E2.15

IMPERIAL COLLEGE LONDON

Master copy - June 2008

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2008

ISE PART II: MEng, BEng and ACGI

LANGUAGE PROCESSORS

Friday, 6 June 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) :	Y.K. Demiris, Y.K. Demiris
	Second Marker(s) :	J.V. Pitt, J.V. Pitt

The Questions

1. [COMPULSORY]

(a) Provide three example functions that the Context Handling phase of compilation performs. Subsequently, provide three reasons for incorporating an intermediate code generation stage during compilation. [6]

(b) Provide the formal definition of a Linearly Bounded Automaton (LBA) and describe the differences between a Push Down Automaton (PDA) and a Linearly-Bounded Automaton (LBA). [6]

(c) Within Chomsky's hierarchy of grammars, describe the main difference between a type-2 and a type-3 grammar, and provide two example production rules (one for each type of grammar) that illustrate this difference. [6]

(d) Describe the data structures involved, and the steps performed by the LR parsing algorithm. [6]

(e) Within the context of converting a Non-Deterministic Finite Automaton (NFA) to a Deterministic Finite Automaton (DFA), describe the subset construction algorithm by providing the two functions required by the algorithm, as well as the steps involved in the operation of the algorithm. [8]

(f) Provide the definition of LL(1) grammars; explain why the grammar below is not LL(1), and use left-factoring to transform it to its LL(1) equivalent.

```
X -> yXa
X -> yXb
X -> c
```

[8]

2. You are required to construct the minimal deterministic finite state automaton (DFA) for the regular expression *a*(b|c|d)*e* following the steps below.

(a) Construct a non-deterministic finite automaton (NFA) using Thompson's algorithm. [12]

(b) Construct the equivalent DFA using the subset construction algorithm. *Explain the intermediate steps you have taken.* [12]

(c) Describe the DFA minimization algorithm, and subsequently apply it to the DFA you have constructed in (b). Show whether your DFA was already minimal or not. *Explain the intermediate steps of the application of the DFA minimization algorithm* [6]

3. (a) 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 definitions of these functions. [6]

(b) Provide the algorithm for computing the canonical LR(0) collection of sets of items. [6]

(c) For the grammar below, construct the canonical set of LR(0) items and provide the DFA that can recognize viable prefixes for the grammar:

G' -> G
G -> G – X | X
X -> X * F | F
F -> (G) | a

[18]

4. (a) Calculate the FIRST and FOLLOW sets for all non-terminal symbols for the grammar below where {a, -, *, (,) } are terminals, and {G', G, T, T', F} are non-terminals:

(1) G -> T G'
(2) G' -> - T G' | ε
(3) T -> F T'
(4) T' -> * F T' | ε
(5) F -> (G) | a

[10]

(b) A partially constructed parsing table for the grammar above is given below, with $ denoting the end of input marker. Complete *ALL* remaining table entries, clearly marking any error entries as shown.

(NB: Make sure you copy this table to your exam booklet) [20]

Non-terminal	Input Symbol					
	a	-	*	(	)	$
G			*Error*			*Error*
G'					G'-> ε	
T				T -> FT'		
T'			T' -> *FT'			
F	*F -> a*	*Error*				