UNIVERSITY *of* LIMERICK

O L L S C O I L L U I M N I G H

COLLEGE *of* INFORMATICS *and* ELECTRONICS

Department of Computer Science
and Information Systems

End-of-Semester Assessment Paper

Academic Year:	**2006/07**	Semester:	**Spring**
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.
- Your first 3 attempts will be marked unless you explicitly state otherwise.

Q1. a) Form a 'LEX' regular expressions for a float, made up of 1 or more numbers, preceded by an optional '-', with one optional decimal place. If the decimal place is present, there must be numbers on its RHS, but not necessarily on its LHS (Examples include: 10, -5, -78.98, .987 and -.87987)

4 Marks

b) Draw a transducer for the regular expression constructed in 'a' above, and from it create a transition table

4 Marks

c) Explain the error in each of the following segments of Flex and Bison code:

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

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

4 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 -> dGH   H -> RT
GH -> cGdG   R -> k
cGd -> ad   T -> p
G -> λ

4 Marks

e) Build a grammar that accommodates the 'Right-to-Left' associativity of the '^' (to the power of) operator, explaining how it works.

4 Marks

Q2. a) Build an LR(0) finite state automata for the following grammar:

```
S->E $
E->E + T
E-> gh
T->F
T->F * p
F->g h y
```

8 Marks

b) Using this Finite State Automata to illustrate the LR(0) parsing of "g h + g h y * p", show how it is insufficient and suggest a remedy

6 Marks

c) Briefly describe Emile Post's contribution to Compiler theory

6 Marks

Q3. a) Transform the following into an LL(1) grammar.

S -> P $
P -> y X X
P -> y X Q
X -> r
X -> g
Q -> f
Q -> j

9 Marks

b) Using the LL(1) grammar below, calculate the predict set for each production:

S -> T P r $
T -> P f
T -> w [T]
P -> j C
P -> λ
C -> h y g
C -> k

8 Marks

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

3 Marks

Q4. a) Create a Context Free grammar that describes personal ads with the following structure:

- Tired emotional young man seeks friendly understanding young lady for relationship – 086 1987678
- Generous rich old lady wants fun handsome hunk for long slow romantic walks – 097 6548374
- Blonde serious gent desires kind serious lady for companionship – 086 5437234
- Woman seeks generous friendly man – 061 3274586
- Kind serious man seeks generous friendly female for deep conversations – 091 5674328

14 Marks

b) Prove the generality of the grammar by deriving a parse tree for the language instance:
Handsome friendly rich old man seeks handsome friendly rich old woman for long serious deep conversations – 086 5464532

6 Marks