00066

THE OPEN UNIVERSITY OF SRI LANKA
FACULTY OF ENGINEERING TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ECX 6235 – Compiler Design

Final Examination – 2009 / 2010 *(No Book Type)*

Date : Tuesday, 23rd of March, 2010 Time : 14:00 – 17:00

INSTRUCTIONS TO CANDIDATES

You must answer any two questions from Part A and another two questions from Part B. Questions 1 – 3 carry 30 marks each and question 4 – 7 carry 20 marks each.

Part A

Consider following description of a compiler to answer questions 1 and 2.

A compiler converts pseudo-Assembler code expressions into their equivalents of simple arithmetic operations in the following way:

ADD A, B	$\Rightarrow$	$C = A + B$		
SUB A, B	$\Rightarrow$	$C = A - B$		
MULB A, B	$\Rightarrow$	$C = A \times B$		
MULT A, B	$\Rightarrow$	$C = A^B$	or	$C = A ** B$
DIV A, B	$\Rightarrow$	$C = A \div B$		

Where A and B will be single digit decimal numbers (no decimal point). All pseudo-Assembler codes are input into the compiler as separate lines. Therefore, one input can carry only one pseudo-Assembler command string. Each command string must end with a blank.

Your task is to build the part of the compiler that performs only the above mentioned conversions of the code.

Note: Write down all your assumptions where necessary, when answering questions 1 and 2 below.

1. (a) What is the alphabet of this parser? [6 marks]

(b) Give the token table for this lexical analyzer. [4 marks]

(c) Write a regular expression to represent the code expressions in the assembly code. [7 marks]

(d) Draw a nondeterministic finite automaton (NDFA) that can accept the regular expression constructed in 1.(c). [6 marks]

(e) List seven possible errors and discuss how to handle them in this lexical analyzer. [7 marks]

2. (a) Define the syntax rules for the above compiler. [7 marks]

(b) List four possible errors detected by the syntax analyzer and discuss how to handle them. [8 marks]

(c) Draw a flowchart of the syntax analyzer rules you've defined. [15 marks]

3. (a) Strings are input into a Turing Machine with an unlimited tape. Each string consists only of 1s and 0s. The Turing Machine converts every third digit into its opposite, i.e. 1 to 0 and 0 to 1. If the digit right after the converted one is opposite to the one just written it changes it back again to its original value. 'B' denotes a blank symbol. You may neglect any possible error occurrences. Assume that the length of the string can be only of multiples of three, i.e. 3, 6, 9, ..., and that there are at least two blank symbols between the strings. The tape will consist of five strings at least, but there may be more. If the Turing Machine will encounter a point (.) symbol it will terminate its operation. Note that the point (.) symbol can only be encountered instead of a string.

For example consider following input and output strings:

```
Input:       ... B B B 1 1 0 0 0 1 0 1 0 B B B ...
Conversion:  ... B B B 1 1 1 0 0 0 0 1 1 B B B ...
Output:      ... B B B 1 1 0 0 0 0 0 1 1 B B B ...
```

(Digits to be changed and the changed ones are underlined.)

Draw the transition graph for the above Turing Machine. [20 marks]

(b) Give the transition table for the transition graph drawn above. [10 marks]

Part B

4. (a) Find the NDFA that corresponds to $(a + b^+)^* a b^+ a$. [3 marks]

(b) Convert the above NDFA to an equivalent DFA. [8 marks]

(c) Give regular expressions to generate the following strings over the alphabet {**x**, **y**, **z**}.

a) Set of all strings where **y** is the third character only. [3 marks]

b) Set of all strings where every **y** is followed by at least one **zx**. [3 marks]

c) Set of all strings of length 4 or more starting with **z** and ending with **x**. [3 marks]

(a) (i) Define an Operator Grammar. [3 marks]

(ii) Give three productions to illustrate an Operator Grammar. [3 marks]

(b) Give the formal definition of Pushdown Automaton (PDA). [6 marks]

(c) (i) What are the major resources needed to implement an algorithm? Explain them briefly. [4 marks]

(ii) What is considered under computational complexity in Turing Machines? [4 marks]

6. (a) A grammar G is defined by the following productions, where uppercase letters are non-terminals and lowercase letters are terminals.

$S \rightarrow M \mid c \mid N\ f\ D \mid e\ P\ a$
$B \rightarrow D\ f\ a \mid a\ L\ e \mid N$
$C \rightarrow e \mid C\ l$
$D \rightarrow a \mid e\ e \mid a\ S\ k$
$L \rightarrow k\ S\ e\ M \mid C\ k\ f$
$M \rightarrow e\ a \mid D$
$N \rightarrow k\ C \mid f \mid C$
$P \rightarrow f\ P\ l \mid S\ a\ P$

Find an equivalent grammar without useless symbols and unit productions. [8 marks]

(b) Convert the above into CNF. [12 marks]

7. (a) A grammar G is given by

$G = (\{S, D, F, G\}, \{m, p, t\}, P, S)$

where P is given by the following productions:

$S \rightarrow t\ m\ D\ p$
$m\ D \rightarrow D\ m$
$F\ p \rightarrow p$
$t\ D \rightarrow t\ t\ m\ G$
$G\ p \rightarrow G\ m\ p$
$m\ G\ m \rightarrow m\ F\ p$
$S \rightarrow t\ m\ p$
$F\ m \rightarrow m\ F$
$F\ p \rightarrow D\ p$

Show that this grammar generates the language

$L = \{ t^n\ m\ p^n \mid n \geq 1 \}$. [12 marks]

(b) Is L context- free? Prove your answer. [8 marks]