IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2007

ISE PART II: MEng, BEng and ACGI

CORRECTED COPY

DISCRETE MATHEMATICS AND COMPUTATIONAL COMPLEXITY

Monday, 21 May 2:00 pm

Time allowed: 2:00 hours

There are FOUR questions on this paper.

Q1 is compulsory.
Answer Q1 and any two of questions 2-4.
Q1 carries 40% of the marks. Questions 2 to 4 carry equal marks (30% each).

Any special instructions for invigilators and information for candidates are on page 1.

Examiners responsible	First Marker(s) :	G.A. Constantinides, G.A. Constantinides
	Second Marker(s) :	T.J.W. Clarke, T.J.W. Clarke

NOTATION

The following notation is used throughout this paper:

$\mathbb{R}$: The set of real numbers.

$\mathbb{Z}$: The set of integers.

$\mathbb{Z}_+$: The set of positive integers.

$\mathbb{C}$: The set of complex numbers.

$\mathbb{N}$: The set of natural numbers.

$\mathscr{P}(S)$: The power set of set S.

The Questions

1. **[Compulsory]**

a) For the sets $S_1 = \{1,2\}$, $S_2 = \{2,3\}$, list the elements of

i) $S_1 \cup S_2$,

ii) $S_1 \cap S_2$,

iii) $S_1 - S_2$,

iv) $S_1 \times S_2$,

v) $\mathscr{P}(S_1)$.

[7]

b) Consider the relation $R = \{(1,2),(2,3),(3,4)\}$ on the set $\mathbb{R}$.

i) Let R be a function from A to B. Find the smallest cardinality A and B for this to be possible.

ii) List the elements of $R \cdot R$.

iii) List the elements of the transitive closure of R.

iv) Draw the digraph of R.

v) How many functions are there from the set R to itself?

[9]

c) Express each of the following statements using appropriate logical syntax. You should take the set of complex numbers as the universe of discourse, and make use of a predicate $P(x)$, meaning 'x is a real number'.

i) Every real number, when squared, gives a non-negative real number.

ii) Every quadratic equation with complex coefficients has two complex roots.

iii) There are two distinct real numbers that are solutions to $x^2 - 1 = 0$.

[9]

d) i) Consider two functions $f(x)$ and $g(x)$. $f(x)$ is known to be $\Theta(x^2)$ and $g(x)$ is known to be $\Theta(x)$. Find functions $p(x)$ and $q(x)$ such that $f(x)+g(x)$ is $O(p(x))$ and $f(x)g(x)$ is $O(q(x))$.

ii) Write some pseudo-code for a function `fun1(x)` that has worst-case execution time $O(g(x))$ and for a function `fun2(x)` that has worst-case execution time $O(f(x))$.

iii) Show that if $r(x)$ is $\Theta(s(x))$ then $s(x)$ is $\Theta(r(x))$.

[9]

e) Write some pseudo-code for a function whose worst-case execution time satisfies $f(n) = 3f(n/2) + n$ whenever n is an even number. Find a big-O expression for the worst-case execution time of this function.

[6]

2. a) Consider the function $f : A \rightarrow \mathbb{C}$ defined by $f(x) = \frac{e^{jx}}{1-x}$, where $j = \sqrt{-1}$.

i) If $A = \mathbb{R} - K$, what is the smallest K, in the sense that if $A = \mathbb{R} - K'$ then $K \subseteq K'$?

ii) Show that for this choice of K, the function f is an injection.

iii) Show that f is not a surjection for this choice of K.

[10]

b) i) Show that the transitive closure of a relation R is equal to its connectivity relation R^*. You may assume that for an arbitrary relation Q, (i) Q is transitive iff Q^n is transitive for all positive integers n, (ii) Q is transitive iff $Q^n \subseteq Q$ for all positive integers n.

ii) Consider the function $g : S \rightarrow S$ defined by $g(x) = \lfloor \sqrt{x} \rfloor$. For $S = \{1, 2, ..., 16\}$, list the elements of $g \cdot g$ and the transitive closure g^* of g.

