THEORY I - CS 3500 D　　　Spring 2003

Final Exam - Sample Problems

---

1. Let $\Sigma = \{a, b\}$. Construct an NFA for the language represented by the regular expression $(a(a \cup b)b^*(a \cup b))^*$.

2. CFL and CFG: Identify all the violations that make this grammar one that is not in Chomsky normal form:

$$S \rightarrow BA|bB|A$$
$$A \rightarrow AS|b|\epsilon$$
$$B \rightarrow BA|a$$

3. Consider the following language $L = \{a^{2^n} | n \geq 0\}$ and answer the questions below. Provide a justification to every answer.

   a Is L a regular language ?

   b Is L context-free ?

   c Is L Turing-recognizable ?

   d Is L Turing-decidable ?

   e Describe the complement of L.

4. Consider the problems below. Justify your True or False answer in each part.

   - True or False: The complement of every context-free language is context-free.
   - True or False: The intersection of every two CFLs is context free.
   - True or False: A Turing machine with 3-tapes is more powerful (that is, accepts a larger class of languages) than a Turing machine with 1 tape.
   - True or False: The complement of every Turing-recognizable language is Turing-recognizable.
   - True or False: There exists a regular language that is not accepted by any push-down automata.
   - True or False: Every Turing machine is a decider.
   - True or False: There exists a non-context-free language that is decidable.
   - True or False: All languages over the alphabet $\{0, 1\}$ are Turing-recognizable.
   - True or False: The set of valid encodings over $\{0, 1\}$ of deterministic Turing machines is uncountable.

5. Consider the problems below. Justify your True or False answer in each part.

   - True or False: The merging of two sorted sequences, consisting of n elements each, takes $\Omega(n \log n)$ comparisons.

- True or False: Given two $n \times n$ matrices, their product can be found in $O(n \log n)$ time.
- True or False: Any depth-first-search of a graph with edge weights produces a minimum spanning forest of that graph.
- True or False: Finding the shortest distance between a vertex s and a vertex t in an undirected graph (with *no* edge weights) with n vertices takes $\Omega(n^3)$ time.
- True or False: There is an implementation of Dijkstra's shortest path algorithm that takes time $O(n \log n)$ for *sparse* graphs.
- True or False: The minimum spanning tree can be found in linear time using Kruskal's algorithm.
- True or False: The problem of deciding if an undirected graph is connected is in **NP**.
- True or False: All problems in **NP** take at least exponential time.
- True or False: No problem in **NP** is solvable in polynomial-time.
- True or False: If an **NP-Complete** problem is solvable in polynomial-time, then every problem in **NP** is solvable in polynomial-time
- True or False: $NP = co\_NP$.

6. Let G be a connected, undirected graph with non-negative integer weights. Let G have n vertices and n edges. Is it possible to find a minimum spanning tree in G in linear time? Why?

7. Suppose we use the array data structure instead of a priority queue in Prim's algorithm for computing a minimum spanning tree of a undirected graph.

   a What is the running time for Prim's algorithm using the array data structure?

   b How does this running time compare with Kruskal's algorithm for *sparse* graphs? Why?

   c How does this running time compare with Kruskal's algorithm for *dense* graphs? Why?

8. Let A be a decision problem to which all NP problems are reducible in polynomial time. Let A be reducible to a decision problem B in polynomial time. Is it true that if B is NP-Complete then A is also NP-Complete? Why?