Theory Qualifier
January 17, 1996

Instructions

Please read these instructions and all questions C A R E F U L L Y before proceeding.

You will have $2\frac{1}{2}$ hours. The exam is closed-book. All questions are of equal weight. You must attempt to answer all of them to earn an unconditional pass. All algorithms should be accompanied by a proof of correctness and complexity analysis unless otherwise directed.

The questions will be graded according to the following criteria in order of importance:

1. correctness and completeness
2. clarity, precision, and conciseness
3. optimality of result

Partial credit will be awarded where appropriate, so be sure to show all work. Unclear or irrelevant arguments will be penalized.

GOOD LUCK!

1. (Automata and formal languages) Let $\Sigma = \{a, b, c\}$. Exactly one of the following four statements is true:

   (a) For any $A \subseteq \Sigma^*$, if A is regular, then so is $\{xx \mid x \in A\}$.

   (b) For any $A \subseteq \Sigma^*$, if A is regular, then so is $\{x \mid xx \in A\}$.

   (c) For any $A \subseteq \Sigma^*$, if A is context-free, then so is $\{xx \mid x \in A\}$.

   (d) For any $A \subseteq \Sigma^*$, if A is context-free, then so is $\{x \mid xx \in A\}$.

   Which? Give a proof and three counterexamples. (*Hint for (d).* Consider the set

$$A \quad = \quad \{a^n b^n c^m a^m b^k c^k \mid n, m, k \geq 1\} \text{ .)}$$

2. (Computability and undecidability) Let $\mathcal{F}$ be any formal deductive system that is strong enough to prove basic facts about Turing machines (Peano Arithmetic is one such system). Assume that $\mathcal{F}$ is *sound*, *i.e.* all theorems of $\mathcal{F}$ are true, and that the set of theorems of $\mathcal{F}$ is recursively enumerable. Formalize and prove the following result: there exists a total computable function $f : \mathbb{N} \to \mathbb{N}$ that grows asymptotically faster than any computable function that is provably total in $\mathcal{F}$.

MORE—PLEASE GO ON TO NEXT PAGE

3. (Complexity, completeness and reducibility) A *propositional Horn clause* is a disjunction of literals with at most one positive literal. The clause

$$\neg P_1 \vee \neg P_2 \vee \cdots \vee \neg P_n \vee Q$$

is often written

$$P_1 \wedge P_2 \wedge \cdots \wedge P_n \quad \rightarrow \quad Q \ ,$$

and the clause

$$\neg P_1 \vee \neg P_2 \vee \cdots \vee \neg P_n$$

can be written

$$P_1 \wedge P_2 \wedge \cdots \wedge P_n \quad \rightarrow \quad \perp \ ,$$

where $\perp$ denotes falsity. Any single positive literal Q is also a Horn clause. Show that the emptiness problem for context-free languages (Given a context-free grammar G, is $L(G) = \varnothing$?) reduces in linear time to the satisfiability problem for finite conjunctions of Horn clauses.

4. (Design and analysis of algorithms) Design efficient algorithms for the problems of Question 3. (Linear time is possible.) You may assume that all occurrences of each symbol P in the given system are assembled in a doubly linked list ℓ_P, and all occurrences of symbols in each clause or production C are assembled in a doubly linked list ℓ_C.

END OF EXAM