



UNIVERSITY *of* LIMERICK

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

COLLEGE *of* Science and Engineering

Department of Computer Science
and Information Systems

End-of-Semester Assessment Paper

Academic Year:	2016/17	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 :	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 2 30-mark questions attempted will be marked, unless you explicitly request otherwise.

- Q1. a) Form a FLEX regular expression and a Transducer which describes a construct that starts with an optional minus sign (-) and ends with a minus sign. In between must begin with a letter (upper or lower case) which can be followed by any combination of letters, numbers and may, at most, contain one asterix (*). Examples include '-p-', '-u8r*y75-', '-yyy-' and 'ggG7*89-'
- 10 Marks
- b) Using a transition table generated from this transducer show how -u8r*y75- is a valid construct according to this transducer.
- 10 Marks
- c) Using a grammar example of your choice, discuss the left recursion problem and show how your chosen grammar can be transformed to remove the problem.
- 10 Marks
- d) Discuss 2 different approaches to the “state-proliferation, memory” problems faced by LR(1) grammars, showing how they work to limit the number of CFMSM states.
- 10 Marks

- Q2 a) Describe the guidelines you should adhere to when writing code for a recursive descent grammar which uses a look-ahead of 1. 6 Marks
- b) Implement those generalized guidelines to generate the recursive descent pseudo-code for the following grammar, after having calculated the predict sets for each rule:
- $S \rightarrow jAXr$
 $A \rightarrow TH$
 $A \rightarrow Rp$
 $R \rightarrow m$
 $R \rightarrow n$
 $R \rightarrow \lambda$
 $T \rightarrow f$
 $T \rightarrow wt$
 $T \rightarrow \lambda$
 $H \rightarrow j$
 $H \rightarrow \lambda$
 $X \rightarrow c$
 $X \rightarrow \lambda$
- 18 Marks
- c) Using this code, manually step through the parsing of the language instance: jmprr 6 Marks
- Q3 a) Compare and contrast Post's work on Production Systems to Chomsky's foundational work on parsing. 8 Marks
- b) Define the difference between a Context free grammar and a Regular grammar (in terms of grammar structure and in terms of their memory requirements) 6 Marks
- c) Specify a Regular grammar that allows for the correct parsing of the following statements:
- Why are apples red?
 Why are elephants big?
 How is a dog trained?
 Where is the food located?
- 10 Marks
- d) Illustrate the generic nature of your grammar by showing the resultant parse tree for the language instance: How are elephants trained? 6 Marks

- Q4. a) Build the LR(0) Finite State Machine (FSM) for the following grammar, identifying the states where it (the grammar) requires an LR(1) FSM for parsing

$S \rightarrow rTaK\$$

$T \rightarrow fyd$

$T \rightarrow f$

$T \rightarrow fK$

$K \rightarrow glm$

$K \rightarrow gl$

6 Marks

- b) Build the LR(1) Finite State Machine for this grammar.

16 Marks

- c) Using the resultant action table and goto table, walk through the parse of the language instance rfydagl\$

8 Marks