APJ ABDULKALAM TECHNOLOGICAL UNIVERSITY
08 PALAKKAD CLUSTER

Q. P. Code : 08CSE19-6012-1 **(Pages: 4)** **Name: .....**

Reg. No:.....

SECOND SEMESTER M.TECH. DEGREE EXAMINATION MAY 2019

Branch: Computer Science and Engineering Specialization: Computer Science and Engineering

08 CS 6012: ADVANCED COMPILER DESIGN

(Common to CSE)

Time:3 hours **Max. marks: 60**

Answer all six questions.

Modules 1 to 6: Part 'a' of each question is compulsory and answer either part 'b' or part 'c' of each question.

Q.no.	Module 1	Marks
1.a	Suppose that we have a production A->BCD. Each of the non-terminals has two attributes *s* as a synthesized attribute and *i* as an inherited attribute. For each sets of rule given below tell whether i) Consistent with an S-attributed definition ii) Consistent with an L-attributed definition iii) Whether the rules are inconsistent with any order. 1) A.s= B.i +C.s and D.i= A.i +B.s 2) A.s =D.i, B.i=A.s+C.s, C.i=B.s 3) A.s =B.s +D.s	**3**
	Answer b or c	
b	Write SDT analogous to the following production, which represents a familiar flow-of-control construct S -> **do** S1 **while** (C).Also revise the above SDT for on-the fly code generation for do-while statements.	**6**
c	Give L-attributed SDD to construct syntax trees during top-down parsing and with the SDD draw dependency graph and write all the topological sorts for the expression **a-4+c**.	**6**

Q.no.	Module 2	Marks
2.a	Show how to transform a three-address code sequence into one in which each defined variable gets a unique variable name.	**3**
	Answer b or c	

b Translate the expression **c + a[i][j]**, assuming the width of an integer is 4 into the sequence of three-address code and draw annotated parse tree for the given expression. **6**

c Write translation scheme for Backpatching Boolean expressions. Using the translation, translate each of the following expressions. Show the true and false lists for each subexpression. You may assume the address for the first instruction generated is 100. **6**

1) (a= =b || c= =d) || e= =f 2) a= =b && (c= =d || e= =f)

Q.no. **Module 3** **Marks**

3.a In the C code given below to compute Fibonacci numbers recursively. **3**

```
i nt f ( i nt  n) {
i nt  t , s;
if (n < 2) r e t u r n 1;
s = f ( n - l );
t = f ( n - 2 );
r e t u r n s+t; }
```

Suppose that the activation record for f includes the following elements in order (return value, argument n, local *s*, local *t);* there will normally be other elements in the activation record as well. The questions below assume that the initial call is f (5).

i) Show the complete activation tree.

ii) What does the stack and its activation records look like the first time f(l) is about to return?

iii) What does the stack and its activation records look like the fifth time f(l) is about to return?

Answer b or c

b With suitable example of network of objects explain Baker mark-and-sweep algorithm which moves objects among lists. **6**

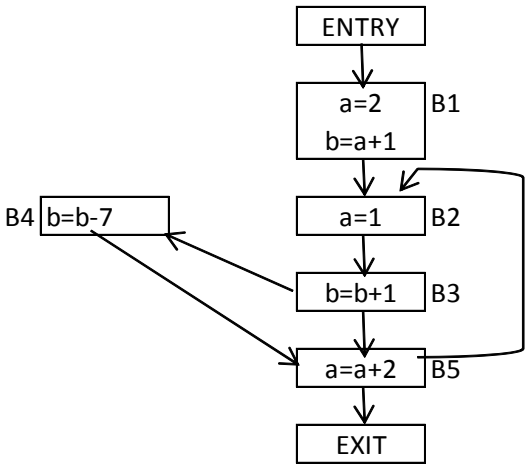
c Using appropriate example explain the storage management technique that is used for data that lives indefinitely, or until the program explicitly deletes it. **6**

Q.no.	Module 4	Marks
4.a	Generate code for the following three-address statements assuming a and b are arrays whose elements are 4-byte values .Also determine the costs of the instruction sequences generated. x= a[i] y= b[i] z= x*y	**3**

Answer b or c

b	Construct the DAG for the basic block d = b * c **e** = a + b b = b * c a = **e** – d Simplify the three-address code given above**,** assuming i) Only *a* is live on exit from the block. ii) *a, b,* and c are live on exit from the block.	**6**
c	Generate the target machine instructions using tree- rewriting rules or by tiling an input tree. Compute Ershov numbers for the given expression a+b*(c*(d+e))	**6**

Q.no.	Module 5	Marks
5.a	For the flow graph given below compute (i) The *gen* and *kill* sets for each block. (ii) The *IN* and *OUT* sets for each block.	**4**



Answer b or c

b Determine the four data flow passes in partial-redundancy elimination. **8**

c Illustrate how region-based analysis works based on the example of reaching definitions. **8**

Q.no.	Module 6	Marks

6.a Given a sequence of assignments exhibiting data dependencies: **4**

1) a = b

2) c = d

3) b = c

4) d = a

5) c = d

6) a = b

Classify above dependencies for each pairs of statements below as (i) true dependence (ii) antidependence (iii) output dependence or (iv) no dependence. Give reasons for each.

1) Statements (1) and (4)

2) Statements (3) and (5)

3) Statements (1) and (6)

4) Statements (3) and (6)

5) Statements (4) and (6)

Answer b or c

b For each of the code fragments given below, draw the dependence graph and write algorithm for list scheduling a basic block. **8**

	(a)	(b)	(c)
1)	LD R1, a	LD R1, a	LD R1, a
2)	LD R2, b	LD R2, b	LD R2, b
3)	SUB R3, R1, R2	SUB R1, R1, R2	SUB R3, R1, R2
4)	ADD R2, R1, R2	ADD R2, R1, R2	ADD R4, R1, R2
5)	ST a, R3	ST a, R1	ST a, R3
6)	ST b, R2	ST b, R2	ST b, R4

c Using globally scheduled machine codes explain data dependences and control dependences to get more parallelism. **8**