Computer Science Department
Stanford University
Comprehensive Examination in Automata, Languages, and Mathematical Theory of Computation
Autumn 1989

November 27, 1989

READ THIS FIRST!

1. You should write your answers for this part of the Comprehensive Examination in a BLUE BOOK. Be sure to write your MAGIC NUMBER on the cover of every blue book that you use.

2. Be sure you have all the pages of this exam. There are 2 pages.

3. The number of POINTS for each problem indicates how elaborate an answer is expected. For example, an essay-type question worth 6 points or less doesn't deserve an extremely detailed answer, even though a person can expound at length on just about any topic in computer science.

4. The total number of points is 60, and the exam takes 60 minutes. This "coincidence" can help you plan your time.

5. This exam is CLOSED BOOK. You may not use any notes, calculators, computers, or outside help.

6. Show your work, since PARTIAL CREDIT will be given for incomplete answers. For example, you can get credit for making a reasonable start on a problem even if the idea doesn't work out; you can also get credit for realizing that certain approaches are incorrect. On a true/false question, you might get partial credit for explaining why you think something is true when it is actually false. But no partial credit can be given if you write nothing.

Comprehensive:

Automata, Languages, and Mathematical Theory of Computation **Autumn 1989** **(60 points)**

Problem 1 (20 points). Given two strings x and y over an alphabet Σ, define *shuffle*(x, y) to be the set of all strings over Σ, which can be obtained by shuffling x and y together in an arbitrary way.

For instance, *shuffle*$(a, bc) = \{abc, bac, bca\}$.

shuffle can be defined recursively as follows:

$$shuffle(x, \epsilon) = \{x\}$$
$$shuffle(\epsilon, x) = \{x\}$$
$$shuffle(ax, by) = \{az : z \in shuffle(x, by)\} \bigcup \{bz : z \in shuffle(ax, y)\}.$$

Let L be an arbitrary regular language over Σ.

1a. Let $L_1 = \{x \in \Sigma^* : \exists y \in \Sigma^* (shuffle(x, y) \cap L \neq \phi)\}$.
Is L_1 necessarily regular?
Is it necessarily context-free?
Justify your answers briefly.

1b. Let $L_2 = \{xy : x, y \in \Sigma^* \text{ and } shuffle(x, y) \cap L \neq \phi\}$.
Is L_2 necessarily regular?
Is it necessarily context-free?
Justify your answers briefly.

Problem 2 (10 points). Assume that there is no polynomial-time program that, given a graph, prints a Hamiltonian cycle, when one exists. Prove that the decision problem "HAMILTONIAN CYCLE" (does the given graph have a Hamiltonian cycle?) is not in the class P.

Problem 3 (20 points).

3a. Give the strongest possible Hoare-style verification rule to prove the partial correctness of the statement **repeat** P **until** E (where P is a program and E an expression).

3b. Consider the following program Q to compute the quotient and remainder of a nonnegative integer x and a positive integer y:

```
{x ≥ 0 ∧ y > 0}
quot := 0; rem := x;
while {p(x, y, quot, rem)} rem ≥ y do
     begin
          i := 1; z := y;
          repeat
               quot := quot + i; rem := rem − z;
               i := i + i; z := z + z
          until {q(x, y, quot, rem, i, z)} rem < z
     end
{x = y · quot + rem ∧ 0 ≤ rem < y}.
```

Sketch a proof of the partial correctness of Q, using the rule you gave in (a). In particular, give loop invariants p and q such that all verification conditions have trivial proofs (you need not prove them).

3c. Define a variant function into a well-founded set, and sketch a proof of the termination of Q.

Problem 4 (10 points). Prove that the set of all Turing machines that accept infinitely many inputs is not r.e.

Hint:: Compare the problem of whether a Turing machine accepts infinitely many inputs with the complement of the halting problem.