Reg No.:_____ Name:_____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION, APRIL 2018

Course Code: CS304

Course Name: COMPILER DESIGN (CS, IT)

Max. Marks: 100 Duration: 3 Hours

PART A

Answer all questions, each carries 3 marks. Marks

1 Draw the transition diagram for the regular definition, (3)

relop→< | <= | = | <> | >= | >

2 With an example source language statement, explain tokens, lexemes and patterns. (3)

3 Define LL(1) grammars. (3)

4 Is the grammar S→S(S)S/ Ɛ ambiguous? Justify your answer. (3)

PART B

Answer any two full questions, each carries 9 marks.

5 a) Apply bootstrapping to develop a compiler for a new high level language P on machine N. (3)

b) Now I have a compiler for P on machine N. Apply bootstrapping to obtain a compiler for P on machine M. (4)

c) Define cross-compilers. (2)

6 a) Consider the following grammar

E→ E or T | T
T→ T and F |F
F→ not F | (E) | true | false (2)

(i) Remove left recursion from the grammar. (4)
(ii) Construct a predictive parsing table. (3)
(iii) Justify the statement " The grammar is LL (1)".

7 a) Design a recursive descent parser for the grammar S→cAd, A→ab/ b (5)

b) For a source language statement a= b*c - 2, where a, b and c are float variables, * and – represents multiplication and subtraction on same data types, show the input and output at each of the compiler phases. (4)

PART C

Answer all questions, each carries 3 marks.

8 Compute the FIRST and FOLLOW for the following Grammar. (3)

S→Bb/Cd B→aB/Ɛ C→cC/Ɛ

9 Demonstrate the identification of handles in operator precedence parsing? (3)

10 Design a Syntax Directed Definition for a Desk calculator that prints the result. (3)

11 Describe the type checking of functions. (3)

PART D

Answer any two full questions, each carries 9 marks.

12 a) Construct canonical LR(0) collection of items for the grammar below. (5)

S → L = R
S → R
L → * R
L → id
R → L

Also identify a shift reduce conflict in the LR(0) collection constructed above.

b) Define S-attributed and L-attributed definitions. Give an example each. (4)

13 a) Explain bottom- up evaluation of S- attributed definitions. (5)

b) With an SDD for a desk calculator, give the appropriate code to be executed at each reduction in the LR parser designed for the calculator. Also give the annotated parse tree for the expression (3*5) –2. (4)

14 a) Construct LALR parse table for the grammar S->CC,C->cC|d (9)

PART E

Answer any four full questions, each carries 10 marks.

15 a) Write syntax directed definitions to construct syntax tree and three address code for assignment statements. (10)

16 a) Explain quadruples and triples with an example each. (5)

b) Construct the syntax tree and then draw the DAG for the statement (5)

e := (a*b) + (c-d) * (a*b)

17 a) Explain static allocation and heap allocation strategies. (10)

18 a) With an example each explain the following loop optimization techniques: (i) Code motion (ii) Induction variable elimination and (iii) strength reduction (10)

19 a) Explain any two issues in the design of a code generator. (5)

b) Explain the optimization of basic blocks. (5)

20 a) Write the Code Generation Algorithm and explain the *getreg* function. (6)

b) Generate a code sequence for the assignment d=(a-b)+(a-c)+(a-c) (4)

****

Reg No.:_____ Name:_____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION(R&S), MAY 2019

Course Code: CS304

Course Name: COMPILER DESIGN

Max. Marks: 100 Duration: 3 Hours

PART A

Answer all questions, each carries3 marks. Marks

1		Describe input buffering scheme in lexical analyzer.	(3)
2		Construct a regular expression to denote a language L over ∑= {0,1} accepting all strings of 0's and 1's that do not contain substring 011	(3)
3		Consider the context free grammar S->aSbS | bSaS | € Check whether the grammar is ambiguous or not	(3)
4		What is Recursive Descent parsing? List the problems faced in designing such a parser.	(3)

PART B

Answer any two full questions, each carries9 marks.

5	a)	Explain the different phases in the design of a compiler.	(5)
	b)	Find the FIRST and FOLLOW of the non-terminals in the grammar S->aABe A->Abc|b B->d	(4)
6	a)	Design a recursive descent parser for the grammar E->E + T | T T->T*F | F F->(E) | id	(5)
	b)	Develop a lexical analyzer for the token identifier.	(4)
7	a)	What is left recursive grammar? Give an example. What are the steps in removing left recursion?	(5)
	b)	Explain any four compiler writing tools	(4)

PART C

Answer all questions, each carries3 marks.