[20]

3. This question uses predicate logic to describe the behaviour of the simple circuit shown in Figure 3.1. Let the universe of discourse, corresponding to the set of clock periods, be $\mathbb{N}$. Each wire $i \in \{1,2\}$ is associated with a predicate $P_i(t)$. A logic-0 is present on a wire at a particular cycle t if the corresponding proposition $P_i(t)$ is false, and a logic-1 is present if the corresponding proposition $P_i(t)$ is true.

An axiom describing the function of the D-type flip-flop is given in equation (3.1).

$$\neg P_1(0) \wedge \forall t(P_1(t+1) \leftrightarrow P_2(t)). \tag{3.1}$$

a) Write a corresponding axiom for the inverter.

[4]

b) Write a proposition corresponding to the English sentence 'the inverter output at cycle 1 will have the opposite logical value to the inverter output at cycle 0'.

[5]

c) From the inverter and the flip-flop axioms, formally derive your proposition as conclusion. State the rule of inference used at each step in your working.

[15]

d) Show further that the conclusion $P_1(1)$ can be reached. State the rule of rule of inference used at each step in your working.

[6]

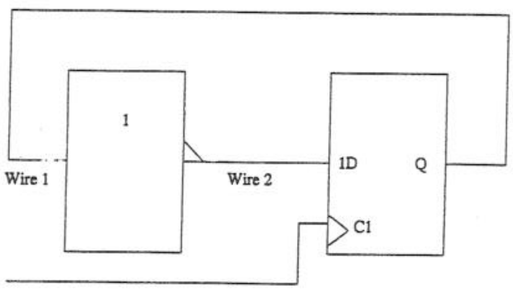


Figure 3.1 A circuit

4. This question is about multiplying two n-bit numbers, where n is a power of two, using only addition and shift arithmetic operations, each of which takes $\Theta(n)$ time for n bits. We shall use the operator '<<' to denote left-shift, *i.e.* $x << y$ means 'x left-shifted by y bits'. We shall also represent n-bit numbers by binary arrays of length n, where the most-sigificant bit is element $n-1$ and the least significant bit is element 0.

a) A possible multiplication algorithm is shown in Figure 4.1. Derive a big-Θ expression for the execution time of this algorithm.

[7]

b) Let us denote the least-significant $n/2$ bits of A by A_L and the most-significant $n/2$ bits by A_H, and similarly for B, so that $A = 2^{n/2}A_H + A_L$ and $B = 2^{n/2}B_H + B_L$. Notice that $AB = 2^n(A_H B_H) + 2^{n/2}\{(A_L + A_H)(B_L + B_H) - A_H B_H - A_L B_L\} + A_L B_L$. Use this observation to propose a recursive multiplication algorithm.

[8]

c) State the Master Theorem.

[8]

d) Derive a big-O expression for the recursive execution time, and comment on the result.

[7]

```
multiply( binary A[n], binary B[n] )
begin
    result := 0
    for i := n − 1 downto 0
        if( A[i] = 1 ) then
            result := (result << 1) + B
        else
            result := (result << 1)
end
```

Figure 4.1 An algorithm for multiplying two numbers

Discrete maths & Computational Complexity 1/14

Q1

(NEW COMPUTED EXAMPLE) E2.17/ E3.20

master - 16/4/07.

a) (i) $S_1 \cup S_2 = \{1, 2, 3\}$

(ii) $S_1 \cap S_2 = \{2\}$

(iii) $S_1 - S_2 = \{1\}$

(iv) $S_1 \times S_2 = \{(1,2), (1,3), (2,2), (2,3)\}$

(v) $P(S_1) = \{\phi, \{1\}, \{2\}, \{1,2\}\}$.

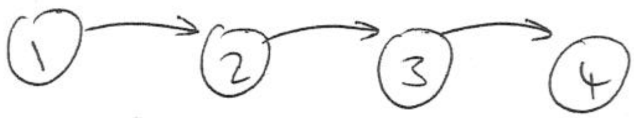
[7]

b) (i) $A = \{1, 2, 3\}, \quad B = \{2, 3, 4\}$

(ii) $R \circ R = \{(1,3), (2,4)\}$

(iii) $R^* = \{(1,2), (1,3), (1,4), (2,3), (2,4), (3,4)\}$.

