UNIVERSITY OF TORONTO
Faculty of Arts and Science

DECEMBER EXAMINATIONS 1995

CSC 238F
Prof. Charles Rackoff   Erindale Campus

Duration — 3 hours

Examination Aids: One 8-1/2 by 11 inch "crib" sheet.

1. [10 marks] Define the function f as follows.

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

Use induction to prove that for all $n \geq 1$,

$$f(n) \geq 2^{n-1}.$$

2. [10 marks] Recall that if $f, g : \mathbb{N} \to \mathbb{R}^+$ (where $\mathbb{R}^+$ is the set of nonnegative reals), then we say that:

- f is $\mathcal{O}(g)$ if for some $c \in \mathbb{R}^+$ and some $n_0 \in \mathbb{N}$, $f(n) \leq c\,g(n)$ for all $n \geq n_0$;
- f is $\Omega(g)$ if for some non-zero $c \in \mathbb{R}^+$ and some $n_0 \in \mathbb{N}$, $f(n) \geq c\,g(n)$ for all $n \geq n_0$;
- f is $\Theta(g)$ if f is $\mathcal{O}(g)$ and f is $\Omega(g)$.

Using these definitions, show that $f = \Theta(g) \iff [f = \mathcal{O}(g) \text{ and } g = \mathcal{O}(f)]$.

3. [8 marks] Give a big-Oh estimate for each of the following functions. For the function g in your estimate ($f = \mathcal{O}(g)$), use a simple function g of the smallest order.

(a) $n \log(n^2 + 1) + n^2 \log n$

(b) $(n \log n + 1)^2 + (\log n + 1)(n^2 + 1)$

(c) $((\log n^2)^2)^2 + n$

(d) $(2^n + n)/(2^{n/2} + 3)$

4. [10 marks] Write an expression in the predicate calculus to represent each of the following sentences.

(a) Every integer greater than or equal to 2 is either a prime number or has a prime factor smaller than itself.

(b) There is a prime number that is 2 greater than the product of two other prime numbers. (*e.g.*, $17 = (3 \times 5) + 2$.)

Assume that the Domain is the set of integers, and that the following symbols exist and are interpreted as follows.

predicates:	$\text{PRIME}(x)$	signifies	"x is a prime"
	$\text{EQ}(x, y)$	signifies	"$x = y$"
	$\text{GE}(x, y)$	signifies	"$x \geq y$"
	$\text{MULT}(x, y, z)$	signifies	$x \times y = z$
	$\text{SUM}(x, y, z)$	signifies	$x + y = z$
constant:	2		

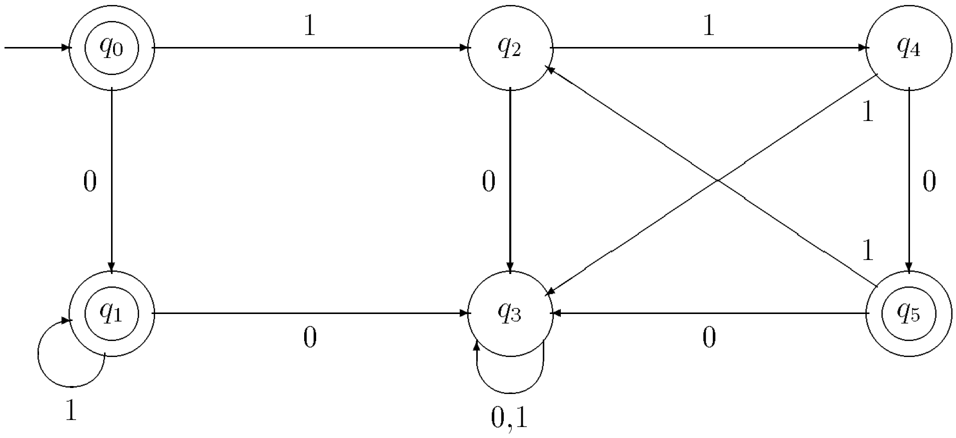
5. [15 marks] We say that a propositional formula F over the variables $\{p_1, \ldots, p_n\}$ is *monotone* if changing the truth values of some propositional variable(s) from 0 to 1 can not change the value of F from 1 to 0. For example, the formula $(p_1 \wedge p_2 \wedge p_3)$ is monotone since its value is 1 only when all variables have the truth value 1. On the other hand, the formula $\neg(p_1 \wedge p_2 \wedge p_3)$ is <u>not</u> monotone since, when $p_1 = 1$, $p_2 = 1$, $p_3 = 0$ the formula has value 1, but when $p_1 = 1$, $p_2 = 1$, $p_3 = 1$ the value of the formula becomes 0.

(a) Show that any propositional formula which uses only the connectives $\wedge$ and $\vee$ is monotone. (HINT: Use induction on the structure of the formula.)

(b) Show that the formula $p \to q$ is not monotone.

(c) Prove that $\{\wedge, \vee\}$ is not a complete set of connectives. (You can use the results of parts (a) and (b) without proving them.)

6. [10 marks] Give a regular expression for the language (a subset of $\{0,1\}^*$) recognized by the following F.S.A. Briefly explain why your regular expression works.



7. [10 marks] Give a F.S.A. that accepts the following language over the alphabet $\{0,1\}$:

$$(0|1)(01^*)^*$$

(The fewer the states, the better the mark. Briefly explain why your F.S.A. recognizes the regular expression.)

8. [10 marks] Consider the following recurrence:

$$T(n) \le \begin{cases} 1 & \text{if } n = 1, \\ 10\,T(\lceil n/4 \rceil) + n2^n & \text{if } n > 1. \end{cases}$$

Prove that $T(n)$ is $\mathcal{O}(n2^n)$.

9. [17 marks] In this question, we examine two different procedures for finding the smallest value (*mini*) and the largest value (*maxi*) in a vector that has n elements.

```
PROC1(vect, n, mini, maxi);
{ Sets mini to be the smallest element and maxi
  to be the largest element in vect[1...n]. }
   mini := vect[1];
   maxi := vect[1];
   for i := 1 to n do begin
      if vect[i] < mini then mini := vect[i];
      if vect[i] > maxi then maxi := vect[i];
   end for
END PROC1
```

```
PROC2(vect, low, high, mini, maxi);
{ Sets mini to be the smallest element and maxi
  to be the largest element in vect[low...high]. }
   if low = high then do
      mini := vect[low];
      maxi := vect[high];
   else do
      mid := ⌊(low + high)/2⌋;
      PROC2(vect,low,mid, min1,max1);
      PROC2(vect,mid+1,high, min2,max2);
      mini := MIN(min1,min2);
      maxi := MAX(max1,max2);
   end if
END PROC2
```

(a) You may assume that PROC1 correctly determines *mini* and *maxi* when *vect* is a vector of n numbers and the procedure is called with "PROC1($vect, n, mini, maxi$)". Give a big-Oh expression for the running time of PROC1, as a function of n. Briefly justify.

(b) Prove that PROC2 correctly determines *mini* and *maxi* when *vect* is a vector of n numbers and the procedure is called with "PROC2($vect, 1, n, mini, maxi$)".

(c) State a recurrence relation that expresses the running time of PROC2.

(d) Give, but do not prove, a big-Oh expression for the running time of PROC2, as a function of n.

Total Marks = 100