8		Explain the main actions in a shift reduce parser	(3)
9		What are different parsing conflicts in SLR parsing table?	(3)
10		What are annotated parse trees? Give examples.	(3)
11		What are L-attributed definitions and S-attributed definitions in a syntax directed translation scheme?	(3)

PART D

Answer any two full questions, each carries9 marks.

12	a)	Find the LR(0) items for the grammar S->SS | a| €.	(4)
	b)	Explain bottom- up evaluation of s-attributed definitions.	(5)
13	a)	Derive LALR (1) parsing algorithm for following grammar S→AS/b A→SA/a	(6)
	b)	Design a type checker for simple arithmetic operations.	(3)
14	a)	Explain the syntax directed definition of a simple desk calculator.	(5)
		Explain operator grammar and operator precedence parsing	(4)

PART E

Answer any four full questions, each carries10 marks.

15	a)	Explain storage organization and storage allocation strategies	(10)
16	a)	Explain intermediate code generation of an assignment statement	(10)
17	a)	Explain quadruples, triples and dags with an example each.	(10)
18	a)	Explain the principal sources of optimization	(10)
19	a)	Explain optimization of basic blocks	(5)
	b)	With suitable examples explain loop optimization.	(5)
20	a)	Explain issues in design of a code generator	(5)
	b)	Explain simple code generation algorithm	(5)

****

Reg No.:_____ Name:_____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION(S), DECEMBER 2019

Course Code: CS304

Course Name: COMPILER DESIGN

Max. Marks: 100 Duration: 3 Hours

PART A

Answer all questions, each carries3 marks. Marks

1 Scanning of source code in compilers can be speeded up using input buffering. Explain. (3)

2 Draw the DFA for the regular expression (a | b)* (abb | a+ b). (3)

3 Differentiate leftmost derivation and rightmost derivation. Show an example for each. (3)

4 Find out context free language for the grammar given below: (3)

S -> abB
A -> aaBb | ε
B -> bbAa

PART B

Answer any two full questions, each carries9 marks.

5 a) Explain compiler writing tools. (5)

b) Given a grammar : (4)

S→ (L)|a

L→ L,S | S

(i) Is the grammar ambiguous? Justify

(ii) Give the parse tree for the string (a,((a,a), (a,a)))

6 a) Construct the predictive parsing table for the following grammar: (5)

S -> (L) | a
L -> L,S | S

b) Explain how the regular expressions and finite state automata are used for the specification and recognition of tokens? (4)

7 a) Explain the working of different phases of a compiler. Illustrate with a source language statement. (5)

b) Can recursive descent parsers used for left recursive grammars? Justify your answer. Give the steps in elimination of left recursion in a grammar. (4)

PART C

Answer all questions, each carries3 marks.

8		Compute FIRST and FOLLOW for the grammar: S -> SS+ | SS* | a	(3)
9		Write the algorithm to construct LR(1) collection for a grammar.	(3)
10		What is an SDD? Show an example.	(3)
11		Distinguish between synthesized and inherited attributes.	(3)

PART D

Answer any two full questions, each carries9 marks.

12	a)	Write algorithm for SLR paring table construction.	(5)
	b)	Construct syntax directed translation scheme for infix to postfix translation.	(4)
13	a)	Construct the SLR table for the grammar: S -> aSbS | a	(5)
	b)	Give the annotated parse tree for the expression: 1*2*3*(4+5) **n**	(4)
14	a)	Differentiate CLR and LALR parsers.	(4)
	b)	Explain the specification of a simple type checker.	(5)

PART E

Answer any four full questions, each carries10 marks.

15	a)	Explain how DAGs help in intermediate code generation?	(4)
	b)	Explain the code generation algorithm. Illustrate with an example.	(6)
16	a)	Define the following and show an example for each. i). Three-address code iii). Triples ii). Quadruples iv). Indirect triples	(6)
	b)	State the issues in design of a code generator.	(4)
17	a)	Explain different stack allocation strategies with suitable examples.	(10)
18	a)	Explain different code optimization techniques available in local and global optimizations?	(10)
19	a)	How is storage organization and management done during runtime?	(4)
	b)	How the optimization of basic blocks is done by a compiler?	(6)
20	a)	Write the algorithm for partitioning a sequence of three-address instructions into basic blocks.	(4)
	b)	Construct the DAG and three address code for the expression a+a*(b-c)+(b-c)*d	(6)

