UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

COLLEGE *of* INFORMATICS *and* ELECTRONICS

Department of Computer Science
and Information Systems

End-of-Semester Assessment Paper

Academic Year:	**2014/15**	Semester:	**Autumn**
Module Title:	**P.L.T.**	Module Code:	**CS4158**
Duration of Exam:	**2½ Hours**	Percent of Total Marks:	**60**
Lecturer(s):	**Jim Buckley**	Paper marked out of :	**60**

Instructions to Candidates:

- Answer question 1.
- Answer 2 of the remaining 3 questions.
- Question 1 carries 20 marks.
- All other questions carry 20 marks.
- If you attempt all three questions (Q2-Q4), then please specify clearly which 2 of these questions should be marked. Otherwise your first 2 attempts will be marked.

Q1. a) Form a FLEX regular expression which describes: A string that starts with 1-to-5 'D's, followed by an optional 'E', followed in turn by 0, 1 or more capital letters, excluding the letters 'D', 'E' and 'P', any of which serve to finish the string.

4 Marks

b) Draw a deterministic FSA for the regular expression in part 'a' above, and from it create a transition table *(Note that even though you might not be able to form the F LEX regular expression, you might still be able to form the FSA)*

5 Marks

c) Build a grammar that embeds the relative priority of the mathematical operators multiplication division, minus, and plus into its structure, briefly explaining how it works.

3 Marks

d) Build a Context Free grammar that could parse phrases of the following structure. Show, using an expression tree, how the grammar would parse the third phrase.

He fell to his knees.
She fell upon her sword.
It landed on its side.

4 Marks

e) Give 4 sentential forms of the following grammar:

S -> hTrD
T -> Pa
T -> X
T -> q
D -> y
X -> z
X -> y
P -> r

4 Marks

Q2. a) The following grammar is not LL(1). Explain why and transform it into an LL(1) grammar, explaining the transformation:

S -> P Q $
P -> P t
P -> y
P → f
Q -> Xr
Q -> j
X-> hyg
X-> λ

8 Marks

b) Create the LL(1) look ahead set for the resultant grammar and generate the associated LL(1) parse table.

9 Marks

c) Briefly describe the role of this LL(1) parse table, illustrating it with an example from the grammar above.

3 Marks

Q3. a) Create a Post Production system that generates strings that consist of an odd number of ‘a’s and an identical (odd) number of ‘b’s. Show how it allows for the creation of abaabbabab.
(Examples include ab, ababab, bbbaaa)
(Examples do NOT include abab, bbbbaaaa)

7 Marks

b) Contextualize the important of Post’s Production Systems for Parsing

6 Marks

c) Flex is a Transducer. Explain what this means and, in the context of a programming language, why it is important.

4 Marks

c) Discuss the role the y.tab.h file plays in a FLEX/BISON parser.

3 Marks

Q4. a) Build the LR(0) Finite State Machine for the following grammar, showing that it (the grammar) requires an LR(1) FSM for parsing

S -> dHpG$
H -> fd
H -> fdr
G -> hq
G -> hqn

10 Marks

b) Build the LR(1) Finite State Machine that would allow language instances of this grammar to be parsed.

6 Marks

c) From this Finite State Machine, build the Action and Goto tables.

4 Marks