Foundations Qualifying Examination

August 29, 1997

1. Let L, L_1, and L_2 be regular languages defined over an alphabet Σ.

   (a) Prove that $\overline{L}$ (i.e, the complement of L) is a regular language.

   (b) Prove that $L_1 \cap L_2$ is a regular language.

   (c) Prove that the class of regular languages over Σ is not closed under infinite unions. (You may assume that there exists a language over Σ that is non-regular.)

2. Prove that the language

   $$L = \{\mathbf{1}^n \mid n \text{ is a composite number (i.e, non-prime number)}\}$$

   is not a regular language.

3. (a) Give a context-free grammar that generates the language

   $$A = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ and either } i = j \text{ or } j = k\}.$$

   (b) Is your grammar ambiguous? Why or why not?

4. Let $\mathbf{N}$ denote the set of natural numbers. Consider the class **Prim** of all primitive recursive functions of type $\mathbf{N} \to \mathbf{N}$. Prove that there are an infinite number of total computable functions of type $\mathbf{N} \to \mathbf{N}$ that do not belong to **Prim**. (Hint: First prove that there exists a total computable function of type $\mathbf{N} \to \mathbf{N}$ that does not belong to **Prim**.)

5. Give a sketch of proof to show that each RAM machine can be simulated by a Turing machine. (Your proof does not need to give the details of the transition function of the Turing machine.)

   The following are the RAM instructions you should use to answer this question. In these instructions, M_i, M_j, and M_k denote the i^{th}, j^{th}, and k^{th} memory cells, respectively, and m denotes the m^{th} instruction in a RAM program.

   (a) $M_i := 1$.

   (b) $M_i := M_j + M_k$.

(c) $M_i := M_j - M_k$ (if the number in M_k is greater than the number in M_j, the contents of M_i is set to 0).

(d) $M_i := \lfloor M_i/2 \rfloor$.

(e) **goto** m **if** $M_i > 0$ (if the number in M_i is greater than 0 then the next instruction to be executed is the m^{th} instruction, otherwise, execute the next instruction).

6. Which of the following properties of non-deterministic Turing machines are decidable? Justify your answer in each case. Clearly state any theorem you apply.

   (a) Given a machine M, are there an infinite number of *deterministic* Turing machines that accept the language accepted by M?

   (b) Given a machine M, is there a machine that uses at most 20 states that accepts the same language as M?

   (c) Given a machine M, does M halt in non-deterministic polynomial time?

7. (a) Let k be a natural number such that $k \geq 2$. Prove that there exists a propositional formula that uses at most $\lceil log_2 k \rceil$ propositional variables and that has exactly k satisfying truth assignments.

   (b) Prove that NP is closed under polynomial time reductions.

   (c) Let k be a natural number such that $k \geq 2$. Let $\mathbf{Prop}_k$ denote the language of propositional formulas that have at least k satisfying truth assignments. Prove that $\mathbf{Prop}_k$ is an NP-complete language.

8. Let $\mathbf{P}_1, \ldots, \mathbf{P}_n$ (where $n \geq 1$) be n *unary* predicate names. Let φ be a first-order sentence over these predicate names. Give a sketch of proof to show that there exists a first-order sentence ψ that uses exactly one first-order variable such that φ and ψ have the same models. (Hint: Observe that the sentence

$$\forall x(P_1(x) \wedge \exists y \neg P_2(y))$$

and the sentence

$$\forall x(P_1(x) \wedge \exists x \neg P_2(x))$$

have the same models.)