Reg No.:_____ Name:_____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Sixth semester B.Tech examinations (S), September 2020

Course Code: CS304
Course Name: COMPILER DESIGN

Max. Marks: 100 Duration: 3 Hours

PART A

Answer all questions, each carries 3 marks. Marks

1 State the role of lexical analyzer. Identify the lexemes and their corresponding tokens in the following statement: printf ("Simple Interest=%f\n", si); (3)

2 Explain any three tools that help a programmer in building a compiler efficiently. (3)

3 Eliminate the ambiguity from the given grammar (3)

E → E*E | E-E | E^E | E/E | E+E | (E) | id.

The associativity of the operators is as given below. The operators are listed in the decreasing order of precedence.

(i) ( )

(ii) / and + are right associative

(iii) ^ is left associative.

(iv) * and – are left associative

4 For what type of grammar, recursive descent parser cannot be constructed? Show the steps involved in recursive descent parsing with backtracking for the string *cad* with the given grammar: S -> cAd A -> ab | a (3)

PART B

Answer any two full questions, each carries 9 marks.

5 a) Trace the output after each phase of the compiler for the assignment statement: **a = b + c * 10**, if variables given are of float type. (6)

b) Show that the following grammar is ambiguous. (3)

bexpr → *bexpr* **OR** *bterm* | *bterm*

bterm → *bterm* **AND** *bfactor* | *bfactor*

bfactor → **NOT** *bfactor* | *(bexpr)* | **TRUE** | **FALSE**

6 a) Left factor the following grammar and then obtain LL(1) parsing table (6)

$E \rightarrow T+E \mid T$

$T \rightarrow float \mid float * T \mid (E)$

b) What is the relevance of input buffering in lexical analysis? (3)

7 a) Write Non-recursive predictive parsing algorithm. (5)

b) Write regular expressions for the following languages: (4)

i) All strings over the English alphabet that contain the five vowels in order.

ii) All strings of a's and b's that do not contain the subsequence *abb*.

PART C

Answer all questions, each carries 3 marks.

8 What is handle pruning? Indicate the handles in the reduction of the right sentential form **S S+ a *** to the start symbol using the grammar below: (3)

S -> S S+ | S S * | a

9 What are viable prefixes? For the given grammar $S \rightarrow 0\ S\ 1 \mid 0\ 1$ write all the viable prefixes for the string 00001111 (3)

10 Give the S-attributed SDD of a simple desk calculator and show annotated parse tree for the expression (3+4)*(5+6). (3)

11 Write a translation scheme for performing type checking of statements. (3)

PART D

Answer any two full questions, each carries 9 marks.

12 a) Construct canonical collection of LR(1) items for the following grammar: (5)

$S \rightarrow AA$

$A \rightarrow Aa \mid b$

b) Differentiate between S-attributed and L-attributed definitions with suitable examples. (4)

13 a) Write the SDD for a simple type declaration and draw the annotated parse tree for the declaration **float a, b, c**. (5)

b) Construct SLR parsing table for the grammar $A \rightarrow a \mid (A)$. (4)

14 a) Using operator precedence relations, parse the string id + (id * id). (5)

b) Construct DAG for the expression (a/10 + (b -10))*(a/10 + (b-10)). Also write the sequence of instructions used for the DAG construction. (4)

PART E

Answer any four full questions, each carries 10 marks.

15 a) Using necessary figure, illustrate how the caller and callee cooperate in (6)

managing various tasks in stack allocation strategy when a procedure is activated.

b) Explain copy propagation with an example. (4)

16 a) Write SDD to produce three-address code for Boolean expressions and obtain the three-address code for the statement given below: (6)

```
while a < b do
    if c < d then
       x = y + z
    else
       x = y – z
```

b) Explain common sub expression elimination with an example. (4)

17 a) Identify any four issues in the design of a Code Generator. (6)

b) Write the three address code sequence for the statement x=y*z + y*-z. Also give its triple representation. (4)

18 Write the code generation algorithm. Using this algorithm generate code sequence for the expression **x = (a – b) + (a + c)**. (10)

19 a) With suitable example of a basic block, explain the code-improving transformations of a basic block. (6)

b) Describe the various fields in an activation record. (4)

20 a) Explain the 3 representations of three-address code statements. (6)

b) What is static allocation strategy? What are its limitations? (4)

****