(iv)



(v) $|R| = 3 \qquad |R|^{|R|} = 3^3 = 27$

[9]

(NEW COMPUTED EXAMPLE)

c) i) $\forall x \left(P(x) \rightarrow (x^2 \geq 0) \right)$

ii) $\forall a \forall b \forall c \exists x_1 \exists x_2 \left(a x_1^2 + b x_1 + c = 0 \wedge a x_2^2 + b x_2 + c = 0 \right)$

iii) $\exists x_1 \exists x_2 \left(x_1 \neq x_2 \wedge P(x_1) \wedge P(x_2) \wedge x_1^2 - 1 = 0 \wedge x_2^2 - 1 = 0 \right)$

[9]

d) i) $p(x) = x^2$ (say) (NEW COMPUTED EXAMPLE)

$q(x) = x^3$ (say)

ii)

```
fun1 (x)
{  total := 0
   for i = 1 to x
       for j = 1 to x
             total := total + 1
}
```

```
fun2 (x)
{
   total := 0
   for i = 1 to x
       total := total + 1
}
```

(iii) $r(x)$ is $\Theta(s(x))$

$\exists c_1, c_2, K$ s.t.

$\Rightarrow c_1|s(x)| \leq |r(x)| \leq c_2|s(x)|$

when $x \geq K$

$c_1|s(x)| \leq |r(x)|$

$\Rightarrow |s(x)| \leq \frac{1}{c_1}|r(x)|$ (as c_1 +ve)

$\Rightarrow$ $s(x)$ is $O(r(x))$ ($c = \frac{1}{c_1}$, K as before)

$c_2|s(x)| \geq |r(x)|$

$\Rightarrow |s(x)| \geq \frac{1}{c_2}|r(x)|$ ($c = \frac{1}{c_2}$, K as before)

$\Rightarrow$ $s(x)$ is $\Omega(r(x))$

$\therefore$ $s(x)$ is $\Theta(r(x))$. [9]

(NEW COMPUTED EXAMPLE)

e)

```
fun(n: integer)
{ total := 0;
  for i = 1 to 3
      fun(⌊n/2⌋)

  for i = 1 to n
      total := total + 1;
}
```

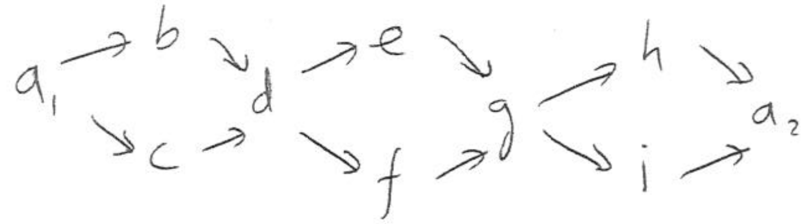
(NEW COMPUTED EXAMPLE)

[6]

c.f. $f(n) = af(n/b) + cn^d$ $\quad a > b^d$

$a=3,\ b=2,\ c=1,\ d=1$ $\Rightarrow \underline{\underline{O(n^{\log_2 3})}}$

Q2

a)(i) 

reach(R, a_1, a_2)

calls reach(R, b, a_2)

& reach(R, c, a_2)

reach(R, b, a_2) calls reach(R, d, a_2)

reach(R, c, a_2) calls reach(R, d, a_2)

Total of 4 subroutine calls

Thus total $= 1 + 4 + 2 \times 4 + 2^2 \times 4 - 2^2 \times 2$

~~= 29 times~~ ~~= 25 time~~

$= 21$ times

(ii) $a_1 \rightarrow b \rightarrow \cdots \rightarrow a_2$

reach(R, a_1, a_2)

calls reach(R, b, a_2)

etc.

