UNIVERSITY OF JOHANNESBURG

FACULTY OF SCIENCE

COMPUTER SCIENCE APK CAMPUS

IT00087 SEMESTER TEST

14 APRIL 2005
MEMO

EXAMINER **Mr. A. Hardy**

TIME 120 MINUTES **MARKS 100**

Please read the following instructions carefully:

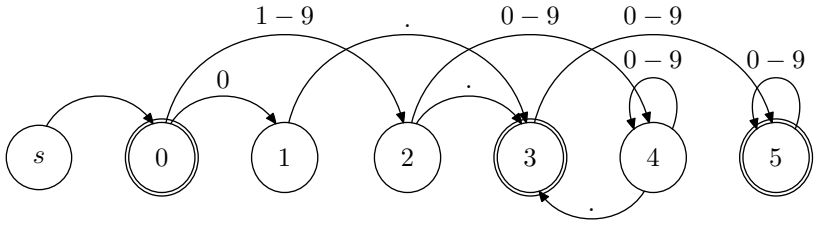
1. Write clearly and legibly.
2. This paper consists of 6 pages.

SECTION A

Answer all of the following questions.

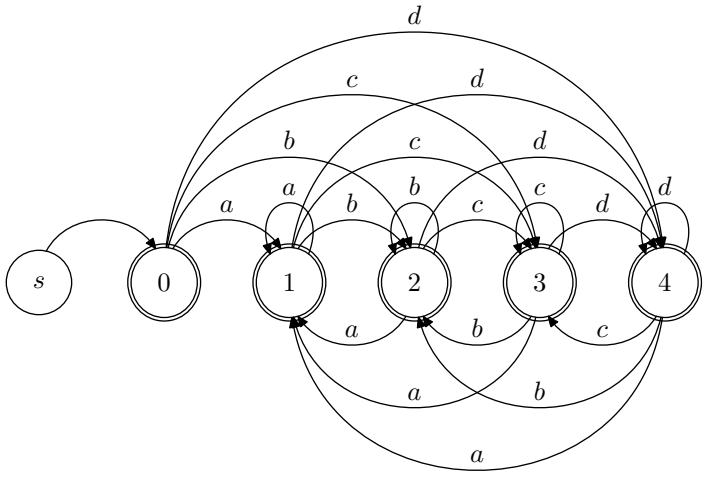
QUESTION 1

1.1 Draw finite automata to recognise the following regular expressions:
a) $(0|([1-9][0-9]*))[.][0-9]*$



(5)

b) $((a*b*c*)d*)*$



(5)

(10)

(10)

QUESTION 2

Consider the following Context Free Grammar:

$$
\begin{aligned}
S &\rightarrow (X \\
S &\rightarrow E] \\
S &\rightarrow F) \\
X &\rightarrow E) \\
X &\rightarrow F] \\
E &\rightarrow A \\
F &\rightarrow A \\
A &\rightarrow
\end{aligned}
$$

S is the start symbol in the grammar.

2.1 Determine the first and follow sets for each nonterminal as well as the nullable attribute.

	Nullable	First	Follow
S	**No**	**(,],),**	
X	**No**	**),],**	
E	**Yes**		**],),**
F	**Yes**		**),],**
A	**Yes**		**],),**

(15)

2.2 Construct a predictive parsing table for the grammar.

	(	)	]
S	$S \rightarrow (X$	$S \rightarrow F)$	$S \rightarrow E]$
X		$X \rightarrow E)$	$X \rightarrow F]$
E		$E \rightarrow A$	$E \rightarrow A$
F		$F \rightarrow A$	$F \rightarrow A$
A		$A \rightarrow$	$A \rightarrow$

(15)

2.3 Is this an LL(1) Grammar? **Yes** (1)

2.4 Remove left recursion (direct and indirect) from the grammar

$$\begin{aligned} X &\rightarrow Ya \\ Y &\rightarrow Z \\ Y &\rightarrow \\ Z &\rightarrow X \\ Z &\rightarrow b \end{aligned}$$

After removing left recursion we get

$$\begin{aligned} X &\rightarrow ZaY \\ Y &\rightarrow aY \\ Z &\rightarrow \\ Z &\rightarrow b \end{aligned}$$

(4)

(35)

QUESTION 3

3.1 List and *describe* the 5 phases of graph colouring by simplification. (No coalescing)
The stages are:

- **Build - Build the interference graph from the interference matrix**
- **Simplify - Remove a node of degree less than the number of registers and place on the stack.**
- **Spill - If there are no such node, mark a node as spilled and place on stack**
- **Select - Pop nodes of stack and colour, if we cannot colour then an actual spill takes place.**
- **Start over - In the case of an actual spill, place the variable in memory and rewrite the code to fetch the variable from memory. This changes everything, so start again.**

(5)

3.2 Consider the following interference matrix for variables and temporaries:

	u	v	x	y	z
u			X		
v			X	X	
x	X	X		X	X
y		X	X		X
z			X	X	

where X denotes an interference. Use graph colouring to allocate registers, assuming that there are three (3) registers available for use. **The graph and stack are omitted at each stage, but one possible sequence is:**

- **Build**
- **Simplify**
- **Pushed variable u on stack (and removed from graph)**
- **Pushed variable v on stack (and removed from graph)**
- **Pushed variable x on stack (and removed from graph)**
- **Pushed variable y on stack (and removed from graph)**
- **Pushed variable z on stack (and removed from graph)**
- **Select**
- **Popped variable z**
- **Assigned register 0 to z**

- **Popped variable y**
- **Assigned register 1 to y**
- **Popped variable x**
- **Assigned register 2 to x**
- **Popped variable v**
- **Assigned register 0 to v**
- **Popped variable u**
- **Assigned register 0 to u**

(10)

(15)

Section total: (60)

SECTION B

Answer two (2) of the following questions.

QUESTION 1

Discuss liveness analysis, and its use in register allocation. Use examples to illustrate how you determine if a variable is live at any particular point, and what the implications on register allocation are.

(20)

QUESTION 2

Discuss abstract parse trees with respect to semantic actions in a compiler. You should discuss how they can be produced, and how they can be used to encode semantic attributes, and how these attributes are retrieved during the semantic analysis phase of the compiler. Also discuss the role of the symbol table in semantic analysis.

(20)

QUESTION 3

Discuss intermediate code generation.

(20)

QUESTION 4

Discuss instruction selection algorithms. Include a discussion of the advantages and disadvantages of the algorithms you discuss

(20)

QUESTION 5

Discuss error recovery algorithms. Also discuss why error recovery is important to a programmer, and how successful these algorithms are at indicating errors correctly.

(20)

Section total: (40)

End of paper
