



Reg.No

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

SNS College of Technology, Coimbatore-35.

(Autonomous)

B.E/B.Tech- Internal Assessment -III

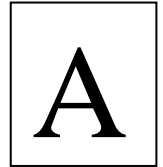
Academic Year 2023-2024(ODD)



Fifth Semester

Computer Science and Engineering

19CSB301 – Automata Theory and Compiler Design



Time: 1.5 Hours

Maximum Marks: 50

Part-A (5 x 2 =10Marks)

	CO	Blooms
1. Infer about back patching Backpatching is basically a process of fulfilling unspecified information . This information is of labels. It basically uses the appropriate semantic actions during the process of code generation. It may indicate the address of the Label in goto statements while producing TACs for the given expressions.	CO4	UND
2. List three kind of intermediate representation Syntax Tree Postfix Notation Three address Statement	CO4	REM
3. Define Dead Code Dead code includes code that can never be executed (unreachable code), and code that only affects dead variables (written to, but never read again), that is, irrelevant to the program.	CO5	REM
4. What is Peephole optimization? Peephole optimization is a simple and effective technique for locally improving target code. Redundant Instruction Elimination Unreachable code	CO5	REM
5. Outline about global data flow analysis To efficiently optimize the code compiler collects all the information about the program and distribute this information to each block of the flow graph. This process is known as data-flow graph analysis. Certain optimization can only be achieved by examining the entire program.	CO5	UND

Part-B (2x13+14=40Marks)

6. a. Construct the CLR parsing table for the following grammar. check whether the string (a) is accepted or not. 13 C A
O4 P
P
- S -> (L)|a
L->L,S|S

1. Construct Augmented Grammar

2. Construct Canonical LR(1) items

3. Construct CLR Parsing table

4. Parsing i/p string using CLR Parse table

After the elimination of left recursion: (2)

$S \rightarrow (L) | a$
 $L \rightarrow SLi$
 $Li \rightarrow ,SLi | \hat{A}$

Calculation of First: (2)

$First(S) = \{(, a\}$
 $First(L) = \{(, a\}$
 $First(Li) = \{, , \hat{A}\}$

Calculation of Follow: (2)

$Follow(S) = \{\$, , , \}$
 $Follow(L) = \{\}$
 $Follow(Li) = \{\}$

Predictive parsing table: (4)

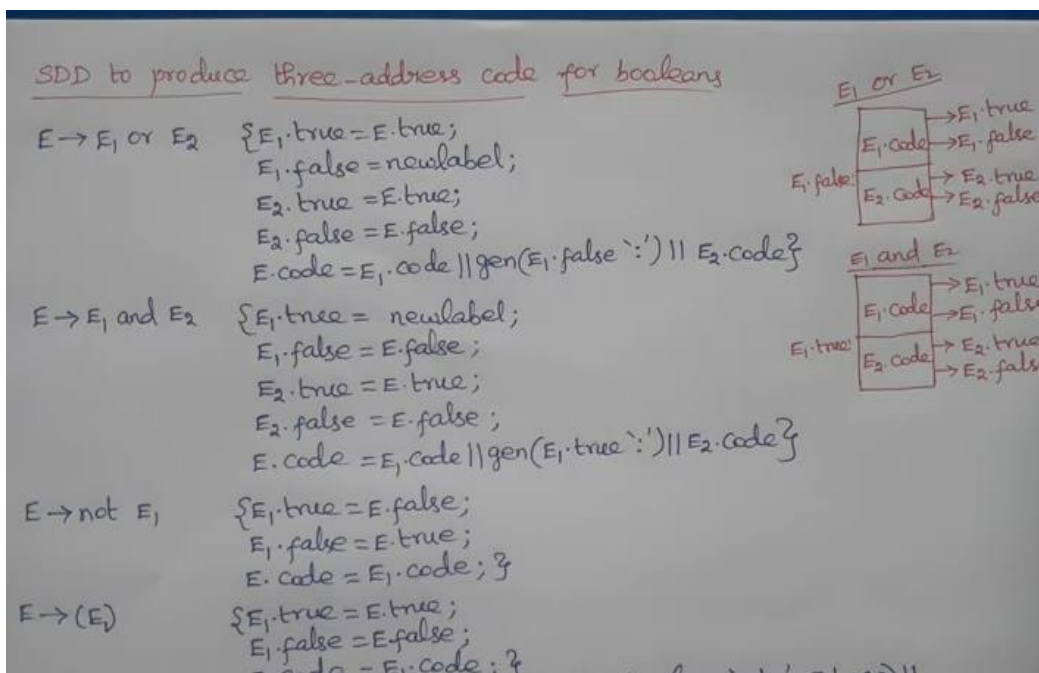
Non terminals	Input symbol				
	a	()	,	\$
S	S->a	S->(L)			
L	L->SLi	L->SLi			
Li			Li->A	Li->,SLi	

or

b. Illustrate the Storage allocation strategies in perspective of compiler with neat diagram. 13 C U

- The **different storage allocation strategies** are :
- **Static allocation** - lays out **storage** for all data objects at compile time. 04 N
- **Stack allocation** - manages the run-time **storage** as a stack. D
- **Heap allocation** - allocates and deallocates **storage** as needed at run time from a data area known as heap.

7. a. Demonstrate about the translation scheme to generate three address code for the Boolean Expression 13 C A
04 P
P

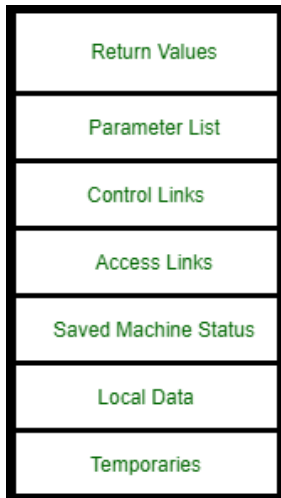


Or

- b. Explain the principle sources of code optimization in detail with example 13 C U
1. Common-Sub expression Elimination: O5 N
 2. Copy Propagation D
 3. Dead Code Elimination
 4. Constant folding
 5. Loop Optimizations

8. a. (i) What is an activation record? Explain how it is related with run time storage organization 10 C A

An activation record is a contiguous block of storage that manages information required by a single execution of a procedure 4



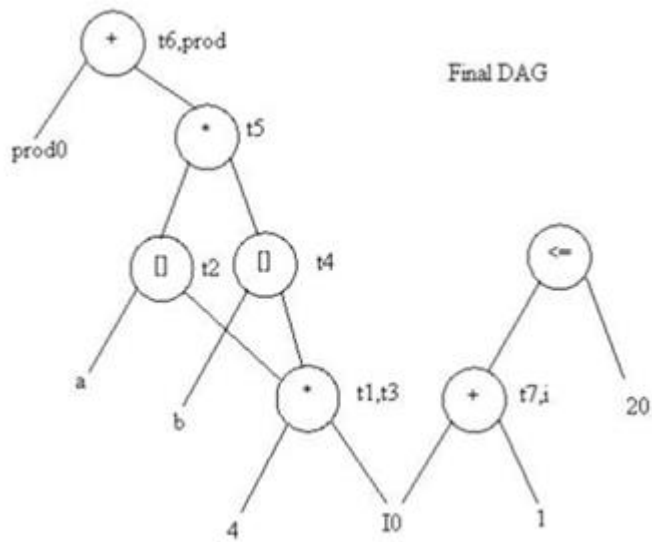
C
O4 U
N
D

- (ii) Summarize on back patching

Backpatching is basically **a process of fulfilling unspecified information**. This information is of labels. It basically uses the appropriate semantic actions during the process of code generation. It may indicate the address of the Label in goto statements while producing TACs for the given expressions

or

- b. Construct the DAG for the following Basic block & explain it. 14 C A
1. $t1 := 4 * i$ O5 P
 2. $t2 := a[t1]$ P
 3. $t3 := 4 * i$
 4. $t4 := b[t3]$
 5. $t5 := t2 * t4$
 6. $t6 := Prod + t5$
 7. $Prod := t6$
 8. $t7 := i + 1$
 9. $i := t7$
 10. if $i \leq 20$ goto (1).



**Und-Understanding Rem-Remembering App-Aplying
Ana-AnalyzeCre-Creating Eva-Evaluating**

Prepared By

Verified By

HoD