Automata and Formal Languages Comprehensive Exam (60 Points)

Fall 2007

Note: this 1-hour exam is *closed book*.

Problem 1 (18 points)

(a) Give an algorithm that takes a DFA over the alphabet $\{0, 1\}$ as input and decides whether or not it accepts a non-empty language. Your algorithm should run in time linear in the number of states of the DFA.

(b) Recall that in a CNF grammar, every production either has the form $A \rightarrow BC$ (where A, B, C are variables) or $A \rightarrow a$ (where A is a variable and a is a terminal).

Fix a CNF grammar G. Give an algorithm that takes a string w as input and decides whether or not w is a member of the language generated by G. Your algorithm should run in time polynomial in the length n of the given string w (e.g., $O(n^3)$ time is fine).

Problem 2 (12 points)

Classify each of the following languages as being in one of the following classes of languages: *empty, finite, regular, context-free, recursive, recursively enumerable, all languages.* You must give the *smallest* class that contains *every possible language* fitting the following definitions. For example, the language of a DFA must always be context-free, but the smallest class that contains all such languages is that of the *regular* languages. *Do not provide explanations. Correct answers receive 3 points, incorrect answers receive -2 points.*

(a) $\{0^n 1^n 2^n \mid n \geq 1\}$.

(b) $\{a^{n!} \mid n \geq 1\}$.

(c) An NP-complete problem.

(d) The intersection of two context-free languages.

Problem 3 (12 points)

Decide whether each of the following are recursive, RE-but-not-recursive, or non-RE. *Do not provide explanations. Correct answers receive 3 points, incorrect answers receive -2 points.*

(a) The set of all TM codes for TMs that halt on every input.

(b) The set of all TM codes for TMs that halt on no input.

(c) The set of all TM codes for TMs that halt on at least one input.

(d) The set of all TM codes for TMs that, on at least one input, fail to halt.

Problem 4 (18 points)

Assume that the following *Node-Cover problem* is NP-complete: given an undirected graph G and an upper limit k, does there exist a node cover (a set of nodes of G that contains at least one endpoint of each edge of G) with at most k nodes?

Use this to prove that the following *Clique* problem is NP-complete: given an undirected graph G and a lower limit k, does there exist a clique (a set of nodes of G such that there is an edge between each pair of nodes in the set) with at least k nodes?