UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

COLLEGE *of* SCIENCE and ENGINEERING

Department of Computer Science
and Information Systems

End-of-Semester Assessment Paper

Academic Year:	**2021/22**	Semester:	**Autumn Repeat**
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 :	**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 attempted questions will be marked, unless you explicitly state otherwise.

Q1. a) Form a FLEX regular expression that describes Euro currency amounts. They should start with the Euro symbol, contain any number of digits before a decimal point and contain 0 or 2 digits after the decimal point. Examples include €401.45, €89765, and €.75. Please note that €4987.6 and €45. are not considered valid Euro currency amounts for the purposes of this exercise.

5 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 FLEX regular expression, you might still be able to form the FSA and this may in turn help form the FLEX regular expression).* Step through the transition table to show that €401.45 is a valid Euro currency amount.

5 Marks

c) Detail the role that y.tab.h plays in Lex/Yacc parser construction and the flag required for it's creation.

5 Marks

d) Describe how you can achieve case-insensitive variable names, method names, class names etc., in a lexer.

5 Marks

Q2. a) Create the LL(1) predict set for the following grammar

S -> Y T k $
Y -> j m
Y -> b h
Y -> λ
T -> X c x
T -> v
T -> λ
X-> h y g
X-> λ

8 Marks

b) Using this predict set, write the (recursive-descent) code that would enable language instances of this grammar to be parsed.

7 Marks

c) Using either the grammar/predict sets (derived in part 'a') OR the recursive descent code you created in part 'b', determine if jmk$ is a language instance of this grammar stating, for each non-terminal vocabulary you break down, which way you break it down and why.

5 Marks

Q3. a) Describe three of Chomsky's core breakthroughs in the field of parsing

6 Marks

b) Derive a Regular Grammar that all the following language instances adhere to (each phase can have 0, 1 or many descriptive words):
The yellow old cat.
A brown young dog.
The pink whale.
The new shiny yellow goldfish.

6 Marks

c) Explicitly note all sentential forms of your regular grammar encountered while deriving the last language instance presented in part 'b'.

4 Marks

d) Describe the common-prefix problem in LL(1) parsing and, using an example, show how it can be addressed.

4 Marks

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

S -> mKpG$
K -> al
K -> alna
G -> Krt
G -> t

6 Marks

b) Build the LR(1) Finite State Machine that would allow language instances of this grammar to be parsed, showing how this increases the size of the FSM generated.

6 Marks

c) Build the LALR and SLR finite state machines for this grammar.

8 Marks