

# DUBLIN CITY UNIVERSITY

## SEMESTER ONE REPEAT EXAMINATIONS 2008

**MODULE:** Compiler Construction 1  
**(Title & Code)** CA448

**COURSE:** B.Sc. in Computer Applications (SE)  
B.Sc. in Computer Applications (CSSE)  
B.Sc. in Computer Applications (IS)  
B.Sc. in Computational Linguistics

**YEAR:** 4

**EXAMINERS:** Dr. F. Bannister  
Dr. P. Gibson  
Dr. G. Hamilton (ext. 5017)

**TIME ALLOWED:** 2 Hours

**INSTRUCTIONS:** Please answer ALL questions.  
All questions carry equal marks

**Requirements for this paper**  
Please tick (X) as appropriate

<input type="checkbox"/>	<b>Log Table</b>
<input type="checkbox"/>	<b>Graph Paper</b>
<input type="checkbox"/>	<b>Attached Answer Sheet</b>
<input type="checkbox"/>	<b>Statistical Tables</b>
<input type="checkbox"/>	<b>Floppy Disk</b>
<input type="checkbox"/>	<b>Actuarial Tables</b>

**THE USE OF PROGRAMMABLE OR TEXT STORING  
CALCULATORS IS EXPRESSLY FORBIDDEN**

**PLEASE DO NOT TURN OVER THIS PAGE UNTIL YOU ARE  
INSTRUCTED TO DO SO**

1. Construct a non-deterministic finite state automaton (NFA) [10 marks]  
which recognises the same language as defined by the  
following regular expression:

$$(1^*01^*0)^*1^*$$

2. Using the subset construction method, convert your NFA in [10 marks]  
Question 1 into a corresponding deterministic finite state  
automaton (DFA), showing clearly the start state and  
accepting states.

3. Show that the following grammar is ambiguous: [10 marks]

$$\begin{aligned} S &\rightarrow \text{if } C \text{ then } S \mid \text{if } C \text{ then } S \text{ else } S \mid \text{id} := E \\ C &\rightarrow E = E \mid E < E \mid E > E \\ E &\rightarrow \text{id} \end{aligned}$$

by giving two different parse trees for the sentence:

$$\text{if } x < y \text{ then if } x < z \text{ then } x := z \text{ else } x := y$$

4. Determine whether or not the following grammar is LL(1): [10 marks]

$$\begin{aligned} S &\rightarrow ABC \\ A &\rightarrow aA \\ A &\rightarrow C \\ B &\rightarrow bB \\ B &\rightarrow \epsilon \\ C &\rightarrow cC \\ C &\rightarrow \epsilon \end{aligned}$$

5. Convert the following grammar into an LL(1) grammar [10 marks]  
which recognises the same language:

$$\begin{aligned} \text{Stat} &\rightarrow \text{with Desig do Stat end} \\ \text{Stat} &\rightarrow \text{Desig} := \text{Exp} \\ \text{Desig} &\rightarrow \text{id} \\ \text{Desig} &\rightarrow \text{Desig} . \text{id} \\ \text{Exp} &\rightarrow \text{Desig} \\ \text{Exp} &\rightarrow \text{Desig} + \text{Exp} \end{aligned}$$

6. Determine whether or not the following grammar is LR(0): [10 marks]

$E \rightarrow E+T$   
 $E \rightarrow T$   
 $T \rightarrow T * F$   
 $T \rightarrow F$   
 $F \rightarrow \text{id}$   
 $F \rightarrow (E)$

7. Determine whether or not the grammar in Question 6 is SLR(1). [10 marks]

8. Determine whether or not the following grammar is LR(1): [10 marks]

$S \rightarrow Ab$   
 $S \rightarrow B$   
 $A \rightarrow aB$   
 $B \rightarrow a$   
 $B \rightarrow aA$

9. Consider the following attribute grammar: [10 marks]

Production	Semantic Rules
$D \rightarrow T L$	$L.in := T.type$
$T \rightarrow \text{int}$	$T.type := \text{integer}$
$T \rightarrow \text{real}$	$T.type := \text{real}$
$L \rightarrow L_1, I$	$L_1.in := L.in$ $I.type := L.in$
$L \rightarrow I$	$I.type := L.in$
$I \rightarrow \text{id}$	$I.name := \text{id}$

Give a sequence in which the attributes would be evaluated for the following string:

int x,y

10. Convert the following expression into three-address code: [10 marks]

$(a + b) - (e - (c + d))$

[Total marks: 100]