UNIVERSITY OF TORONTO
Faculty of Arts and Science

AUGUST EXAMINATIONS 1998

CSC 238F
Instructor: Eric Ruppert

Duration - 3 hours

Examination Aids: One sheet of paper measuring $8\frac{1}{2} \times 11$ inches, handwritten on both sides. Typed or photocopied sheets are prohibited.

- There should be 9 pages in the exam, including this page.
- Do all 11 problems.
- Write all answers on the examination paper.
- Write legibly.
- Concise, well thought-out answers will be rewarded more than long, rambling answers.

Name ______________________________

(Please underline your family name.)

Student Number ______________________________

1.	________/3
2.	________/3
3.	________/6
4.	________/10
5.	________/6
6.	________/3
7.	________/14
8.	________/15
9.	________/8
10.	________/8
11.	________/4
Total:	________/80

1. [3 marks] Let P, Q and R be propositional variables. Write down a formula in *disjunctive normal form* that is true when *exactly one* of the three variables is true.

2. [3 marks] Let α be a propositional formula.
Let β be a contradiction.
Determine whether the following statement is true or false. Briefly justify your answer.

If $(\sim \beta) \rightarrow \alpha$ is a tautology, then α is satisfiable. TRUE FALSE

Justification:

3. [6 marks] Consider the following predicate logic symbols with their intended meanings.
The universe is $\mathbb{N} \cup \{z \mid z \text{ is a person}\}$.

predicate symbol	meaning
$L(x, y)$	"$x < y$" $(x, y \in \mathbb{N})$
$M(v)$	"v is a man"

function symbol	meaning
$g(v, w)$	"the amount of money person v gave to person w (in dollars)"

constant symbol	meaning
a	Alex
e	Eugenia
s	Steve

Translate the following statements into predicate logic formulae using the symbols above.

[3] **(a)** No man gave Eugenia less money than she gave Steve.

[3] **(b)** Every man gave Alex at least as much money as Alex gave Eugenia.

4. [10 marks] Recall the proof system for propositional logic that we used in class. It allows natural deduction, replacement of a wff by an equivalent wff and the following inference rules:

From $P \to Q$ and P	infer Q	Modus Ponens
From $P \vee Q$ and $P \to R$ and $Q \to R$	infer R	Proof by Cases
From $P \wedge Q$	infer P	Conjunctive Simplification
From P and Q	infer $P \wedge Q$	Conjunctive Amplification
From $P \vee Q$ and $\sim Q$	infer P	Disjunctive Simplification
From P	infer $P \vee Q$	Disjunctive Amplification
From $P \to Q$ and $Q \to P$	infer $P \leftrightarrow Q$	Biconditional Amplification
From $P \leftrightarrow Q$	infer $(P \to Q) \wedge (Q \to P)$	Biconditional Simplification

Use this proof system to give a formal proof of

$$(P \leftrightarrow Q) \to [(P \wedge R) \to (R \wedge Q)].$$

5. [6 marks] Use the definition of Θ to prove that $7n(\log_2 n)^2 + n \notin \Theta(n \log_2 n)$.

6. [3 marks] Let $T : \mathbb{N} \rightarrow \mathbb{R}$ be the function defined by

$$\begin{aligned} T(0) &= 1, \\ T(1) &= 1, \\ T(n) &= 3T(\lfloor n/2 \rfloor) + 4T(\lceil n/2 \rceil) + 8n^3 \text{ for } n > 1. \end{aligned}$$

Which of the following statements are true? Circle the labels of all of the correct answers.

1. $T(n) \in \Omega(n^3 \log n)$
2. $T(n) \in \Omega((n+3)^3)$
3. $T(n) \in \Omega(n^{\log_2 7})$
4. $T(n) \in O(7^{\log_2 n})$
5. $T(n) \in O(n^3 \log n)$
6. $T(n) \in O(\sum_{i=1}^{n} i^2)$

7. [14 marks] Let $A[1..n]$ be an array of non-zero integers. The following algorithm computes the *length* of the longest subarray whose elements all have the same sign. For example, if A contains the elements $[1, -2, -3, 6, 5, 3, 2, -8, -7, -1, 7, -9]$, then the algorithm outputs 4, since the longest block of elements of the same sign is [6, 5, 3, 2].

```
LongestBlock(A)
% precondition: n ≥ 1 and A[1..n] is an array of n non-zero integers.
    i ← 1
    maxSoFar ← 1
    currentLength ← 1
    loop
        exit when i = n
        i ← i + 1
        if A[i − 1] · A[i] > 0 then
            currentLength ← currentLength +1
            maxSoFar ← max(maxSoFar, currentLength)
        else
            currentLength ← 1
        end if
    end loop
    return maxSoFar
    % postcondition: returns length of longest subarray of A whose elements all have the same sign.
end
```

[4] **(a)** Show that the statement "The elements of A[1..currentLength] all have the same sign" is *not* always true at the beginning of every iteration of the loop.

[10] **(b)** Suppose the following four statements are true at the beginning of some iteration of the loop.

(1) $1 \leq \text{currentLength} \leq i < n$

(2) The elements of $A[i - \text{currentLength} + 1..i]$ all have the same sign.

(3) Either $i = \text{currentLength}$ or $A[i - \text{currentLength}]$ and $A[i - \text{currentLength} + 1]$ have opposite signs.

(4) maxSoFar is the length of the longest subarray of $A[1..i]$ whose elements all have the same sign.

Prove that statement (3) is also true at the end of the iteration. (Use i' and $\text{currentLength}'$ to denote the values of the variables at the end of the iteration.)

8. [15 marks] A palindrome is a string that reads the same backwards and forwards. For example, "kayak" and "toot" are palindromes. The following *recursive* algorithm determines whether part of a given string is a palindrome. The input string is represented as an array of characters.

```
palindrome(A[1..m]: array of characters, f : integer, l : integer)
% precondition: 1 ≤ f ≤ l + 1 ≤ m + 1
    if f = l + 1 or f = l then
        return "yes"
    else if A[f] = A[l] then
        return palindrome(A, f + 1, l − 1)
    else
        return "no"
    end if
% postcondition: The algorithm returns "yes" if A[f..l] is a palindrome (i.e. if A[f + i] = A[l − i] for
% all i satisfying 0 ≤ i ≤ l − f). Otherwise, the algorithm returns "no."
end
```

[12] **(a)** Prove that the palindrome algorithm is correct with respect to the precondition and postcondition. Hint: use strong induction on $n = l - f + 1$, the size of the subarray being tested.

[3] **(b)** Let $T(n)$ be the worst-case running time of the palindrome algorithm for inputs where $l - f + 1 = n$. State a recurrence that defines $T(n)$ for all $n \in \mathbb{N}$. Do *not* solve the recurrence.

9. [8 marks] Solve the following recurrence. Prove that your answer is correct by induction.

$$\begin{aligned} g(0) &= 0, \\ g(n) &= g(n-1) + 2n + 4 \text{ for } n > 0. \end{aligned}$$

10. [8 marks] Design a finite automaton that accepts the language

$$\{w \in \{0,1\}^* \mid \text{ the number of 0's in } w \text{ is even and the number of 1's in } w \text{ is also even}\}.$$

For full marks, your automaton should have at most 4 states. Briefly explain why your automaton is correct.

Is your automaton deterministic? Circle the correct answer. YES NO

11. [4 marks] Write down a regular expression for the language accepted by the non-deterministic finite automaton shown below. Try to make your regular expression as simple as possible. Do *not* prove that your answer is correct.