Indian Association for the Cultivation of Science

(Deemed to be University under *de novo* Category)

Master's/Integrated Master's-PhD Program/ Integrated Bachelor's-Master's Program/PhD Course

Mid-Semester (Sem-IV) Examination-Spring 2026

Subject: **Compiler Construction** *Subject Code:* **COM 5202**

Full Marks: **25** *Time Allotted:* **2 hours**

Q 1. Answer five (5) questions with brief justifications. [5 × 2 = 10]

(a) Draw a *deterministic state transition diagram* to generate tokens for the patterns { `a, b, baa, aaba, abaab` }, input $x \in$ {`a,b`}*. Show roll-backs from reachable non-final states for maximum length matching:

(b) Give an unambiguous *context-free grammar* of integer arithmetic expressions with binary operators '`+`', '`*`', and '`**`' (power, `2**3 = 8`) incorporating proper precedence and associativity of operators.

(c) Give a regular expression of a signed (+/ − /) binary floating point number. It must have a binary dot ('·'), mantissa must have some digit, and the exponent is optional. Use *name definitions*.

(d) Give a *context-free grammar* for variable declarations of simple variables and multi-dimensional arrays of types `int` and `double`.

(e) The system call `int fork()` creates a child process. It returns the child process-id (`pid`) to the parent. The system call code for `fork()` is **57**. It has no other parameters. Create an equivalent `myFork()` system call by filling the (`????`) with inline x86-64 code.

```
int myFork(){ int pid; __asm__volatile__ ( ???? ); return pid}
```

(f) Following C code prints the value of `a+1` as `0x7ffc3005b1c4` (GCC x86-64). What is the value of `a` in hex?

```
int main(){ int a[4][5]; printf("a+1: %p", a+1); return 0; }
```

(g) In a *flex* specification the terminals are {`if, else, while, for, int, (, ), {, }, :. ;, =, >, <. +, -, *, /, id, ic`}. What will be the effect if the rule for pattern 'dot(·)' is placed before all other rules?

(h) GCC x86-64 uses the register `rdi/edi` to pass the first parameter of a function call (size 64/32-bits). Suggest a scheme to pass a larger size parameter such as a structure.

Page 2: →

Answer any three (3) of the following questions.

$[\mathbf{3 \times 5 = 15}]$

Q 2. [4+1]

Consider the following C program:

```
int main(){int a=10, b, c; b=6*a; c=a+b; printf("%d\n", c); return 0;}
```

GNU x86-64 compiler translates the C program to the following *non-optimized* (sequence: 1 - 22) and *optimized* (seq: 1-7) assembly language code of x86-64.

(a) Explain the *non-optimized* code (seq: 1-22) and show its connection with the C program. Use the sequence numbers.

(b) Explain code improvements performed in both the codes.

```
pushq %rbp            # 1 movl  -8(%rbp), %eax #12
movq  %rsp, %rbp      # 2 addl  %edx, %eax     #13
subq  $16, %rsp       # 3 movl  %eax, -4(%rbp) #14
movl  $10, -12(%rbp)  # 4 movl  -4(%rbp), %eax #15
movl -12(%rbp), %edx  # 5 movl  %eax, %esi     #16
movl  %edx, %eax      # 6 leaq .LC0(%rip), %rdi#17
addl  %eax, %eax      # 7 movl  $0, %eax       #18
addl  %edx, %eax      # 8 call  printf@PLT     #19
addl  %eax, %eax      # 9 movl  $0, %eax       #20
movl  %eax, -8(%rbp) #10 leave                 #21
movl -12(%rbp), %edx #11 ret                   #22
```

Optimized Code

```
subq  $8, %rsp        # 1
movl  $70, %esi       # 2
leaq .LC0(%rip), %rdi# 3
call  printf@PLT      # 4
xorl  %eax, %eax      # 5
addq  $8, %rsp        # 6
ret                   # 7
```

Q 3. $[\mathbf{2 + 1 + 2}]$

(a) Design an NFA with at least one nondeterministic transition and no more than 4 states for the language of the regular expression $\mathbf{1}(\mathbf{01})^*$.

(b) Use subset construction to convert the NFA to a DFA.

(c) Associate set of *dotted items* of the regular expression to every state of the DFA (except the dead/trap state).

Q 4. $[\mathbf{3 + 2}]$

(a) Augment the regular expression $\mathbf{b}(\mathbf{ab})^*$ with # and draw the *syntax tree*. Identify the *nullable nodes* of the tree, label the leaf nodes with $\{1, 2, \cdots\}$, attach *firstpos* and *lastpose* data with every node.

(b) Find *followpos* corresponding to different positions of the syntax tree and draw the NFA using the *followpos* table.

Q 5. $[\mathbf{1 + 2 + 2}]$

Consider the context-free grammar G_5 with S as the *start symbol*. The *producion rules* are, $S \rightarrow A\ a \mid b$ and $A \rightarrow A\ c \mid S\ d \mid \varepsilon$. The language is $L(G_5)$.

(a) Remove ε-rule from G_5 to get a grammar G_{5a} ($L(G_5) = L(G_{5a})$.

(b) Remove *left-recursion* from G_{5a} to get an equivalent grammar G_{5b}.

(c) Is $L(G_5)$ a regular language? Justify your answer.

$* * *$ **End** $* * *$