Total of 11 subroutine calls + original call

$=$ 12 calls

[14]

(NEW COMPUTED EXAMPLE)

b) Extending the instance from fig 1.2(a) gives a digraph with $1 + 3K$ nodes and $|R| = 4K$ edges.

Total # ~~call~~ subroutine calls is

$$4 + 2 \times 4 + \dots + 2^{K-1} \times 4 - 2^{K-1} \times 2$$

$$= 4\left(1 + 2 + \dots + 2^{K-1}\right) - 2^{K}$$

$$= 4\left(\frac{1 - 2^K}{1 - 2}\right) - 2^K$$

$$= 4\left(2^K - 1\right) - 2^K$$

$$= 4\left(2^{|R|/4} - 1\right) - 2^{|R|/4}$$

is $\Omega\left(2^{|R|}\right)$

$\Rightarrow$ Exponential time.

[13]

(NEW COMPUTED EXAMPLE)

c) Two main improvements

(i) early exit

(ii) mark visited nodes

```
visited[ ] → init to false
reach( R, a1, a2 )
begin
    if (a1, a2) ∈ R then
        result := true
    else begin
        result := false
        while( result = false ) do
            select next a s.t. (a1, a) ∈ R
            if reach(R, a, a2) then
                result := true
            else
            if NOT visited[a] then
                if reach(R, a, a2) then
                    result := true
        end
    end
end
```

[GB]

c) [continued]

Total #subroutine calls is at most #nodes.

$\Rightarrow$ poly time. $(R \subseteq V \times V)$.

[13]

Q3

a) (i) Every real x has a corresponding $f(x) \in \mathbb{C}$ except 1. So $K = \{1\}$.

(ii) $f(x) = f(y)$

$$\frac{e^{jx}}{1-x} = \frac{e^{jy}}{1-y} \Rightarrow \left|\frac{e^{jx}}{1-x}\right| = \left|\frac{e^{jy}}{1-y}\right|$$

$$\Rightarrow \frac{1}{1-x} = \frac{1}{1-y} \Rightarrow 1-y = 1-x$$

$$\Rightarrow \underline{\underline{x = y}}$$

(iii) $\frac{1}{1-\pi} \in \mathbb{C}$

for $f(x) = \frac{1}{1-\pi}$

~~we~~ would require $\frac{e^{jx}}{1-x} = \frac{1}{1-\pi}$

$$\Rightarrow \frac{1}{1-x} = \frac{1}{1-\pi} \Rightarrow x = \pi.$$

But at $x = \pi$, $f(x) = \frac{j}{1-\pi} \neq \frac{1}{1-\pi}$

CONTRADICTION.

[10]

b) (i) $R^* = R \cup R^2 \cup \ldots$

We want to show R^* is the smallest transitive relation containing R.

Proof: a) $R \subseteq R^*$ directly

We must show b) R^* is transitive

c) $R^* \subseteq S$ for any transitive S s.t. $R \subseteq S$.

b) $(a,b) \in R^*$, $(b,c) \in R^*$.

R^* is set of all paths $\Rightarrow$ $(a,c) \in R^*$;

follow path $a \to b$ & then $b \to c$.

c) Since S is transitive, S^n is transitive and $S^n \subseteq S$.

$S^* = S \cup S^2 \cup \ldots$, $S^n \subseteq S$

$\Rightarrow S^* \subseteq S$

$R \subseteq S \Rightarrow R^* \subseteq S^*$

$\therefore \underline{\underline{R^* \subseteq S^* \subseteq S}}$.

