038

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

ECX 6235 – Compiler Design /
ECU 4307 – Theory of Computation

Final Examination – 2005 / 2006 ***(Closed Book Type)***

Date : Wednesday, 22nd of March 2006 **Time : 09:30 – 12:30**

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.

You are asked to build encryption software, which will accept strings containing only three words. Words will be separated by a single period (".") and strings will be separated by a single space (blank) (" ", "B"). A word can contain both letters and digits and it must not be longer than 5 characters and shorter than 1 character. Characters can only contain first three letters of English alphabet and only first three digits, i.e. a, b, c, and 0, 1, 2. Encryption program will read the input string and will reorganize it into five word string, changing digits into corresponding letters and letters into corresponding digits. The number of words in the output string is increased by inserting periods after the first two characters of the whole string and before the last two characters of the whole string.

Example 1:
Input string: `cab02.2ac1.01bc2`
Output string: `20.1ac.c02b.ab1.2c`

If the length of the first and/or the last words in the input string is less than three characters additional two words in the output string will be formed by adding "00" at the beginning and "aa" at the end of the original string.

Example 2:
Input string: `2a.cb1ba.c`
Output string: `00.c0.21b10.2.aa`

1. (a) What is the alphabet of this compiler? [5 marks]

   (b) Give the token table for this lexical analyzer. [7 marks]

   (c) What are the functions of the lexical analyzer in the above compiler? [4 marks]

   (d) Write an algorithm for the above lexical analyzer in pseudo code. [14 marks]

2. (a) What are the functions of the syntax analyzer for the above compiler? [3 marks]

   (b) Write the set of rules of the syntax analyzer. [7 marks]

(c) Write an algorithm for the syntax analyzer in pseudo code. [20 marks]

3. (a) A string is input into the Turing Machine on an unlimited tape. The Turing Machine performs two actions on the given string:

1. it inserts an asterix (*) into the string after it encounters '1' the second time, and
2. it converts the string into its 1's complement.

Assume that the string cannot be longer than 20 characters and that each string will contain at least three '1's. 'B' denotes a blank symbol. You may neglect any possible error occurrences.

For example consider following input and output strings:

```
... B B B 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 1 0 0 1 1 B B B ...
... B B B 1 1 1 0 1 1 1 1 1 0 * 0 0 1 1 1 0 1 1 0 0 B B B ...
```

Draw the transition graph for the above Turing Machine. Note that the Turing Machine must perform both actions during a single (direct) parse (scan) of the string. [20 marks]

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

Part B

4. (a) Find the NDFA equivalent to **a b (c + b)* a c**. [5 marks]

(b) Convert the above NDFA to equivalent DFA. [10 marks]

(c) Explain briefly where finite automata is used in a compiler and for what purposes. [5 marks]

5. (a) Find a PDA that accepts the language generated by grammar defined by (Hint: use GNF.)

S → x z A y B | B z C

A → B C | x x z y C

B → y C A z | z A z

C → x S y

[15 marks]

(b) What is the place of PDA in a compiler? How is it used? [5 marks]

6. (a) A grammar G is defined by the following productions

S → kQmPS
P → m
S → P
P → T
S → hPiSjS
T → Q
Q → xz
U → km

Find equivalent grammar without useless symbols and unit productions. [10 marks]

(b) Convert above into CNF. [10 marks]

7. (a) Identify the type of following grammar. Explain your answer.

$$G = (\{X\}, \{a, b, c\}, \{X \rightarrow X\,c\,X,\ X \rightarrow a,\ X \rightarrow b,\ X \rightarrow c\}, X)$$ [4 marks]

(b) Find leftmost and rightmost derivations for the string **acbcacbca**. [8 marks]

(c) Suggest a method to make above grammar unambiguous. Show how your method work using an example. [8 marks]