FACULTY OF SCIENCE
FINAL EXAMINATION

COMPUTER SCIENCE COMP 330

Theoretical Aspects of Computer Science

Examiner: Prof. Prakash Panangaden
Associate Examiner: Prof. Patrick Hayden

8th December 2006
9 am to 12 noon

Instructions:

This exam has 6 questions. Please answer all questions. Each question is worth 10 points. 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**. You may keep the questions. For your reference, there is a list of results that you can use without proof on the last page, which is page 2. The questions appear on page 1 and this title page is not numbered.

This exam is printed on both sides (double-sided)

Question 1
If w is any word over the alphabet Σ we write w^R for w backwards. Suppose L_1 and L_2 are regular languages show that

$$L \stackrel{\text{def}}{=} \{w | \exists x \in L_1 \text{ and } y \in L_2 \text{ such that } w = xy^R\}$$

is also regular.

Question 2
Give an algorithm to decide whether the language accepted by a DFA is Σ^*.

Question 3
Show that the language $a^i b^j$ with $j = i^2$ is not context free. [Hint: if you use the pumping lemma there are 5 cases to consider, 4 of them are very easy.]

Question 4
Suppose that L is a context-free language and R is a regular language. One of the following two questions is undecidable and the other is decidable. For the decidable one give an algorithm, and for the undecidable one give a proof of undecidability. The questions are:

1. Is $L \subseteq R$?
2. Is $R \subseteq L$?

Question 5
Is it decidable whether a one-tape deterministic Turing machine with an initially blank tape ever uses more than 330 cells of its tape? Give a proof of undecidability or an algorithm to support your answer. A short sketch of the proof or algorithm is required, do not try to formalize it in detail. Recall that the number of states is finite and the tape alphabet is also finite.

Question 6
Are the following statements true? No explanations are needed.

1. The union of *infinitely many* regular languages is always regular.
2. If a set is RE, then there is an enumerator for that set that never enumerates the same element twice.
3. Primitive recursive functions cannot be enumerated by a total computable function.
4. It is undecidable whether the complement of a CFL is empty.
5. This question cannot be answered.

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 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 the language of a CFG is empty.
13. There is an algorithm to decide if a given word is accepted by a given grammar, the CKY dynamic programming algorithm.
14. There is an algorithm to put a grammar G into Chomsky normal form G' so that $L(G) = L(G') \cup \{\epsilon\}$.
15. The complement of a CFL may not be a CFL.
16. The intersection of two CFLs may not be a CFL.
17. The union of two CFLs is a CFL.
18. Turing machines are equivalent to **while** programs, to RAM machines, to machines with two stacks, to machines with two counters, to multitape Turing machines, to nondeterministic Turing machines, to multidimensional Turing machines, to Post production systems, to λ-calculus and to any of the common programming languages.
19. The halting problem for any of the above algorithmic frameworks is unsolvable.
20. The acceptance problem for Turing machines is unsolvable.
21. Any of the theorems on the computability handouts.
22. The PCP is unsolvable.
23. It is undecidable whether $L(G) = \Sigma^*$ for a CFG G.
24. Every infinite RE set contains an infinite recursive subset.
25. Rice's theorem.
26. The recursion theorem.
27. The validity problem for first-order logic is RE.
28. The truths of arithmetic are not even RE.