C **Reg. No. :** | | | | | | | | | | |

Question Paper Code: 56201

B.E./B.Tech. DEGREE EXAMINATION, APRIL 2019

Sixth Semester

Computer science and Engineering

15UCS601- PRINCIPLES OF COMPILER DESIGN

(Regulation 2015)

Duration: Three hours Maximum: 100 Marks

Answer ALL Questions

PART A - (5 x 1 = 5 Marks)

1. There are some Context Free Grammars for which shift reduce parsing cannot be used. Why? CO1- R
   (i) Cannot decide whether to shift or to reduce
   (ii) Cannot decide which of several reductions to make

   (a) Only i (b) only ii (c) i and ii (d) None of the above

2. What is the technique have been developed to reduce the amount of overhead required to process a single input character? CO2- R
   (i) Single Buffer Scheme
   (ii) Two-Buffer Scheme

   (a) Only i (b) only ii (c) i and ii (d) None of the above

3. What are the functions used to manipulate list of jumps in One-Pass code generation using Backpatching? CO3- R

   (a) makelist(), merge(), backpatch() (b) makelist(), translate(), backpatch()

   (c) makelist(), backpatch() (d) translate(), backpatch()

4. In three-address code, how many operator must present on the right side of the instruction CO4- R

   (a) At most one (b) At least one (c) More than one (d) None of the above

5. What kind of information useful for locality optimization and parallelizing compiler? CO5- R

(i) Data reuse

(ii) Data dependence

(a) Only i (b) only ii (c) i and ii (d) None of the above

PART – B (5 x 3= 15 Marks)

6. Draw the transition diagram for relational operator CO1- R

7. Compute FIRST and FOLLOW for the grammar CO2- R

S → SS+ | SS*| a

8. Translate the arithmetic expression a+ - (b+c) into a syntax tree and triples CO3- R

9. Apply the possible optimization technique to the following code CO4- R

```
dp=0
i=0
l: t1=i*8
t2=A[t1]
t3=i*8
t4=B[t3]
t5=t2*t4
dp=dp+t5
i=i+1
if i<n goto L
```

10. Explain the types of data reuse. CO5- R

PART – C (5 x 16= 80 Marks)

11. (a) Illustrate the various phases of compiler for the statement s=x-y/38 CO1- App (16)

Or

(b) (i) Construct the automata which accepts all strings of regular expression a(a/b)*a CO1- App (8)

(ii) Explain with necessary examples about lexical errors and its error recovery actions. CO1- App (8)

12.	(a)	Construct the SLR parser for the following grammar and check whether the string 0001111 is accepted or not ? S →AS | ε A → 0A1 | A1 | 01	CO2- App	(16)
		Or		
	(b)	Illustrate how error recovery is done in predictive parser for an erroneous input id*+id of the grammar E→E+T|T T → T*F | F F →(E)|id	CO2- App	(16)
13.	(a)	Explain the semantic actions required for translation of array reference.	CO3- U	(16)
		Or		
	(b)	Explain in detail about a simple code generator with necessary examples. Demonstrate the changes in register and address descriptors.	CO3- U	(16)
14.	(a)	Explain the ways in which the storage allocation is done in stack heap memory management.	CO4 U	(16)
		Or		
	(b)	Explain the concept of back patching for (i) Boolean expressions	CO4- U	(8)
		(ii) Flow-of-control statements	CO4- U	(8)
15.	(a)	Illustrate a data-flow analysis schemas for an example flow graph. Demonstrate how IN, OUT, GEN and KILL of each basic block can be found.	CO5 U	(16)
		Or		
	(b)	Construct iteration spaces for the multiplication of two matrices.	CO5- U	(16)