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:	**Autumn Rpt.**
Module Title:	**P.L.T.**	Module Code:	**CS4158**
Duration of Exam:	**2½ Hours**	Percent of Total Marks:	**100**
Lecturer(s):	**Jim Buckley**	Paper marked out of :	**100**

Instructions to Candidates:

- Answer question 1.
- Answer 2 of the remaining 3 questions.
- Question 1 carries 40 marks.
- All other questions carry 30 marks.
- Your first 3 attempts will be marked unless you explicitly state otherwise.

Q1. a) Form a 'LEX' regular expression which has an optional '€' followed by a number that may or may not contain a decimal point. If it does contain a decimal point there may be a number before that decimal point but there must be 2 numbers after that decimal point. So valid expressions include: €78.85, €.67, 7.56, 7 and €7
Invalid expressions include €4.7, and €78. (where the '.' is a decimal point)

8 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.

8 Marks

c) Explain how LEX and YACC have a common understanding of terminal vocabulary

8 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 -> sHs	E -> gH
E -> dG	H -> T
H -> cGd	T -> kT
G -> ad	T -> p
G -> λ	

8 Marks

e) Give 4 sentential forms of 'Grammar 1' in part d of this question

8 Marks

Q2. a) Building the LR(0) finite state automata for the following grammar

S->E$
E->E+T
E->T
T->d
T->(E)

12 Marks

b) Show using this Finite State Automata, how it would parse d+(d)+d$

8 Marks

c) Build the LR(1) finite state automata for the following grammar, showing why the LR(1) automata is required

S-> P$
P-> R
R-> R-T
R-> q
T-> h

10 Marks

Q3. a) 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

b) Create a Post Production system that generates statements that consist of an odd number of ‘a’s and an identical (odd) number of ‘b’s.
(Examples include ab, ababab, bbbaaa)
(Examples do NOT include abab, bbbbaaaa)

8 Marks

c) Show how, using this production system, you would create the language instance: abaabbabab

4 Marks

d) Explain the error in the following segment of Flex code and the error in the following piece of Bison code:

```
Flex:
[a-zA-Z]+     {printf(“Is an identifier”);  }
For           {printf(“Is the keyword: FOR”); }

Bison
Expression: NUMBER ‘+’ NUMBER   {$$=$1+$2}
```

8 Marks

Q4. 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
```

12 Marks

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

12 Marks

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

6 Marks