Discrete Structures	Final Exam Review
CS2800 Fall 2013	Tuesday, December 10, 2013

Our final exam will be on Thursday, December 19, 9-11:30am in Barton Hall. The exam will be $2\frac{1}{2}$ hours and will consist mostly of short answer questions, although there may a proof or two. It will be a cumulative exam, covering the entire course. In addition to the material covered on the earlier exams (see the Prelim 1 and Prelim 2 review sheets for coverage), the exam will cover §10.1-6 and §13.1-4 of the course text and the handouts on regular expressions and finite automata and DFA state minimization.

Material covered on Prelim 1 (see the Prelim 1 review sheet for details)

- §1: Propositional logic, predicate logic, proof construction
- §2.1-5 and §9.1, §9.3-6: Sets, functions, relations
- §4.1-4: Number theory, modular arithmetic
- §5.1-3: Mathematical induction

Material covered on Prelim 2 (see the Prelim 2 review sheet for details)

- §6: Counting techniques
- §7: Discrete probability theory
- §8: Recurrence relations, generating functions, inclusion-exclusion

Material covered since Prelim 2

- §10.1-6: Graphs
- §13.1-4: Finite-state automata, context-free grammars
- Handout: Regular expressions
- Handout: DFA state minimization

§10.1-6 covers graphs. You should know the definitions of the following concepts: undirected graph, directed graph, node or vertex, edge or arc, endpoints, adjacency, degree of a node in an undirected graph, outdegree and indegree of a node in a directed graph, complete graph K_n, bipartite graph, complete bipartite graph $K_{m,n}$, coloring, subgraph, induced subgraph, path, length of a path, cycle or circuit, length of a cycle, simple path and simple cycle,[1] connected, connected component, strongly connected component, cut vertex, cut edge, acyclic, planar, isomorphism of graphs, Euler path or circuit, Hamiltonian path or circuit, shortest paths.

You should be familiar with the basic graph representations: adjacency matrix, incidence matrix, adjacency list. Given a graph in any one of these representations or by picture or other form of specification, you should be able to convert among the different representations.

You should understand depth-first search and breadth-first search algorithms and how to apply them. Given a graph, you should be able to determine the order in which the nodes are visited and be able to give the depth-first and breadth-first numbering of the vertices. You should know what a topological sort of a directed

[1]Note: the definition of *simple* as given in the text (Definitions 1 and 2, pp. 679 and 680) is not consistent with standard usage. A path or cycle is *simple* if it contains no vertex more than once. In an undirected graph, a single vertex or two vertices connected by an edge are usually not considered to be cycles. Thus an undirected cycle can only be of length three or more.

acyclic graph is and how to compute one. You should understand Dijkstra's shortest path algorithm and how to apply it. Given a graph labeled with edge weights and a start node s, you should be able to show the sequence of nodes for which the shortest path from s is found using Dijkstra's algorithm.

Study the Review Questions on pp. 737-738.

Practice problems:
pp. 649ff: 13, 27, 29
pp. 665ff: 18, 21, 23, 25, 39, 41, 55, 59(a-c)
pp. 675ff: 1, 5, 35, 37, 49, 57(a), 61, 63
pp. 689ff: 11, 15, 28, 30, 56, 61
pp. 703ff: 1, 7, 31, 37
pp. 716ff: 3
pp. 738ff: 1, 3, 7, 29, 35, 37, 39. If you are up for an extra challenge, a really nice problem is 42. (*Hint*: Induction on n.)

§13.1-4 and the two handouts cover finite-state automata and context-free grammars. You should know the definitions of the following concepts: alphabet, string, language, deterministic finite automaton, nondeterministic finite automaton, transition function, acceptance, regular expression, regular set, context-free grammar, context-free language, production, terminal and nonterminal symbols, subset construction, quotient construction.

Given a simple finite automaton, you should be able to describe the set of strings accepted and vice versa. Given a simple regular expression, you should be able to describe the set of strings matching it and vice versa. Given a simple context-free grammar, you should be able to describe the set of strings generated and vice versa. You should be able to construct a regular expression equivalent to a given finite automaton and vice-versa. You should know how to minimize a given deterministic finite automaton using the quotient construction. Given a nondeterministic finite automaton, you should know how to construct an equivalent deterministic one using the subset construction.

You should know the inductive definition of the extended transition function $\widehat{\delta}$ of a deterministic finite automaton. You may be asked to prove a theorem by induction involving this function.

Study the Review Questions on pp. 900ff.

Practice problems:
pp. 855ff: 4, 17, 21
pp. 875ff: 1, 5, 9, 11, 23, 25, 43, 45, 51, 62(b)
pp. 887ff: 1, 3, 7, 13.