \equiv 1(\mod p)$
- **RSA Cryptosystem** p, q
  $N = p * q$
  $\varphi = (p-1)(q-1)$
  $gcd(e, \varphi) = 1$
  $(e * (d) \mod (\varphi)) = 1$, compute d
  Public encryption key is N, e
  Private (decryption) key: is d

- If $m > 1$ and $gcd(m, n) = 1$ then an inverse of n modulo m exists and it is equal to s in the following equation: s x n + t x m = 1
- $m^{p-1} \equiv 1 \mod p \equiv m^p \equiv m \mod p$

Finite Automata

- Two types of FAs:
  - **DFA**: Each state has one transition for each input symbol.
  - **NFA**: States can have multiple transitions or none for each input symbol.
- Both DFAs and NFAs recognize regular languages.
- The language accepted by an FA consists of strings that reach an accept state.
- A DFA is formally defined by the 5-tuple:

  $$(Q, \Sigma, \delta, q_0, F)$$

  where Q is the set of states, Σ is the alphabet, δ is the transition function, q_0 is the start state, and F is the set of accept states.

Regex

- {a,b}*

Tree Traversal

- **Preorder (Prefix)**: Visit `root → left → right`
- **Inorder (Infix)**: Visit `left → root → right`
- **Postorder (Postfix)**: Visit `left → right → root`

Build Tree from Prefix

1. Read the expression from **left to right**.
2. If the token is an **number**, create a leaf node.
3. If the token is an **operator**
    - Create an operator node.
    - Recursively build the **left** and then the **right** subtree from the next tokens.

Build Tree from Postfix (Right to Left)

1. Read the expression from **right to left**.
2. If the token is an **operator**:
    - Create an operator node.
    - Recursively build the **right** subtree first.
    - Then recursively build the **left** subtree.
    - Attach both subtrees to the operator node.
3. If the token is a **number**, create a leaf node and return it