TOPICWISE : COMPILER DESIGN-2 (GATE - 2019) - REPORTS

OVERALL ANALYSIS   COMPARISON REPORT   SOLUTION REPORT

ALL(17)   CORRECT(0)   INCORRECT(0)   SKIPPED(17)

Q. 1

Consider the regular expression-token mapping given below:

Regex	Token
ca*b	1
(a|b)*b	2
c*	3

Choose the correct output when lexical analyzer scans the following input: "aaabccabbb"

Have any Doubt ?

A   232   Correct Option

Solution :
(a)
aaab → 2
cc → 3
abbb → 2

B   132

C   231

D   123

📊 QUESTION ANALYTICS   +

Q. 2

Given the following attribute grammar:

$E \rightarrow TE'$ $E'.y = E.x$
                                        $E.x = E'.y$
$E_1' \rightarrow + TE_2'$ $E_2'.y = E_1'.y + T.x$
                                        $E_1'.x = E_2'.x$

Which of the following is true?

Have any Doubt ?

A   Both x and y are inherited attributes

B   Both x and y are synthesized attributes

C x is inherited, y is synthesized

D y is inherited, x is synthesized   Correct Option

Solution :
(d)
In case of y, the translation on RHS of production is defined in terms of translation of nonterminal on the left. So, y is inherited.
In case of x, translation of nonterminal on the left side of production is defined as function of translation of non-terminals on right hand side. So, x is synthesized
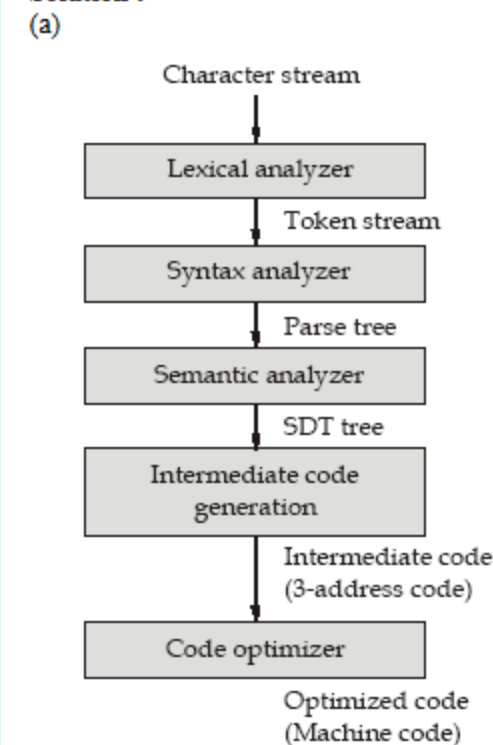
📊 QUESTION ANALYTICS   +

Q. 3

Choose the correct sequence of occurrence during compilation process

Have any Doubt ?

A   Character stream → parse tree → optimized code   Correct Option

Solution :
(a)

📊 QUESTION ANALYTICS **+**

Q. 4

Consider the following statements:
S_1 : Analysis phase of compiler includes code optimization stage.
S_2 : Synthesis phase of compiler is followed by analysis phase.

Have any Doubt ? 🔖

A S_1 is correct, S_2 is not

B S_2 is correct, S_1 is not

C S_1 and S_2 are correct

D S_1 and S_2 are incorrect   Correct Option

Solution :
(d)
Analysis phase {lexical analysis, syntax analysis, semantic analysis} is followed by synthesis phase {intermediate code generation, code optimizer, machine code generation}.

📊 QUESTION ANALYTICS **+**

Q. 5

Consider the following statements:
S_1 : In stack allocation, memory is allocated at compile time while in heap allocation, memory is allocated in runtime.
S_2 : Recursion is supported in stack allocation technique and not supported in static allocation technique.
Which of the following is true?

Have any Doubt ? 🔖

A S_1 is correct, S_2 is not

B S_2 is correct, S_1 is not   Correct Option

Solution :
(b)
In both stack and heap allocation, memory allocated at runtime.
Static allocation does not support recursion. However, in stack allocation, storage is organized as a stack and activation records are pushed and popped as activation begin and end respectively.

C Both are correct

D Both are incorrect

📊 QUESTION ANALYTICS **+**

Q. 6

The minimum number of variables required in 3 address code for given expression are _____.
$a \times b \times c + d - a + e \times f - g + h$
Order of precedence: $\times > + > -$; with $\times$ is left associative and $+, -$ are right associative.

Have any Doubt ? 🔖

● 3   Correct Option

Solution :
3

$$((((a \times b) \times c) + d) - ((a + (e \times f)) - ((g + h))))$$

$$t1 = a \times b$$
$$t1 = t1 \times c$$
$$t1 = t1 + d$$
$$t2 = e \times f$$
$$t2 = t2 + a$$
$$t3 = g + h$$
$$t2 = t2 - t3$$
$$t1 = t1 - t2$$

📊 QUESTION ANALYTICS **+**

Q. 7

The number of inadequate states in SLR(1) parsing table of following grammar are _____.
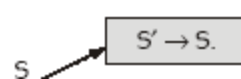$S \rightarrow V \mid aSc$
$V \rightarrow bV \mid d$

Have any Doubt ? 🔖

● 0   Correct Option

Solution :
0

SLR Parser:

$S' \rightarrow S.$

S

Zero inadequate states since no SR conflict or RR conflict is present.

