YORK UNIVERSITY

FACULTY OF SCIENCE AND ENGINEERING

2009 FALL TERM EXAMINATION

Course Number: CSE2001

Title: Introduction to Theory of Computation

Duration: 3 hours

No aids allowed.

- There should be 11 pages in the exam, including this page.
- Write all answers on the examination paper. If your answer does not fit in the space provided, you can continue your answer on the back of a page or on page 11, indicating clearly that you have done so.
- You may use Church's Thesis in your answer to any question.
- Write legibly.

Name ____

(Please underline your family name.)

Student Number ________________________________

1.	________________/7
2.	________________/10
3.	________________/3
4.	________________/2
5.	________________/2
6.	________________/4
7.	________________/4
8.	________________/4
9.	________________/4
10.	________________/4
11.	________________/4
12.	________________/2
Total:	________________/50

1. [7 marks] For each of the following languages, you must determine whether the language is regular, context-free, decidable, recognizable or not recognizable. For each language, circle the *leftmost* correct answer. For example, if a language is both recognizable and decidable, but not context-free, circle decidable.

(a) $\{w : w \in \{0,1\}^* \text{ and } w = w^R\}$

regular context-free decidable recognizable not recognizable

(b) $\{\langle M, w \rangle : M \text{ is a Turing machine and } w \notin L(M)\}$

regular context-free decidable recognizable not recognizable

(c) $\{\langle M, w \rangle : M$ is a Turing machine that accepts input string w in fewer than 2009 steps$\}$

regular context-free decidable recognizable not recognizable

(d) $\{\langle M, w \rangle : M$ is a Turing machine that accepts input string w after taking more than 2009 steps$\}$

regular context-free decidable recognizable not recognizable

(e) The set of all binary strings in which each 0 is immediately followed by a 1

regular context-free decidable recognizable not recognizable

(f) $\{0^i 1^j 2^k : i + j = k\}$

regular context-free decidable recognizable not recognizable

(g) $\{0^p : p \text{ is a prime number}\}$

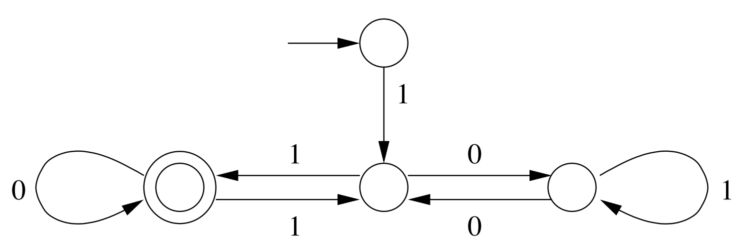
regular context-free decidable recognizable not recognizable

2. [10 marks] Recall that $B(n)$ is the binary representation of the natural number n with no leading 0's. (For example, $B(25) = 11001$.) For each of the following parts, you do not have to prove your answer is correct.

[3] **(a)** Draw the transition diagram of a DFA for $L_1 = \{B(n) : n \text{ is a positive, even integer}\}$.

[4] **(b)** Let $L_2 = \{B(n)\#(B(2n))^R : n \geq 1\}$. (For example, 1011#01101 is in L_2 because 1101 = $B(13)$ and 01011 is the reverse of $B(26)$.) Draw the transition diagram of a pushdown automaton for L_2.

[3] **(c)** A deterministic finite automaton for $L_3 = \{B(n) : n \text{ is a positive multiple of } 3\}$ is shown below. Give a regular expression for L_3.



3. [3 marks] Let $L_4 \subseteq \{0,1\}^*$ be the set of all palindromes whose first character is 1. Give a context-free grammar for L_4. (Do not prove your answer is correct.)

4. [2 marks] Consider the context-free grammar G with a single variable S and three rules, $S \to 0S1$, $S \to S1$ and $S \to \varepsilon$. Suppose you want to prove that G is a grammar for the language $L_5 = \{0^i 1^j : i, j \in \mathbb{N} \text{ and } i \leq j\}$. You could do this by proving *two* of the following claims. Which two?

1. For all $n \geq 0$, G generates some string in L_5 that has length n.
2. For all $n \geq 1$, for all i and j, if G generates $0^i 1^j$ in n steps, then $i \leq j$.
3. For all $n \geq 1$, for all strings $x \in \{0,1\}^*$, if G generates x in n steps, then x is of the form $0^i 1^j$, where $i \leq j$.
4. For all $n \geq 1$, every string in L_5 has an n-step derivation using G.
5. For all $n \geq 0$, every string in L_5 of length n is generated by G.

5. [2 marks] Suppose you would like to prove that some language L_6 is recognizable but not decidable. You could do this by proving *two* of the following statements. Which two?

1. $\overline{A_{TM}} \leq_m L_6$
2. $L_6 \leq_m A_{TM}$
3. $L_6 \leq_m \overline{A_{TM}}$
4. $\overline{E_{TM}} = \{\langle M \rangle : M \text{ is a Turing machine with } L(M) \neq \emptyset\} \leq_m L_6$
5. $\overline{L_6}$ is decidable

6. [4 marks] Is it true that every context-free grammar in Chomsky normal form is unambiguous? Circle the correct answer and then prove your answer is correct.

YES NO

7. [4 marks] Prove that the language $L_7 = \{q\#r : q, r \in \{0,1\}^* \text{ and } r \text{ contains } q \text{ as a substring}\}$ is not regular.

8. [4 marks] Consider the grammar with a single variable S and two rules, $S \to 000S11$ and $S \to \varepsilon$. Give a careful proof that every string in $L_8 = \{0^i 1^j : i, j \in \mathbb{N} \text{ and } 2i = 3j\}$ is generated by the grammar.

If you use a proof by induction, you must state the claim that you are proving and the variable that you are doing induction on.

9. [4 marks] If L is a language, let $\textsc{Suffixes}(L) = \{y : \text{there exists a string } x \text{ such that } xy \in L\}$. For example, if $L = \{001, 101, 00\}$, then $\textsc{Suffixes}(L) = \{001, 01, 1, \varepsilon, 101, 00, 0\}$. Show that, for every regular language L, the language $\textsc{Suffixes}(L)$ is regular.

10. [4 marks] Let $L_{10} = \{\langle M \rangle : M \text{ is a Turing machine that accepts the input string } 101\}$. Prove that L_{10} is undecidable.

11. [4 marks] Let $L_{11} = \{\langle G_1, G_2\rangle : G_1 \text{ and } G_2 \text{ are CFG's such that } L(G_1) \neq L(G_2)\}$. Give an algorithm that recognizes L_{11}.

12. [2 marks] After passing CSE2001 and then graduating, you get a job on a large software project. Dozens of people have been working on this large programme for 25 years and there are over 12 million lines of code. Your boss suggests that some of this code is obsolete because it can never be executed. (For example, in the following chunk of code, the line marked with a star can never be executed.)

```
  if (a != 7 or b == 3) then subroutineA(a, b);
  else if (b != 4) then subroutineB(a, b);
  else if (a * b != 28) then
*     subroutineC(a, b);
```

It would simplify the project if the pieces of code that are never executed were removed from the large programme. So your boss asks you to design a programme that will take, as its input, the large computer programme and find all the lines of code that can never be executed. What is the best way for you to respond (to avoid getting fired for incompetence)?

This (nearly) blank page is for extra work space.