



DUBLIN CITY UNIVERSITY

SEMESTER ONE REPEAT EXAMINATIONS 2009

MODULE TITLE: Compiler Construction 1
MODULE CODE: CA448
COURSE: BSc. in Computer Applications (Software Engineering)
YEAR: 4
EXAMINERS: Dr. F. Bannister
Dr. P. Gibson
Dr. G. Hamilton, Ext no. 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"/> | <i>Log Table</i> |
| <input type="checkbox"/> | <i>Graph Paper</i> |
| <input type="checkbox"/> | <i>Attached Answer Sheet</i> |
| <input type="checkbox"/> | <i>Statistical Tables</i> |
| <input type="checkbox"/> | <i>Floppy Disk</i> |
| <input type="checkbox"/> | <i>Actuarial Tables</i> |

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

Question 1**[10 marks]**

An e-mail address is defined as a user name, followed by an optional location. A user name consists of a letter followed by zero or more letters or digits. A location consists of the @ character, followed by one or more domain names, where domain names are separated by full stops. Domain names consist of a series of one or more letters. Using the input alphabet {letter, digit, @, .}, express an e-mail address as a regular expression.

Question 2**[10 marks]**

Construct a deterministic finite state automaton for e-mail addresses as described in Question 1.

Question 3**[10 marks]**

Show that the following grammar for Boolean expressions is ambiguous:

$$E \rightarrow E \text{ or } E \mid E \text{ and } E \mid \text{not } E \mid (E) \mid \text{id}$$

Question 4**[10 marks]**

Determine whether or not the following grammar is LL(1):

$$S \rightarrow ABC$$

$$A \rightarrow aA$$

$$A \rightarrow C$$

$$B \rightarrow bB$$

$$B \rightarrow \varepsilon$$

$$C \rightarrow cC$$

$$C \rightarrow \varepsilon$$

Question 5**[10 marks]**

Convert the following grammar into an LL(1) grammar which recognises the same language:

$$\text{Vardec} \rightarrow \text{var Declist}$$

$$\text{Declist} \rightarrow \text{Dec Declist}$$

$$\text{Declist} \rightarrow \text{Dec}$$

$$\text{Dec} \rightarrow \text{Idlist} : \text{id}$$

$$\text{Idlist} \rightarrow \text{Idlist} , \text{id}$$

$$\text{Idlist} \rightarrow \text{id}$$

Question 6**[10 marks]**

Determine whether or not the following grammar is LR(0):

```
S → Ab
S → B
A → aB
B → a
B → aA
```

Question 7**[10 marks]**

Determine whether or not the following grammar is SLR(1):

```
S → A
A → Ba
A → C
B → aC
B → b
C → B
```

Question 8**[10 marks]**

Determine whether or not the grammar in Question 7 is LR(1).

Question 9**[10 marks]**

Construct a control flow graph corresponding to the following piece of code:

```
a := b+c;
d++;
if a < b then
  c := a;
else
  c := b;
end;
d := c;
```

Question 10**[10 marks]**

Consider the following attribute grammar for binary numbers:

Production	Semantic Rules
$\text{Num} \rightarrow \text{Bitlist}$	$\text{Bitlist.pos} := 0$ $\text{Num.val} := \text{Bitlist.val}$
$\text{Bitlist} \rightarrow \text{Bit}$	$\text{Bit.pos} := \text{Bitlist.pos}$ $\text{Bitlist.val} := \text{Bit.val}$
$\text{Bitlist}_1 \rightarrow \text{Bitlist}_2 \text{Bit}$	$\text{Bitlist}_2.\text{pos} := \text{Bitlist}_1.\text{pos} + 1$ $\text{Bit.pos} := \text{Bitlist}_1.\text{pos}$ $\text{Bitlist}_1.\text{val} := \text{Bitlist}_2.\text{val} + \text{Bit.val}$
$\text{Bit} \rightarrow 0$	$\text{Bit.val} := 0$
$\text{Bit} \rightarrow 1$	$\text{Bit.val} := 2^{\text{Bit.pos}}$

Give a possible order of evaluation of the attributes for the following parse tree:

