**Assignment(Module-3,4,5)**

**[submission deadline 25th March 2025]**

**CSE304: Compiler Construction**

**Q.1** Check whether the following grammar is LL(1), LR(0), SLR(1), CLR(1) and LALR(1)

or not? If grammar is CLR(1) or LALR(1), then construct a Parsing table for that parser.

|  |  |  |  |
| --- | --- | --- | --- |
| 𝑎) 𝑆→(𝐿)/𝑎 𝐿→𝐿, 𝑆/𝑆 | 𝑏) 𝑆→𝐿= 𝑅 𝑆→𝑅  𝐿→∗𝑅  𝐿→𝑖𝑑  𝑅→𝐿 | 𝑐) 𝑆→𝐴𝑎𝐴𝑏/𝐵𝑏𝐵𝑎 𝐴→∈  𝐵→∈ | 𝑑) 𝑆→𝐴𝑎/𝑏𝐴𝑐/𝐵𝑐/𝑏𝐵𝑎 𝐴→𝑑  𝐵→𝑑 |

**Q.2**: (i) Construct a CLR(1) Parsing table for the grammar given in part c) and parse the

string **ab**

**(ii)** Construct a LALR(1) Parsing table for the grammar given in part d) and parse the string

**bda**

**Q.3**: Consider the following grammar

𝑺→𝑨𝒂|𝒃𝑨𝒄|𝒅𝒄|𝒃𝒅𝒂

𝑨→𝒅

(a) Check whether the given grammar is SLR(1) or not?

(b) Construct a LALR(1) parsing table (using LR(1) items DFA).

(c) Parse the string 𝒃𝒅𝒄

**Q.4** What is symbol table? Discuss any method to implement a symbol table.

**Q.5** Translate the following expression into quadruples, triples and indirect triples.

- (a\*b) + (b+c) – (a+b+c+d)   
**Q.6** Differentiate between synthesized attributes and Inherited attributes in SDT.

**Q.7** Construct a syntax-directed translation (SDT) scheme for desktop calculator for the input string 3\*5+4. Draw the annotated parse tree also?

**Q.8** Let the attribute **val** gives the value of octal number generated by S in the following grammar:

𝑆→𝐿. 𝐿/𝐿

𝐿→𝐿𝑂/𝑂

𝑂→0/1/2/3/4/5/6/7

For example an input (15.25)8 = (13.3230)10 , that is, 𝒔. 𝒗𝒂𝒍= 𝟏𝟑. 𝟑𝟐𝟑𝟎 for the input value 15.25.

Construct a SDT scheme using only synthesized attributes to determine **s.val**

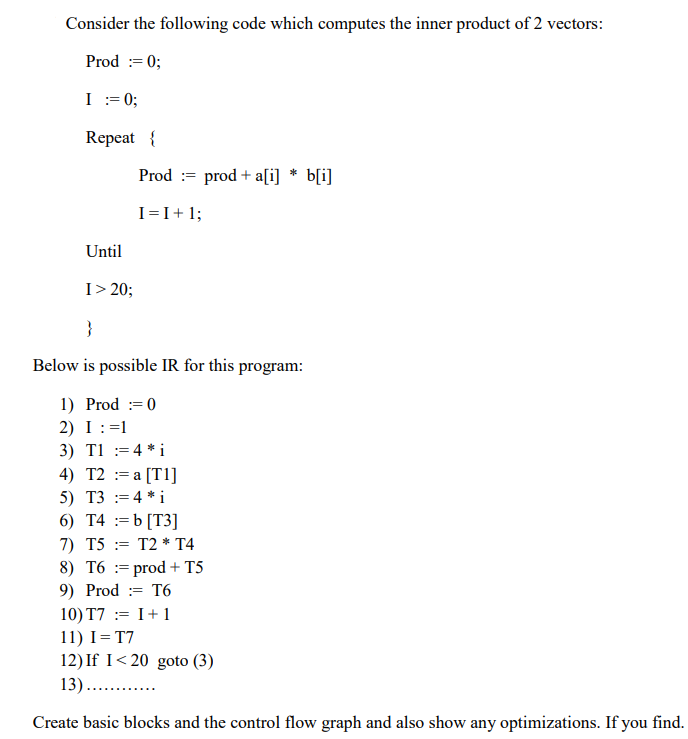
**Q.9** Consider the following code   
𝑺𝒖𝒎= 𝟎;   
𝑭𝒐𝒓(𝒊= 𝟏; 𝒊≤𝟐𝟎; 𝒊+ +)   
 𝑺𝒖𝒎= 𝒔𝒖𝒎+ 𝒂[𝒊] + 𝒃[𝒊];

(a) Generate three address code   
(b) Create basic blocks and control flow graph.

(c) Construct a DAG representation of 3 address code generated in part(a)

**Q.10** Write a 3-address code for the following program fragment, where a and b are arrays of size 20\*20 and there are 8 bytes per word (that is W=4):   
𝑠𝑢𝑚= 0;  
𝑖= 1;  
𝑗= 1;  
𝑑𝑜   
{   
𝑠𝑢𝑚= 𝑠𝑢𝑚+ 𝑎[𝑖, 𝑗] ∗𝑏[𝑗, 𝑖];  
𝑖= 𝑖+ 1;  
𝑗= 𝑗+ 1;   
}  
𝑤ℎ𝑖𝑙𝑒(𝑖≤10 *&&* 𝑗≤10);

**Q.11** Explain the following with an example:   
 (a) Code Optimization Phase   
 (b) Peephole optimization   
 (c) Reduction in strength   
 (d) Induction variable elimination   
**Q.12**



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*