(BOOKWORK)

(ii) $g = \{(1,1), (2,1), (3,1), (4,2), (5,2), (6,2), (7,2), (8,2), (9,3), (10,3), (11,3), (12,3), (13,3), (14,3), (15,3), (16,4)\}$

$g \cdot g = \{(1,1), (2,1), (3,1), (4,1), (5,1), (6,1), (7,1), (8,1), (9,1), (10,1), (11,1), (12,1), (13,1), (14,1), (15,1), (16,2)\}$

$g^* = \{(1,1), (2,1), (3,1), (4,1), (5,1), (6,1), (7,1), (8,1), (9,1), (10,1), (11,1)\ (12,1), (13,1), (14,1), (15,1), (16,1), (4,2), (5,2), (6,2), (7,2), (8,2), (16,2), (9,3), (10,3), (11,3), (12,3), (13,3), (14,3), (15,3), (16,4)\}.$

[illegible]
[20]

(NEW COMPUTED EXAMPLE)

4. a) $\forall t\,(P_2(t) \Leftrightarrow \neg P_1(t))$ ~~[3]~~ [4]

b) $P_2(1) \Leftrightarrow \neg P_2(0)$ ~~[3]~~ [5]

c) $\forall t\,(P_1(t+1) \Leftrightarrow P_2(t))$

(Simplification from 4.1)

$P_1(1) \Leftrightarrow P_2(0)$ (*)

(Universal instantiation)

$P_2(1) \Leftrightarrow \neg P_1(1)$ (+)

(Universal instantiation of (b))

$P_2(1) \rightarrow \neg P_1(1)$

(Simplification)

$P_2(0) \rightarrow P_1(1)$

(Simplification of *)

$\neg P_1(1) \rightarrow \neg P_2(0)$ [Contrapositive]

$P_2(1) \rightarrow \neg P_2(0)$

(Hypothetical Syllogism)

Also $P_1(1) \rightarrow P_2(0)$ (Simplification of *)

$\neg P_2(0) \rightarrow \neg P_1(1)$ [Contrapositive]

$\neg P_1(1) \rightarrow P_2(1)$ (Simplification of +)

$\neg P_2(0) \rightarrow P_2(1)$ (Hypothetical syllogism)

~~[10]~~ [15]

d) We have

$$P_2(1) \leftrightarrow \neg P_2(0)$$

and $\neg P_1(0)$ (simplification from 4.1)

The latter gives $P_2(0)$

(Universal instant + modus ponens)

Thus from (c) we get $\neg P_2(1)$

(modus tollens)

Finally, we obtain $P_1(1)$

(Universal instant + modus tollens)

~~[4]~~

[6]

Q5 a) "for" loop executes $\Theta(n)$ times.

Assuming tests take $\Theta(1)$ time, each iteration is $\Theta(n)$.

Total is $\Theta(n) \times \Theta(n) + \Theta(1)$

$= \underline{\underline{\Theta(n^2)}}$.

~~[8]~~ [7]

(NEW COMPUTED EXAMPLE)

b) newmult(binary A[n], binary B[n])

begin

~~ostuno~~

if (n = 1)

if (A[1] = 0)

result := 0

else result := B

else begin

$A_H = A[n-1 \ldots n/2]$

$A_L = A[n/2 - 1 \ldots 0]$

$B_H = B[n-1 \ldots n/2]$

$B_L = B[n/2 - 1 \ldots 0]$

T1 = ~~AH~~ newmult(A_H, B_H)

T2 = newmult($A_L + A_H$, $B_L + B_H$)

T3 = newmult(A_L, B_L)

result := (T1 << n) + (T2 − T1 − T3) << (n/2)

+ (T3).

~~[8]~~ [8]

end

end

c) Let $a \geq 1$ be real, $b > 1$ be integer, $c > 0$ be real, $d \geq 0$ be real

Let $f(n)$ be increasing s.t.

$$f(n) = a f(n/b) + cn^d \quad \text{when}$$

$n = b^k$, for +ve integer k.

(i) If $a < b^d$, $f(n)$ is $O(n^d)$

(ii) If $a = b^d$, $f(n)$ is $O(n^d \log n)$

(iii) If $a > b^d$, $f(n)$ is $O(n^{\log_b a})$

~~[8]~~ [8]

(BOOKWORK)

d) We have

$$f(n) = 3f(n/2) + \Theta(n)$$

$a > b^d$

so $f(n)$ is $O(n^{\log_2 3})$

and so is run time. This is an improvement on the $\Theta(n^2)$ algorithm.

~~[8]~~ [7]

(NEW COMPUTED EXAMPLE)