Total no. of Pages:01 Roll No.....

(72)

FIFTH SEMESTER-B. TECH

MID-SEMESTER EXAMINATION, September, 2022

Course Code: COCSC14/ CACSC14/ CDCSC14

Course Title: Principles of Compiler Construction

Time:1 hr 30 mins. Max.Marks:15

Note: - Attempt all questions. Missing data/information (if any), may be suitably assumed and mentioned in the answer.

Q1	a) Differentiate between Compiler and Interpreter b) With the help of an example, explain phases of compiler.	(1+2)	CO1

Q2	a) Count the number of lexemes and tokens from the following code.
	main ()
	{
	int x=0, y;
	x=x+1;
	y=x+20;
	printf ("%d%d", x,y);
	}
	b) What is the role of input buffering scheme in lexical analyzer? Define token, lexeme and pattern.
		(1+2)	CO2

Q3	a) Use Thompson algorithm to construct an automaton (NFA) for the regular expression (ab*|c)*d*. b) Convert the NFA constructed in part a) to DFA and Minimized DFA.	(1+2)	CO2
Q4	Consider Context Free Grammar G: E → E + T | T T → T F | F F → F * | a | b a) Convert the above grammar G to G' by removing left recursion and perform left factoring, if any b) For the grammar G', find the First and Follow set and derive the predictive parsing table.	(1+2)	CO3
Q5	a) For the grammar G' in question 4 b) show the moves of the predictive parser for the string a + b * b. b) Consider the following Context Free Grammar : S → S S + | S S * | a Draw the parse tree and write the leftmost derivation for the string aa + a *. Find whether this grammar is ambiguous or not, justify. What language is generated by this grammar?	(1+2)	CO3

Total no. of Pages:02 Roll No..... 02 2022

FIFTH SEMESTER-B. TECH

END-SEMESTER EXAMINATION, DECEMBER, 2021

Course Code: CACSC14/ COCSC14

Course Title: Principles of Compiler Construction

Time:3hrs. Max.Marks:40

Note: - Attempt all five questions. Missing data/information (if any), may be suitably assumed and mentioned in the answer.

Q1.	Attempt any two parts of the following	(4+4)	CO1
	a) Draw and explain the block diagram of a Compiler. Indicate the output of every stage corresponding to input z = x + y * v. Differentiate pass and phase of a compiler.		
	b) Explain the language processing system and the role of linker, loader and assembler.		
	c) Explain compiler construction tools.		
Q2.	Attempt any two parts of the following	(4+4)	CO2
	a) Construct the minimized DFA from regular expression a(a|b)*ab using Thompson's construction.		
	b) List out the functions of a Lexical Analyzer? State the reasons for the Separation of Analysis phase into Lexical, Syntax, and Semantic Analysis. Write regular expression for the language containing all strings of 0's and 1's that do not contain 011.		
	c) Write a lexical analyzer for keywords in 'C' language and show it using LEX.		
Q3.	Attempt any two parts of the following	(4+4)	CO3
	a) Differentiate between regular expressions and Context free grammar. Consider the following grammar S–>SS+|SS*|a Show how the string aa+a* can be generated by this grammar. Construct parse tree for this string. Find whether this grammar is ambiguous or not.		
	b) Construct the SLR parsing table for the given grammar and show the parsing moves for the input string "zxxx" S –> AxB S –>Bc A –> yA A –> z B –> xB B –> ε		
	c) Construct unambiguous context-free grammar for Arithmetic Expressions in postfix notation show it using YACC.		
Q4.	Attempt any two parts of the following	(4+4)	CO4

	a) What is basic block? How do you construct basic blocks? Construct basic block and flow graph for the following code
	i=0;
	s=0;
	while (i<10)
	{ s=s+i;
	i=i+1;
	}		

	b) Explain Syntax directed definition and Syntax directed Translation scheme by taking suitable example. Write SDT for converting infix to postfix.		
	c) Explain the different components of activation record. Translate the expression into quadruples, triples and indirect triples. -(a+b) * (c+d) + (a+b+c)		
Q5.	Attempt any two parts of the following	(4+4)	CO4, CO5
	a) What are the issues in generation of target code? Draw directed acyclic graph (DAG) for the following expression. What are the advantages of using DAG. e = (a+b) * (b-c) + (a+b) * (b-c)		

b) Using the Sethi Ullman algorithm generate target code for the following expression assuming only two registers r1 and r2 are available. Explain the steps.

((a+b) - (c*d)) + ((e-f) * (u-v))

c) What is code optimization? What are the issues associated with code optimization? Explain various code optimization techniques

Total No of Pages: 02 Roll No. _____

SIXTH SEMESTER B. Tech. (IT)
END SEMESTER EXAMINATION, May-2022

Course Code: **ITITC20**
Course Title: **Compiler and Translator Design**
Time: 3:00 Hours Max. Marks: 40

Note: Attempt All the Five questions. Missing data/information if any may be suitably assumed & mentioned in the answer.

