UNIVERSITY OF WESTERN ONTARIO

Computer Science 331a, Fall 1996

Foundations of Theoretical Computer Science

FINAL EXAMINATION

9:00-12:00, Saturday, December 7, 1996

Name ______________________________

ID ______________________________

1. Answer the following questions briefly.

(1) What is the theoretical notion corresponding to the notion of a *computer program*?

(2) What is the theoretical notion corresponding to the notion of an *all purpose computer*?

(3) What is the class of problems that can be efficiently (i.e. in polynomial time) solved by an electronic computer?

(4) What are the relations between the following classes of languages ($=$, $\subseteq$, $\subset$, $\supseteq$, $\supset$, incomparable, unknown)?

REG CF,	DTM CF,
DTM REC,	NFA RegExp,
NFA DFA,	NPDA CF,
DPDA REG,	DPDA NPDA,
PSPACE REC,	P NP,
NPSPACE DPSPACE,	NPSPACE NPTIME.

2. Show that the intersection of two recursive languages is a recursive language.

3. Construct a context-free grammar generating the language denoted by the regular expression $a^*bb^*[a \cup b]ab^*$.

4. Given the right linear grammar

$S \longrightarrow aaA|\ bB|\ cbba$

$A \longrightarrow ab|\ B|\ abbA$

$B \longrightarrow cbc|\ aB|\ bA$

construct an equivalent finite automaton.

5. For each of the following languages, answer whether it is regular, context-free but not regular, or non-context-free. If it is regular, give a finite automaton for it. If it is context-free but not regular, give a push-down automaton that accepts it. Otherwise, just briefly justify your answer.

(a) $L = h(L_1)$ where $L_1 = \{a^n b^n c^n \mid n \geq 0\}$, and $h : \{a, b, c\}^* \longrightarrow \{a, b\}^*$ is the morphism defined by $h(a) = a$, $h(b) = b$, and $h(c) = b$.

(b) $L = L_1 \cup L_2$ where $L_1 = \{a^i b^j c^k \mid i > j > k \geq 0\}$, and $L_2 = a^* b^* c^*$.

(c) $L = L_1 \cap L_2$ where $L_1 = \{a^n b^n c^m \mid n, m \geq 0\}$, $L_2 = \{a^p b^q c^q \mid p, q \geq 0\}$.

6. Prove that the language $L = \{a^i b^j c^i d^j \mid i \geq 1, j \geq 1\}$ is not a context-free language using the Pumping Lemma for CFLs.

7. Construct a Turing machine of any type which accepts the set of strings over $\{0, 1\}$ with an equal number of 0's and 1's.

8. Are the following problems decidable or undecidable? Justify your answers.

(a) SUBWORD MEMBERSHIP PROBLEM
INPUT: A DFA $M = (Q, \Sigma, \delta, s, F)$ and a word $x \in \Sigma^*$.
QUESTION: Does there exist a word uxv in $L(M)$ for some u and v in Σ^*?

(b) DTM STATE REACHABILITY
INPUT: A DTM, $M = (Q, \Sigma, \Gamma, \delta, s, f)$, a state q in Q, and a word x in Σ^*.
QUESTION: Does M ever enter state q when started on input word x?

If the problem is decidable, describe the procedure (algorithm, program, decision-making Turing machine) that answers the question for any given input. If the problem is undecidable, justify why it is so.