FACULTY OF SCIENCE
FINAL EXAMINATION

COMPUTER SCIENCE COMP 330

Theoretical Aspects of Computer Science

Examiner: Prof. Prakash Panangaden
Associate Examiner: Prof. Doina Precup

12th December 2007
2 pm to 5pm

Instructions:

This exam has 5 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**. You may keep the questions. For your reference, there is a list of results that you can use without proof on page 3 and 4. The questions appear on pages 1 and 2; this title page is not numbered. There are a total of five pages including this title page.

Question 1[10 points]
Give an algorithm to decide whether the language R accepted by a *given* DFA contains a *given* context-free language L.

Question 2[15 points]
Consider the language $L = a^{i+j}b^{i+k}c^{j+k}$ with $i, j, k \geq 0$. 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 assertion. 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[15points]
Let L be any language with alphabet Σ. The language $perms(L)$ is the language of *all* permutations of words from L. Clearly $L \subseteq perms(L)$. Which of the following claims is true? prove all your answers.

1. If L is regular then $perms(L)$ is regular.
2. If L is recursive then $perms(L)$ is recursive.
3. If L is r.e. then $perms(L)$ is r.e.

Question 4[15 points]
Let ϕ stand for an enumeration of partial recursive functions (or Turing machines, C++ programs, or whatever is your favourite formalism) in some Gödel numbering. Thus, ϕ_n stands for the nth partial recursive function in this numbering. We write $\text{dom}(f)$ for the set of values for which the function

f is defined. We write $|S|$ for the size of a set, this may be infinite in some cases. Define the following set of natural numbers

$$Q = \{n : |\mathrm{dom}(\phi_n)| = 330\}.$$

By Rice's theorem it is clearly not decidable. Show that this is not r.e. by giving an appropriate reduction.

Question 5[5 points]

Are the following statements true? No explanations are needed.

1. The intersection of two context-free languages is always a context-free language.
2. The intersection of an r.e. set and a regular set is always recursive.
3. Given L_1 and L_2 context-free languages, it is undecidable whether $L_1 \subseteq L_2$.
4. If a set is not r.e. its complement cannot be co r.e.
5. The new multi-core machines are so powerful that they can solve the halting problem in a few seconds.

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 intersection of a regular language and a CFL is a CFL.
16. The complement of a CFL 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 multi-tape 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.