Q.No	Questions	Marks	COs
Q 1	Attempt any **two** parts of the following:		
1a	Differentiate between compiler and interpreter. Explain the need for dividing the compilation process into various phases and explain its functions.	4	1
1b	Construct the NFA for regular expression (aa*|bb*) using Thompson's Construction and convert it into DFA.	4	2
1c	Convert the regular expression (a|b).(ab)*.b into DFA without constructing NFA (By computing Firstpos, Nullable, Followpos...)	4	2
Q 2	Attempt any **two** parts of the following:		
2a	Consider the following grammar: S → Aa | bAc | Bc | bBa A → d B → d Construct the collection LR(1) items and its parsing table for the above grammar.	4	3
2b	Show all moves of the above LR(1) parser for an input string "dc".	4	3
2c	Explain error recovery in predictive parsing.	4	3
Q 3	Attempt any **two** parts of the following:		
3a	Write the grammar and syntax-directed translation for desk-calculator.	4	4
3b	Show the annotated parse tree for expression (3 + 4)*(5 + 6).	4	4
3c	Translate the expression (a + b)*(c + d) + (a + b + c) into quadruples, triples and indirect triples.	4	5

103

Total No. Pages: 1
SIXTH SEMESTER

Roll No.---------
B.E. (IT)

MID SEMESTER EXAMINATION, Febuary-2020

ITC20: Compiler and Translator Design

Time: 1:30 Hrs

Max. Marks: 25

Note: All questions are compulsory. Assume suitable missing data, if any.

1. Give formal definition with an example: Regular Expression, Regular Definitions, Lexemes, Token and Pattern. [5]

2. What is a primary function of a compiler? What are its secondary functions? [2]

3. (a) Consider the below grammar:

$E \rightarrow 5 + T \mid 3 - T$
$T \rightarrow V \mid V*V \mid V + V$
$V \rightarrow a \mid b$

(i) Do left factoring.
(ii) Find First and Follow of non-terminals. [1+3]

4. Construct NFA for the regular expression aa*|bb* using Thompson's construction. Also construct the corresponding DFA from the above NFA. [4]

5. Write the two methods used in a lexical analyzer for buffering the input. Which method is used for speeding up the lexical analyzer? [4]

6. Construct the SLR parsing table for the following augmented grammar:

$S' \rightarrow S$
$S \rightarrow aABe$
$A \rightarrow Abc$
$A \rightarrow b$
$B \rightarrow d$

where S, A and B are non-terminals, and a, b, c, d and e are terminals. Also, parse the string abbcde using SLR parser. [6]

----------x----------

Total no. of Pages:01 Roll No.....

FIFTH SEMESTER-B. TECH

END-SEMESTER EXAMINATION, Nov-Dec, 2022

Course Code: CACSC14/ CDCSC14/ COCSC14

Course Title: Principles of Compiler Construction

Time:3hrs. Max.Marks:40

Note:-Attempt any two parts from each question. Assume missing data/information and mention in the answer (if any)

Q1	a)	What is the role of symbol table in compiler. Describe different data structures used for symbol table implementation.	(4+4)	CO1 CO2
	b)	What is type checking and how it is implemented in compiler.		
	c)	What is the function of preprocessor, assembler, linker and loader.		
Q2	a)	Construct minimum state DFA for the regular expression (0|1)*010. Describe the language denoted by the given regular expression.	(4+4)	CO2 CO3
	b)	What is a parser, explain the drawbacks of top-down parsing. Explain with the help of an example shift-reduce conflict in bottom-up parsing.		
	c)	Describe in brief LEX and YACC tools in compiler construction.		
Q3	a)	What are different types of errors in a program. State the function of error handler. How error-recovery is performed in bottom-up parsing.	(4+4)	CO2 CO3
	b)	Show that the following grammar S → Aa | bAc | dc | bda A → a is LALR(1) but not SLR(1)		
	c)	Construct SLR parsing table for the following grammar E → E + T | T T → T*F | F F → (E) | id | id++		
Q4	a)	What is the need of generating intermediate-code. Translate the following expression in Three-address code as Quadruples, Triples and Indirect Triples -(a+b) * (a+b+c) – (b+c)	(4+4)	CO3 CO4
	b)	What is DAG? How DAG is different from syntax tree. Construct the DAG for the following T1 = a + b T2 = a – b T3 = T1 * T2 T4 = T1 – T3 T5 = T4 + T3		
	c)	Explain the difference among S-attribute and L-attribute? Write the semantic rules for the following grammar showing appropriate attributes. S → TL T→int T→float T→char T→double L→L,id L→id		
Q5	a)	Write the pseudo code for bubble sort. Find basic blocks and draw flow graph for the same.	(4+4)	CO4 CO5
	b)	Differentiate between local and global code optimization. Explain the following with an example i. Dead code elimination ii. Copy propagation, constant folding. iii. Strength Reduction		
	c)	Generate code for the following expression using labelling algorithm and code generation for labelled tree (Sethi Ullman algorithm) assuming 2 registers are available. x*y+(u-v*w)		

