B.E. COMPUTER SCIENCE AND ENGINEERING
THIRD YEAR
SECOND SEMESTER EXAM 2024
Subject: ***Compiler Design***

Time : Three hours **Full Marks: 100**

Group-1 [CO1] 15 marks	Answer any two questions: [7.5+7.5=15] 1. In Python, the prefixes '0o' and '0x' indicates that the string represents an octal number or a hexadecimal number respectively. Write regular grammars to recognize octal and hexadecimal numbers. Draw a single finite automata for recognizing the above numbers during lexical analysis. 2. Construct a DFA to recognize an even binary number. 3. (a) What are the different phases at the back-end of a compiler? Explain the importance of each. (b) With examples explain the difference between *context free grammar* and *context sensitive grammar*.
Group-2 [CO2] 35 marks	Answer question number 4 and any two from the rest: 4. Python allows to write multiline assignment statements using either '\' or parenthesis '()' For example, `add = 50 + \` `40 - \` `52` OR `add = (50 +` `40 -` `52)` (i) Write a context free grammar for the above syntax for multiline assignment statements. (Use only addition and subtraction operators). (ii) Is your grammar LL(1)? Justify. (iii) Using the grammar defined by you, show the leftmost derivation of at least one string shown above. 10+2+3=15 5. Construct the LR(1) item set for the following grammar: $p \rightarrow b$ $b \rightarrow B\ st\ E$ $st \rightarrow st\ ;\ s \mid s$ $s \rightarrow b \mid v = e$ $v \rightarrow ID \mid ID\ e$ $e \rightarrow e + t \mid t$ $t \rightarrow v \mid v\ (\ e\ )$ (p is the start symbol; b, st, s, v, e, t are non-terminals; B, E, ID, `;', `=', `+', `(' and `)' are terminals.) (meanings are given here: p is program, b is block, st is statement list, s is statement, v is variable, e is expression, t is term, B means `begin', E means `end', ID is identifier, `;', `=', `+', `(' and `)' are operators and punctuations.) 10 6. Consider the following grammar: $S \rightarrow AB$ $A \rightarrow xA$ $A \rightarrow B$ $B \rightarrow yzB$ $B \rightarrow z$ (i) What are the terminals and non-terminals of the above grammar? (ii) Write two sentences (with at least 4 symbols) generated by this grammar and show the rightmost derivations of both strings. (iii) Construct the LL(1) Parsing table of the grammar with error entries. 1+2+7=10

	7. Consider the grammar shown in question 6. (i) Construct the SLR parsing table for the above grammar. (ii) Show the parsing of a sentence generated by the grammar. 8+2=10

Group-3 [CO3 & CO4] 30 marks	Answer any two questions:
	8. (a) What is an attribute grammar? Define S-attributed and L-attributed definitions?
	(b) Consider a grammar for signed binary numbers
	number → *sign list*
	sign → +| -
	list → *list bit* | *bit*
	bit → 0 | 1
	Build attribute grammar that annotates *number* with the value it represents.
	(c) Draw an annotated parse tree and a dependency graph for the binary representation of – 23.
	4+6+5=15
	9. (a) Why is intermediate representation important for a compiler? What are the different intermediate representations?
	(b) Represent the following code as three address code in *triple* format.
	prod = prod + a[i] * b[i];
	i = i +1;
	(c) Consider the following grammar:
	E → E + E | E * E | m E | (E) | id
	Write semantic rules to generate three address code for an expression using the above grammar.
	(m is the unary minus operator)
	(d) Show how the three address code is generated for the expression
	m a * b
	4+3+6+2=15
	10. (a) What do you mean by static and dynamic types in programming languages? What are the advantages of static typing? Give an example of statically typed language.
	(b) What do you mean by a *Sound Type System*?
	(c) Discuss the different implementations of scope rules in symbol tables.
	(d) What do you mean by static scoping and dynamic scoping? Explain with an example.
	4+2+4+5=15

Group-4 [CO 5) 20marks	11. Answer any four questions: [5x4=20] (i) Consider the following code segment: $L1: \quad t1 = -1$ $L2: \quad t2 = 0$ $L3: \quad t3 = 0$ $L4: \quad t4 = 4 * t3$ $L5: \quad t5 = 4 * t2$ $L6: \quad t6 = t5 * M$ $L7: \quad t7 = t4 + t6$ $L8: \quad t8 = a[t7]$ $L9: \quad if\ t8 <= max\ goto\ L11$ $L10: \quad t1 = t8$ $L11: \quad t3 = t3 + 1$ $L12: \quad if\ t3 < M\ gotoL4$ $L13: \quad t2 = t2 + 1$ $L14: \quad if\ t2 < N\ goto\ L3$ $L15: \quad max = t1$

Ref. No.:Ex/CSE/PC/B/T/323/2024

Draw a Control Flow Diagram for the code segment. How many basic blocks do you have in the control flow diagram? How many lines are there in the largest basic block?

(ii) What are loop optimizations? Give two examples of loop optimizations. Explain the benefits of these optimization techniques.

(iii) Optimize the following code and discuss each optimization technique that you have applied stating their advantages:

```
for (j=0; j<n; j++)
{
  d=n%2;
  if (d)
  {
    x = x + 4*j;
    y[j] = x + 4*j;
  }
}
for (j=0; j<n; j++)
{
    a[j] = y[j] * 7 + j * j;
}
```

(iv) Find the liveness and next use for each variable at each statement in the following code.

```
q = x1/x2
t = q * x2
r = x1 – t
x1 = x2
x2 = r
```

(v) What are *register descriptors* and *address descriptors*?

Consider the following code:

```
LD R3, d
LD R2, c
ADD R3, R2
LD R2, e
MUL R3, R2
```

Show the register descriptors for each *register* and *address descriptors* for each variable in the above code.