Midterm 3 note sheet

Relations and functions, graphs, finite automata and trees

Table 1: **Relations and functions.**

Relation on A	$R \subseteq A \times A$, and $(a\ R\ b) \Leftrightarrow ((a,b) \in R)$
Reflexive	$\forall a \in A, a\ R\ a$
Symmetric	$\forall a, b \in A, ((a\ R\ b) \Rightarrow (b\ R\ a))$
Transitive	$\forall a, b, c \in A, (((a\ R\ b) \wedge (b\ R\ c)) \Rightarrow (a\ R\ c))$
Antisymmetric	$\forall a, b \in A, (((a\ R\ b) \wedge (b\ R\ a)) \Rightarrow (a = b))$
Equivalence relation	reflexive, symmetric and transitive
Equivalence class	$[a] = \{b \in A : b\ R\ a\}$
Injective $f : A \to B$	$\forall a_1, a_2 \in A, ((f(a_1) = f(a_2)) \Rightarrow (a_1 = a_2))$
Surjective	$\forall b \in B, \exists a \in A, f(a) = b$
Bijective	injective and surjective
Inverse of a bijection	$f^{-1} : B \to A$ with $(f^{-1}(b) = a) \Leftrightarrow (f(a) = b)$
Floor	$\lfloor x \rfloor = \max\{n \in \mathbb{Z} : n \leq x\}$

Table 2: **Graphs.**

Degree	$\deg(v)$: edges with endpoint v, a loop counting twice
Walk from u to v	$v_0, e_1, v_1, \ldots, e_n, v_n$, $v_0 = u$, $v_n = v$, e_i joins v_{i-1} and v_i; length n
Trail, path	a walk with no repeated edge; with no repeated vertex
Circuit	a trail of length ≥ 1 from a vertex to itself
Connected	a walk between every two vertices
Euler circuit	a circuit that contains every edge
Euler trail	a trail from u to $v \neq u$ that contains every edge
Hamiltonian cycle	a circuit that visits every vertex once, then returns to its start
Bipartite	$V = V_1 \cup V_2$ with $V_1 \cap V_2 = \emptyset$, and every edge has one endpoint in V_1 and one in V_2

```
procedure bfs(G, s)
    mark s, and set dist[s] := 0
    Q := a queue containing only s
    while Q is not empty
        v := remove the front of Q
        for each vertex u in the list of v, in order
            if u is not marked
                mark u, and set dist[u] := dist[v] + 1
                add u to the back of Q
```

```
procedure dfs(G, v)   // start: dfs(G, s)
    mark v
    for each vertex u in the list of v, in order
        if u is not marked
            dfs(G, u)
```

Table 3: **Finite automata.**

DFA	$M = (S, \Sigma, \delta, s_0, F)$: states, alphabet, $\delta : S \times \Sigma \to S$, initial state, accepting states
Transition table	$\to$ marks the initial state, $*$ each accepting state
Empty string	λ
Run on $w = x_1 \cdots x_n$	$r_0, \ldots, r_n$ with $r_0 = s_0$ and $r_i = \delta(r_{i-1}, x_i)$
Accepts w	$r_n \in F$
Language	$L(M) = \{w : M \text{ accepts } w\}$

Table 4: **Trees.**

Tree	a connected graph with no cycles
Edges of a tree	a tree with n vertices has $n-1$ edges
Binary tree	`nil`, or node(x, L, R): root label x, subtrees L and R
In Racket	`nil` is `'()`, node(x, L, R) is `(list x L R)`
Preorder	root, preorder of L, preorder of R
Inorder	inorder of L, root, inorder of R
Postorder	postorder of L, postorder of R, root
Level, height	the root is at level 0, a child one below its parent; height is the largest level
Leaf	a vertex with no children
Leaf of a tree with no root	a vertex of degree 1