SIXTH SEMESTER B. Tech. (IT)

MID-SEMESTER EXAMINATION, FEB-MARCH 2022

Course Code: **ITITC20**
Course Title: **Compiler and Translator Design**

Time- 1:30 Hours | Max. Marks- 15

Note: - Attempt all questions. Missing data/ information if any, may be suitably assumed & mentioned in the answer.

Q. No.	Question	Marks	CO
1a	Write down the phases of a compiler. What is the advantage of dividing the design of a compiler into front-end design and back-end design?	1.5	CO1
1b	Define the following terms with an example: Token, Lexeme and Regular Definitions	1.5	CO2
2a	Construct NFA for the regular expression (a|b)*ab using Thompson's construction. Construct the corresponding DFA from the above NFA	1.5	CO2
2b	Define L-attributed definitions with an example.	1.5	CO4
3a	Remove the left recursion for the below grammar: exp → exp addop term | term addop → + | - term → term mulop factor | factor mulop → * factor → (exp) | number where exp, addop, term, mulop and factor are non-terminals, and +, -, *, (,) and number are terminals.	1.5	CO3
3b	Find the First and Follow of non-terminals for the above grammar given in Q. No. 3a.	1.5	CO3
4a	Test whether the grammar is LL(1) or not, and construct a predictive parsing table for the following grammar: S → A A → aB | Ad B → bBC | f C → g	1.5	CO3
4b	Parse the input string **abfg** using non-recursive predictive parser for the above grammar given in Q. No. 4a.	1.5	CO3
5a	Construct the SLR parsing table for the following augmented grammar: S' → S S → aABe A → Abc A → b B → d where S, A and B are non-terminals, and a, b, c, d and e are terminals.	1.5	CO3
5b	Also, parse the string **abbcde** using SLR parser for the above grammar given in Q. No. 5a.	1.5	CO3

SIXTH SEMESTER B. Tech. (IT)
END SEMESTER EXAMINATION, May-2022

Course Code: **ITITC20**
Course Title: **Compiler and Translator Design**
Time: 3:00 Hours

Max. Marks: 40

Note: Attempt All the Five questions. Missing data/information if any may be suitably assumed & mentioned in the answer.

Q.No	Questions	Marks	COs
Q 1	Attempt any **two** parts of the following:		
1a	Differentiate between compiler and interpreter. Explain the need for dividing the compilation process into various phases and explain its functions.	4	1
1b	Construct the NFA for regular expression (aa*|bb*) using Thompson's Construction and convert it into DFA.	4	2
1c	Convert the regular expression (a|b).(ab)*.b into DFA without constructing NFA (By computing Firstpos, Nullable, Followpos...)	4	2
Q 2	Attempt any **two** parts of the following:		
2a	Consider the following grammar: $S \rightarrow Aa \mid bAc \mid Bc \mid bBa$ $A \rightarrow d$ $B \rightarrow d$ Construct the collection LR(1) items and its parsing table for the above grammar.	4	3
2b	Show all moves of the above LR(1) parser for an input string "dc".	4	3
2c	Explain error recovery in predictive parsing.	4	3
Q 3	Attempt any **two** parts of the following:		
3a	Write the grammar and syntax-directed translation for desk-calculator.	4	4
3b	Show the annotated parse tree for expression (3 + 4)*(5 + 6).	4	4
3c	Translate the expression (a + b)*(c + d) + (a + b + c) into quadruples, triples and indirect triples.	4	5

148

Q 4	Attempt any **two** parts of the following:		

4a	What is a basic block? Construct basic blocks and flow diagram for the following code:
	double fun(int A[], int n)
	{
	int i, sum;
	double average = 0;
	for (i = 0; I < n; i++)
	sum = sum + A[i];
	average = sum/n;
	}
		4	5

4b	Explain peephole optimization technique.	4	6
4c	Compare the three different storage allocation strategies.	4	1
Q 5	Attempt any **two** parts of the following:		
5a	Discuss dead code elimination and code motion.	4	6
5b	Differentiate between copy propagation and constant propagation. What are the benefits of these two methods with respect to optimization?	4	6
5c	Discuss loops in flow graphs with an example.	4	6

---END--

Total No. of Pages: 01 Roll No. .....

