UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

COLLEGE *of* INFORMATICS *and* ELECTRONICS

Department of Computer Science
and Information Systems

End-of-Semester Assessment Paper

Academic Year:	**2007/08**	Semester:	**Spring**
Module Title:	**P.L.T.**	Module Code:	**CS4158**
Duration of Exam:	**2½ Hours**	Percent of Total Marks:	**65**
Lecturer(s):	**Jim Buckley**	Paper marked out of :	**65**

Instructions to Candidates:

- Answer question 1.
- Answer 2 of the remaining 3 questions.
- Question 1 carries 25 marks.
- All other questions carry 20 marks.
- Your first 3 attempts will be marked unless you explicitly state otherwise.

Q1. a) Form a ‘LEX’ regular expression that identifies a token that starts with 1, 2 or 3 ‘a’s, and is then made up of a sequence of any characters (numbers letters, punctuation) apart from the letter ‘a’. The final character in the token must be a ‘t’. For example ***aac80505^&%$$rt***, ***at***, and ***aaa&*^rd$5t*** are valid, while ***aaaacjir*^86t aa^%rat*** and ***aa&*gtrf*** are invalid.

5 Marks

b) Draw both a deterministic FSA and a transducer for the regular expression constructed in part ‘a’ above, and from it, create a transition table.

5 Marks

c) Describe why it is inappropriate to use an ambiguous grammar in a compiler.

5 Marks

d) Classify the following 2 grammars as either Context Free Grammars, Context Sensitive Grammars, Unrestricted Grammars or Regular Grammars (Explaining your reasons)

Grammar 1	Grammar 2
E -> aB	E -> gH
B -> dH	H -> RT
H -> cGd	rR -> k
G -> ad	T -> p
G -> λ	

5 Marks

e) Give 4 sentential forms of the grammar in question 4(a)

5 Marks

Q2. a) Describe the structural issue addressed by the mathematical conventions of associatively and priority

4 Marks

b) Write a CF grammar that embeds the relative priority of the operators ‘+’, ‘-‘, ‘*’ and ‘/’ in its structure, demonstrating its correctness by drawing the parse tree for 7+6*5+4*3

10 Marks

c) Write a CF grammar that embeds the associativity of the ‘^’ (to the power of) operator in its structure, demonstrating its correctness by drawing the parse tree for 7^5^3

6 Marks

Q3. a) Transform the following into an LL(1) grammar, explaining the transformation:

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

7.5 Marks

b) Using the resultant LL(1) grammar, calculate the predict set for each production.

9.5 Marks

c) Use this predict set to form a LL(1) Parse table for the grammar.

3 Marks

Q4. a) Build the LR(1) FSA for the grammar, and use it to illustrate how the LR(0) equivalent would be unsuitable

S->R $
R->R / Q
R->Q
Q->Q - z
Q-> e

12 Marks

b) Discuss the problem associated with LR(1) parsing of real programming languages and identify 2 possible solutions.

2 Marks

c) For one of these solutions, discuss how it addresses this problem and how it loses precision somewhat in doing so.

6 Marks