Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Fundamentals of Computation

Date: Thursday 26th May 2016
Time: 09:45 - 11:45

Please answer any THREE Questions from the FOUR Questions provided.

Use a SEPARATE answerbook for each SECTION.

This is a CLOSED book examination

The use of electronic calculators is NOT permitted
Section A

1. a) Consider the language $L$ of all words over the alphabet $\{a, b, c\}$ in which every occurrence of the letter $b$ is both preceded and followed by the letter $a$.

Give a description of $L$ via the following means:

i) using a DFA  (3 marks)
ii) using a grammar  (2 marks)
iii) using a regular expression  (2 marks)
iv) using the language of set theory  (2 marks)

b) Give a DFA for the language of all words over the alphabet $\{a, b, c\}$ that match the regular expression

$$a^*bc^* | ab^*c$$

(8 marks)

c) For a language $L$ over an alphabet $\Sigma$, we can construct the language $L^R$:

$$L^R = \{x_nx_{n-1} \cdots x_2x_1 \mid x_1x_2 \cdots x_{n-1}x_n \in L\}.$$ 

*Proposition: For every regular language $L_1$, the language $L_1^R$ is also regular.*

Sketch out a proof for the above proposition.  (3 marks)
2. a) Consider the two DFAs $X$ and $Y$ given below.

$X$

```
A -- a -> B -- a -> C
  b   c
```

$Y$

```
0 -- a -> 1
   b   a
  2   b
        a
  3   a
```

Proposition I These two automata are not equivalent.

Proposition II By adding one or more transitions to automaton $Y$, we can produce a new DFA $Y'$ which is equivalent to $X$.

i) Provide a proof for Proposition I above, i.e. demonstrate that the automata are not equivalent. (2 marks)

ii) Draw an automaton that satisfies the conditions for $Y'$. (4 marks)

iii) Demonstrate that Proposition II holds, i.e. that $X$ and your proposed automaton $Y'$ are equivalent. (4 marks)
b) Are the following languages over the alphabet \( \{a, b\} \) regular? Justify your answer.

i) The language \( L_1 = \{a^n \mid n \in \mathbb{N}, \text{n is even}\} \). (2 marks)

ii) The language \( L_2 = \{a^n b^n \mid n \in \mathbb{N}, \text{n is even}\} \). (2 marks)

c) Consider the following grammar. The underlying alphabet is \( \{a, b, c\} \), there are two non-terminal symbols \( S \) and \( T \), the start symbol is \( S \) and the production rules are:

\[
S \rightarrow TabT \\
T \rightarrow aT \mid bT \mid cT \mid \varepsilon
\]

i) Show that this grammar is ambiguous. (2 marks)

ii) Describe the language generated by this grammar (1 mark)

iii) Give an unambiguous grammar for the same language (3 marks)
3. a) The following program calculates a function $f(n)$ for each input value $n$, with the result being stored in variable $a$. Tabulate the function for values of $n$ satisfying $0 \leq n \leq 5$.

```plaintext
if n = 0 then a := 0 else a := 1;
b := 0;
while 2 <= n do
    ( tmp := a;
      a := a + b;
      b := tmp;
      n := n - 1
    )
```

(4 marks)

b) Using induction or otherwise show that the function implemented in part (a) can be defined recursively as:

$$
\begin{align*}
f(0) &= 0, \\
f(1) &= 1, \\
f(n+2) &= f(n+1) + f(n).
\end{align*}
$$

(You need not use pre- and post-conditions, but you should explain the relationship of your proof to the program.) (6 marks)

c) Let

$$
\phi = \frac{1}{2}(1 + \sqrt{5}), \\
\hat{\phi} = \frac{1}{2}(1 - \sqrt{5}) = 1 - \phi.
$$

Using induction or otherwise show that

$$f(n) = \frac{1}{\sqrt{5}} (\phi^n - \hat{\phi}^n).$$

(6 marks)

d) Using the fact that $|\hat{\phi}| < 1$, argue that the function $f$ is in $O(\phi^n)$.

(4 marks)
4. a) Explain the following concepts: computable function; decidable predicate; semi-decidable predicate, and the characteristic function of a predicate. (4 marks)

b) What is the Church-Turing Thesis? Argue that testing whether a number is a prime number is a decidable function. (6 marks)

c) What is the Diagonal Program? (6 marks)

d) i) Could you write a program which takes as input the source code of a while program and a value and outputs whether or not that program will halt when run with that value as input? If not, why not? (1 mark)

ii) If there was a special halt construct added to the while language that told us whether or not a program would halt when run on a particular input value, what can you say about the resulting programming language? (3 marks)