FIFTH SEMESTER BE (COE)

MID SEMESTER EXAMINATION, SEPTEMBER 2018

CEC17 COMPILER CONSTRUCTION

Time: 1 1/2 Hrs. Max. Marks: 15

Note: Attempt all questions. Assume missing data, if any.

1. Define what is a compiler. Describe the phases of compiler. [2]

2. Explain how input buffering is used in lexical analysis. [2]

3. Consider the following grammar [2+2]

A → A or B |B
B → B and C |C
C → id | not id | (A)

a) Draw parse tree and write the leftmost derivation and rightmost derivation for the string **not id and (id or id)**

b) Remove left recursion from the above grammar.

4. Consider the following grammar [2]

A → A or A |A and A |id |not id |(A)

Is this grammar ambiguous? Can this grammar be left factored? Justify.

5. Differentiate between lexemes, patterns and tokens. Use McNaughton-Yamada-Thompson algorithm to construct a minimum state DFA for ((a|b)c(a|b*))*. [5]

No. of Page(s): 1 Roll No. .....

FIFTH SEMESTER B.E. (COE)

B.E. MID-SEMESTER EXAMINATION, September 2019

CEC17 Compiler Construction

Time: 1:30 Hrs. Max. Marks: 15

1. (a) Differentiate between compiler and interpreter.
   (b) What are patterns, lexemes and tokens?
   (c) How LEX is related to lexical analysis?
   (d) Explain how input buffering is used in lexical analysis.

[4x1]

2. (b) Construct an NFA for the following regular expression.

   (a|b)(ba|ab*)*

   Convert the NFA into a DFA and then minimize the number of states in it.

   (c) Is the following grammar ambiguous? Justify your answer.

   E → E or E | E xor E | E and E | not E | (E) | id

   Also write the leftmost derivation and draw the parse tree for the string

   not(id or id)

[3+3]

3. Construct the predictive parsing table for the following grammar.

   A → A + B | B
   B → B * C | C
   C → (A) | id | id ++ | id --

   Use the parsing table to parse the string id * (id + id ++)

[5]

----X----

Total no. of Pages:02 Roll No.....

FIFTH SEMESTER-B. TECH

END-SEMESTER EXAMINATION, DECEMBER, 2021

Course Code: CACSC14/ COCSC14

Course Title: Principles of Compiler Construction

Time:3hrs. Max.Marks:40

Note: - Attempt all five questions. Missing data/information (if any), may be suitably assumed and mentioned in the answer.

Q1.	Attempt any two parts of the following a) Draw and explain the block diagram of a Compiler. Indicate the output of every stage corresponding to input z = x + y * v. Differentiate pass and phase of a compiler. b) Explain the language processing system and the role of linker, loader and assembler. c) Explain compiler construction tools.	(4+4)	CO1
Q2.	Attempt any two parts of the following a) Construct the minimized DFA from regular expression a(a|b)*ab using Thompson's construction. b) List out the functions of a Lexical Analyzer? State the reasons for the Separation of Analysis phase into Lexical, Syntax, and Semantic Analysis. Write regular expression for the language containing all strings of 0's and 1's that do not contain 011. c) Write a lexical analyzer for keywords in 'C' language and show it using LEX.	(4+4)	CO2
Q3.	Attempt any two parts of the following a) Differentiate between regular expressions and Context free grammar. Consider the following grammar S->SS+|SS*|a Show how the string aa+a* can be generated by this grammar. Construct parse tree for this string. Find whether this grammar is ambiguous or not. b) Construct the SLR parsing table for the given grammar and show the parsing moves for the input string "zxxx" S -> AxB S ->Bc A -> yA A -> z B -> xB B -> ε c) Construct unambiguous context-free grammar for Arithmetic Expressions in postfix notation show it using YACC.	(4+4)	CO3

Q4.	Attempt any two parts of the following
	a) What is basic block? How do you construct basic blocks? Construct basic block and flow graph for the following code
	i=0;
	s=0;
	while (i<10)
	{ s=s+i;
	i=i+1;
	}
	b) Explain Syntax directed definition and Syntax directed Translation scheme by taking suitable example. Write SDT for converting infix to postfix.
	c) Explain the different components of activation record. Translate the expression into quadruples, triples and indirect triples.
	-(a+b) * (c+d) + (a+b+c)
		(4+4)	CO4

Q5.	Attempt any two parts of the following a) What are the issues in generation of target code? Draw directed acyclic graph (DAG) for the following expression. What are the advantages of using DAG. e = (a+b) * (b-c) + (a+b) * (b-c)	(4+4)	CO4, CO5

b) Using the Sethi Ullman algorithm generate target code for the following expression assuming only two registers r1 and r2 are available. Explain the steps. ((a+b) - (c*d)) + ((e-f) * (u-v)) c) What is code optimization? What are the issues associated with code optimization? Explain various code optimization techniques		