University of Toronto, Faculty of Arts and Science

December 2013 EXAMINATIONS

CSC236H1S

Professor Azadeh Farzan

Duration: 3 hours

Name: ______________________

Student Number: ______________________

Session (registered): ____ Morning ____ Evening ____

Read the following before you start to work.

- Write your name and student number. Please write down your complete name (first name followed by last name) as it appears on the university records. Circle the session in which you are registred.
- This is a closed book exam. You are allowed a double-sided **handwritten** 8.5 × 11 sheet of paper.
- Reminder: you need to get a mark of 35% or higher in this exam (19.6 marks in this case) to pass this course, regardless of your marks for the rest of the coursework
- You should have 18 pages including 7 problems. Do all work in the space provided. Ask the proctor if you need more paper. You can find The statement of the Master Theorem on the last page.

Problem (1)	/8
Problem (2)	/5
Problem (3)	/8
Problem (4)	/8
Problem (5)	/10
Problem (6)	/5
Problem (7)	/12
Total	/56

Problem 1 Determine the running time of the following algorithm:

```
Woof(n)
    If (n ≥ 1)
        Woof(⌊n/3⌋)
        i = 1
        while (i < n)
            print "woof"
            i = i + 3
        Woof(⌊n/3⌋)
```

by doing the following:

(a) (5 points) Write a recursive definition for the runtime function for this algorithm. Briefly justify your recursive definition.

(b) (3 points) Use the Master Theorem to solve your recursive function from (a) and find the asymptotic running time of the algorithm.

Continue with the solution of the problem, if you need more space ...

Problem 2 (5 points) Prove, using closure properties of regular languages, that if L_1 and L_2 are regular, then so is $L' = \{xy |\ x \in L_1 \ \wedge \ (y \in L_2 \vee y \notin L_1)\}$. Note that proofs that are not based on closure properties of regular languages will get no credit.

Continue with the solution of the problem, if you need more space ...

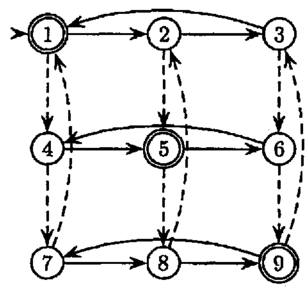
Problem 3 Consider the language L of all strings over the alphabet $\{0, 1\}$ that end in 000, and other than at the very end, contain no occurrence of three consecutive 0's. For example $001000 \in L$, but $0000 \notin L$ and $0100 \notin L$.

(a) (4 points) Write a regular expression r such that $L = L(r)$. Explain why your regular expression is correct by explaining what the various parts of it represent.

(b) (4 points) Propose a nondeterministic finite automaton N that accepts L (Note: you can attempt to give a DFA instead, but the DFA is going to be huge! And, you will only get partial marks for doing all the tedious work of drawing the huge DFA).

Continue with the solution of the problem, if you need more space ...

Problem 4 Consider the following DFA. Solid edges have label a and dashed edges have label b.



(a) (2 points) Describe the language of this DFA in one sentence.

(b) (3 points) Draw a 3-state DFA that recognizes the same language as the DFA above.

(c) (3 points) Write state invariants for your DFA from part (b) (no need to prove anything correct; just make sure your invariants are correct).

Continue with the solution of the problem, if you need more space ...

Problem 5 Consider an array A with integer elements. The following algorithm recursively finds and returns the smallest element in $A[b \dots e]$.

```
Min(A, b, e)
    If (b = e)
        return A[b]
    m = ⌊(b + e)/2⌋
    x = Min(A, b, m)
    y = Min(A, m + 1, e)
    If (x < y)
        return x
    else
        return y
```

(a) (3 points) Write the appropriate precondition and postcondition that specify the correctness of this function.

(b) (7 points) Prove that the function is correct by showing that the your specification from part (a) is inductive. Don't forget about termination.

Continue with the solution of the problem, if you need more space ...

Continue with the solution of the problem, if you need more space ...

Problem 6 (5 points) Consider the following program and the given precondition. Prove that if the precondition holds at the beginning of the program, then the loop will always terminate.

PreCondition: $x, y \in \mathbb{Z}$, $z \in \mathbb{N}$

```
While (x ≠ y ∧ z ≥ 0)
    If (x > y)
        x = x − z
    else
        y = y − z
    z = z − 1
```

Continue with the solution of the problem, if you need more space ...

Problem 7 Consider an array A. The following algorithm computes the average value of the elements of the array (in a non-standard way).

```
algorithm Average(A)
    l =length(A)
    If (l = 1)
        return A[0]
    a = (A[0] + A[l − 1])
    b = 1
    e = l − 2
    while (b < e)
        a = a + A[b] + A[e]
        b = b + 1
        e = e − 1
    If (b ≠ e)
        return a/l
    else
        return (a + A[b])/l
```

(a) (2 points) Write the appropriate precondition and postcondition that specify the correctness of this function.

(b) (2 points) Write the appropriate loop invariant that you can use to prove this function correct.

(c) (8 points) Prove that the function is correct by showing that the your specification from parts (a) and (b) are inductive. Don't forget about termination.

Continue with the solution of the problem, if you need more space ...

Continue with the solution of the problem, if you need more space ...

Master Theorem:

$$T(n) = aT(n/b) + f(n), \text{ where } a \geq 1 \text{ and } b > 1$$

- If $c < \log_b a$ and $f(n) = \Theta(n^c)$ then $T(n) = \Theta(n^{\log_b a})$.
- If $c > \log_b a$ and $f(n) = \Theta(n^c)$ then $T(n) = \Theta(f(n))$.
- If $c = \log_b a$ and $f(n) = \Theta(n^c \log^k n)$ then $T(n) = \Theta(n^c \log^{k+1} n)$.