Automata and Formal Languages Comprehensive Exam (65 Points)

Fall 2006

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

Problem 1 (20 points)

For each of the following statements, provide a high-level proof or an explicit counterexample. You can assume that the various descriptions of regular and context-free languages are equivalent. If you provide a counterexample, you do not need to prove that it is not regular/context-free as long as it is correct.

(a) The union of two regular languages is regular.

(b) The intersection of two regular languages is regular.

(c) The union of two context-free languages is context-free.

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

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) A PSPACE-complete problem.

(b) A subset of a recursive language.

(c) $\{0^n1^n \mid n \geq 1\}$.

(d) $\{a^n \mid n \text{ is prime}\}$.

Problem 3 (18 points)

Let $L_{ne} = \{M \mid L(M) \neq \emptyset\}$ be the strings that encode a Turing machine that accepts a non-empty language.

(a) Give a high-level proof showing that L_{ne} is a recursively enumerable language.

(b) Give a high-level proof showing that L_{ne} is not recursive (i.e., is undecidable). Assume that the universal language $L_u = \{(M, w) \mid M \text{ accepts } w\}$ is not recursive.

Problem 4 (15 points)

The *subgraph-isomorphism problem* is the following: given graphs G_1 and G_2, does G_1 contain a copy of G_2 as a subgraph? That is, can we find a subset of the nodes of G_1 that, together with the edges among them in G_1, forms an exact copy of G_2? Prove that the subgraph-isomorphism problem is NP-hard.

[Hint: make use of the NP-complete problem CLIQUE.]