IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2014

EIE PART II: MEng, Beng and ACGI

Corrected Copy

LANGUAGE PROCESSORS

Friday, 13 June 2:00 pm

Time allowed: 2:00 hours

There are THREE questions on this paper.

Answer ALL questions.
Q1 carries 40% of the marks. Questions 2 and 3 carry equal marks (30% each).

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

Examiners responsible First Marker(s) : Y.K. Demiris
Second Marker(s) : J.V. Pitt

The Questions

Answer ALL questions

1.

(a) Chomsky's hierarchy of grammars defines four types of grammars; describe each of the four types discussing the restrictions that each type imposes on its grammar production rules. [6]

(b) Provide the formal definition of a Turing Machine. Explain the differences between a Linearly-bounded Automaton (LBA) and a Turing Machine. Describe the benefits of an LBA over a Turing Machine. [6]

(c) Define the concept of *translation scheme,* and provide the translation scheme for a infix-to-postfix notation translator that only has to deal with a single infix expression, containing one or more addition, subtraction, multiplication and division operators, and single digit numbers as operands. [6]

(d) In the context of shift-reduce parsing, and in order to construct the parsing table, the functions closure(I) and goto(I, X) need to be defined (where I is a set of items and X is a grammar symbol). Provide the definition of these functions. [6]

(e) In the context of three-address intermediate code generation, describe the concept of basic blocks, and provide the algorithm for partitioning a sequence of three address statements into basic blocks. Explain the utility of this algorithm. [6]

(f) In the context of top-down parsing, the calculation of FIRST and FOLLOW sets are used in the construction of parsing tables. Assuming that these sets have been calculated provide the algorithm for constructing a top-down parsing table. [10]

2. You are required to construct the minimal deterministic finite state automaton (DFA) for the regular expression ***a*\|(bc*)\| d**** following the steps below. You should *clearly mark all final states* in *all* the automata you construct in the process.

(a)

i. Specify Thompson's construction algorithm for constructing a non-deterministic finite automaton (NFA) from a regular expression. [4]

ii. Construct a non-deterministic finite automaton (NFA) using Thompson's algorithm. [6]

(b)

i. Specify the subset construction algorithm for constructing an equivalent DFA from an NFA, including a specification of the required functions, *ε-closure* and *move*. [4]

ii. Construct the equivalent DFA using the subset construction algorithm. *Explain the intermediate steps you have taken.* [6]

(c)

i. Specify the DFA minimization algorithm. [5]

ii. Apply the DFA minimization algorithm to the DFA you have constructed in (b). Show whether your DFA was already minimal or not. *Explain clearly the intermediate steps of the application of the DFA minimization algorithm.* [5]

3. (a) For the augmented grammar below,

0. E' -> E
1. E -> E + T
2. E -> T
3. T -> T * F
4. T -> F
5. F -> (E)
6. F -> id

compute the canonical set of LR(0) items. [12]

(b) Each set I_i from the ones computed above constructs a state i. Provide the four rules for the simple LR parsing table construction algorithm that determines the "action" and "goto" entries for state i. [8]

(c) Using the canonical set of LR(0) computed in (3a), and the table construction algorithm you provided in (3b), construct the parsing table for this grammar using the table format below (three entries already completed). Show the construction of the FIRST and FOLLOW sets of all the non-terminal symbols [10]

State	Action entries						Goto entries		
	id	+	*	(	)	$	E	T	F
0									
1		s6							
2									
3			r4						
4								2	
5									
6									
7									
8									
9									
10									
11									

(NB: Make sure you copy this table to your exam booklet!)