FACULTY OF SCIENCE
FINAL EXAMINATION

COMPUTER SCIENCE COMP 330

Theoretical Aspects of Computer Science

Examiner: Prof. Prakash Panangaden
Associate Examiner: Prof. Laurie Hendren

11th December 2009
2 pm to 5 pm

Instructions:

This exam has 6 questions. Please answer all questions. The maximum score for this exam is 60. The marks for each question are indicated just after each question. This is an **open book exam**: you may use any books or notes that you have, including dictionaries. You have three hours in all. You may **not** use calculators, computers, cell phones or electronic aids of any kind. Please answer all questions **in the official answer book**. The questions appear on pages 1 and 2; this title page is not numbered. Page 3 contains a *partial* list of useful results. There are a total of four pages including this title page.

Question 1[10 points]
Consider the language of bit strings interpreted as natural numbers in binary notation. It is easy to design a DFA that accepts all strings that correspond to numbers that are **not** divisible by 3. Very likely you would come up with a machine that has two accept states. Prove that you cannot design a DFA *with only one accept state* to recognize this language.

Question 2[10 points]
Consider the language $L = a^n b^{n+m} c^m$ with $n, m \geq 0$. Classify this as *one* of the three following:

1. regular,
2. context-free but not regular,
3. recursive but not context-free.

You have to prove each asssertion. For example, if you say that it is regular, you must give an NFA to recognize it; of course, in this case it is obvious that it does not belong to the other two classes. Similarly, if you claim that it is context-free, but not regular, you have to give a context-free grammar **and a proof that it is not regular**, but, of course you will not have to prove that it is recursive. If you claim that it belongs to the last class, you must show that it is not context-free (it is then immediate that it is not regular) *and* give an algorithm to recognize it.

Question 3[10 points]
Suppose that X and Y are two RE sets. Show that there are two *disjoint* RE sets X' and Y' (so $X' \cap Y' = \emptyset$) with the property that $X \cup Y = X' \cup Y'$.

Question 4[10 points]
One of the following is decidable and the other is undecidable. Which one is which? Prove your claims. For the undecidable one you must give a proof by reduction to a known undecidable problem. For the decidable one you must give an algorithm. You may use algorithms described in class by just mentioning them by name without repeating how they work.

1. Given two context-free grammars G_1 and G_2 is $L(G_1) \cap L(G_2) = \emptyset$?
2. Given two context-free grammars G_1 and G_2 and a word w is $w \in L(G_1) \cap L(G_2)$?

Question 5[10 points]
Suppose that M is a one-tape deterministic Turing machine and suppose that w is a word that is on the tape when it starts up. Is it decidable whether the total space used by the Turing machine is more than twice the space taken up by the input word? Prove your answer.

Question 6[10 points]
Are the following statements true? No explanations are needed. In the following G always means a context-free grammar and M always means a one-tape deterministic Turing machine. We write $\langle G \rangle$ for an encoding of a CFG and $\langle M \rangle$ for an encoding of a Turing machine. Similarly we write $\langle M, w \rangle$ for the encoding of a Turing machine and a word and $\langle G, w \rangle$ for the encoding of a CFG and a word. The notation $L(G)$ means the language defined by the grammar G and $L(M)$ is the language accepted by a Turing machine M. The alphabet is Σ.

1. The set $\{\langle G \rangle | L(G) = \emptyset\}$ is RE.
2. The set $\{\langle G \rangle | L(G) = \emptyset\}$ is co-RE.
3. The set $\{\langle G \rangle | L(G) = \Sigma^*\}$ is RE.
4. The set $\{\langle G \rangle | L(G) = \Sigma^*\}$ is co-RE.
5. The set $\{\langle M \rangle | L(M) = \emptyset\}$ is RE.
6. The set $\{\langle M \rangle | L(M) = \emptyset\}$ is co-RE.
7. The set $\{\langle M \rangle | L(M) \text{ is a finite set.}\}$ is RE.
8. The set $\{\langle M \rangle | L(M) \text{ is a finite set.}\}$ is co-RE.
9. The set $\{\langle M \rangle | L(M) \text{ is a finite set.}\}$ is neither RE nor co-RE.
10. Quantum computers can solve the halting problem.

Partial List of Algorithms, Results and Theorems

1. A DFA can be converted to a minimal form using the splitting algorithm.
2. The Myhill-Nerode theorem, which implies that the minimal form is unique for DFAs.
3. An NFA with ϵ moves is equivalent to an NFA which is equivalent to a DFA.
4. Regular expressions define exactly the same languages as DFAs.
5. Equality of regular expressions is decidable, but only by going through DFAs and minimization. You cannot assume that regular expressions can be tested for equality if you are using this to show something about DFAs.
6. Regular languages are closed under the following operations: union, intersection, complement, star, quotient and concatenation.
7. The pumping lemma for regular languages.
8. Context-free languages are recognized by pushdown automata.
9. Pushdown automata cannot, in general, be made deterministic.
10. Pumping lemma for CFLs.
11. All regular languages are CFLs.
12. There is an algorithm to decide if a given word is accepted by a given grammar, the CKY dynamic programming algorithm.
13. There is an algorithm to put a grammar G into Chomsky normal form G' so that $L(G) = L(G') \cup \{\epsilon\}$.
14. The complement of a CFL may not be a CFL.
15. The intersection of two CFLs may not be a CFL.
16. The union of two CFLs is a CFL.
17. Any of the theorems on the computability handouts.
18. The Post Correspondence Problem is unsolvable.
19. It is undecidable whether $L(G) = \Sigma^*$ for a CFG G.
20. Every infinite RE set contains an infinite recursive subset.
21. Rice's theorem.
22. The recursion theorem.