Deferred exam for Software Engineering 4I03 Michael Soltys

DURATION OF EXAMINATION: 3 Hours
MCMASTER UNIVERSITY FINAL EXAMINATION April, 2004

THIS EXAMINATION PAPER INCLUDES 1 PAGE AND 6 QUESTIONS. YOU ARE RESPONSIBLE FOR ENSURING THAT YOUR COPY OF THE PAPER IS COMPLETE. BRING ANY DISCREPANCY TO THE ATTENTION OF YOUR INVIGILATOR.

No aids allowed.
All questions worth 10 marks, for a total of 60.

1. Prove that all regular languages are context-free. That is, prove that if L is a regular language, then there exists a context-free grammar G such that $L = L(G)$.

   What about the reverse direction; are all context-free languages regular? Justify your answer.

2. Define what it means for a context-free grammar to be in Chomsky Normal Form (CNF). Show how to transform a general context-free grammar into CNF. Finally, given a context-free grammar G in CNF, give an algorithm that for any input w decides whether $w \in L(G)$.

3. All Turing machines can be encoded with strings. Suppose that $\langle M_1\rangle, \langle M_2\rangle, \langle M_3\rangle, \ldots$ is one such encoding of Turing machines (thus, for any Turing machine M there exists an i such that M_i is M, and the encoding of M is $\langle M_i\rangle$). Is it possible to design a context-free grammar G which generates precisely those encodings corresponding to Turing machines that halt on all inputs? That is, $x \in L(G)$ iff $x = \langle M_i\rangle$ for some i, and M_i halts on all inputs? Justify your answer.

4. Let $L = \{w \in \{0,1\}^* | w = x111y \text{ where } x, y \in \{0,1\}^*\}$. That is, L is the set of binary strings which have three consecutive ones. Give a Turing machine M such that $L = L(M)$. Your M must halt on all inputs, in one of the following states: q_{yes} or q_{no} (input is in L or not, respectively).

5. Design a Deterministic Finite Automaton (DFA) which recognizes the language given by $L = \{w \in \{0,1\}^* |$ number of 1s in w is divisible by 5 $\}$.

6. Suppose that we allow our Turing machines to use at most 217 squares of tape. That is, the head is free to range only over squares 1 through 217, where square 1 is the square where the head is positioned when the machine is turned on. What class of languages does this restricted type of Turing machine recognize? Justify your answer.

   What if instead of limiting the number of squares of tape, we allow the Turing machine to have any number of tapes and any alphabet, but at most 10 different states? Justify your answer.

The End