Q. 8

Consider a stack S given below with elements 1, 2, 3, along with the usual operations of push( ), pop( ) and top( ) as shown.

Stack 'S'	SDT:	
1	$A \rightarrow B \times C$	{S.push(5)}
3	$B \rightarrow C/D$	{S.pop( )}
2	$C \rightarrow A$	{S.top( )}
	$B, C, D \rightarrow id.num$	{S.top( )}

Input string: $1/2 \times 3 \times 4$

The number of elements will remain in the stack after the parsing of above input string according to the above SDT rules _____.

[**push(x):** insert x into stack, pop( ): remove top most element of stack, top( ): print top most element of stack without removing]

FAQ | Have any Doubt ?

⬤ 4   Correct Option

Solution :
4

Q. 9

Given the 3-address code for a basic block:

Num	Instruction	Meaning
1	Ld a, T_1	$T_1 \leftarrow a$
2	Ld b, T_2	$T_2 \leftarrow b$
3	Ld c, T_3	$T_3 \leftarrow c$
4	Ld d, T_4	$T_4 \leftarrow d$
5	Add T_1, T_2, T_5	$T_5 \leftarrow T_1 + T_2$
6	Add T_5, T_3, T_6	$T_6 \leftarrow T_3 + T_5$
7	Add T_6, T_4, T_7	$T_7 \leftarrow T_6 + T_4$
8	ST T_7, a	$a \leftarrow T_7$

The number of registers that are needed to allocate this basic block with no spills are _____.

Have any Doubt ?

⬤ 4   Correct Option

Solution :
4

$$R_1 \leftarrow a$$
$$R_2 \leftarrow b$$
$$R_3 \leftarrow c$$
$$R_4 \leftarrow d$$
$$R_2 \leftarrow R_1 + R_2$$
$$R_3 \leftarrow R_3 + R_2$$
$$R_4 \leftarrow R_3 + R_4$$
$$a \leftarrow R_4$$

So, 4 registers are needed.

Q. 10

Consider the following SDT:

$A \rightarrow A \times B$	{1}
$A \rightarrow E$	{A.val = E.val}
$B \rightarrow C + D$	{2}
$C \rightarrow C - D$	{C.val = C.val + D.val}

$D \rightarrow D - E$ {D.val = D.val + E.val}

$E \rightarrow E/F$ {3}

$F \rightarrow G$ {F.val = G.val}

A, B, C, D, E, F, G $\rightarrow$ id   {A.val = id.num

                         B.val = id.num

                         C.val = id.num

                         D.val = id.num

                         E.val = id.num

                         F.val = id.num

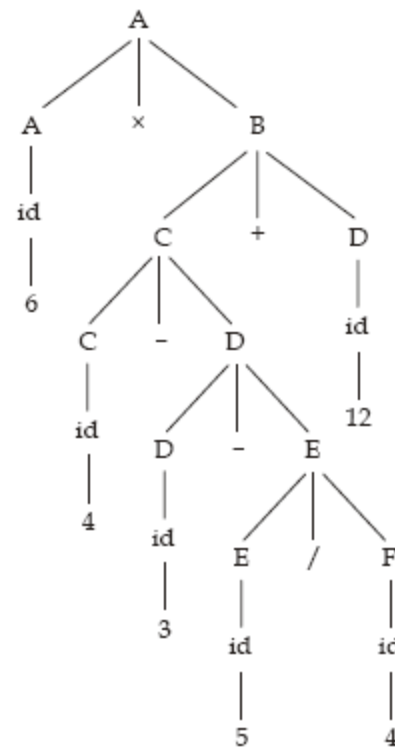   G.val = id.num}

Fill 1,2,3 so that the string $6 \times 4 - 3 - 5/4 + 12$ outputs 32

A
A.val = A.val × B.val
B.val = C.val × D.val
E.val = E.val + F.val

B
A.val = A.val × B.val
B.val = C.val – D.val
E.val = E.val + F.val

C   Correct Option
A.val = A.val/B.val
B.val = C.val × D.val
E.val = E.val + F.val

Solution :
(c)

D
A.val = A.val/B.val
B.val = C.val/D.val
E.val = E.val × F.val

📊 QUESTION ANALYTICS   +

Q. 11

Consider the following statements:

S_1 : During a program execution, stack is used for dynamic memory allocation and heap is used for static memory allocation.

S_2 : During a program execution, heap is stored in main memory and stack is present in secondary memory.

S_3 : During a program execution, access to heap memory variables is slower as compared to accessing variables allocated on stack.

S_4 : During a program execution, in a multithreaded situation, each thread has its own stack and share a common heap memory.

Which of the following are true?

A S_1 and S_3 only

B S_1 and S_4 only

C S_3 and S_4 only   Correct Option

Solution :
(c)

S_1 : Stack and heap is used for dynamic memory allocation.

S_2 : Heap and stack both are present in main memory.

S_3 : Access to heap memory variables is slower as compared to accessing variables allocated on stack. Because to access a heap memory variable we need access pointer variable first.

S_4 : In a multithreaded situation, each thread has its own stack and share a common heap memory.

D S_2 and S_3 only

📊 QUESTION ANALYTICS   +

Q. 12

What would be the output of following program for static and dynamic scoping respectively?

```
int x;
int main( ) {x = 23; f( ); g( );}
void f( ) {int x = 22; int y = 99; h( );}
void g( ) {int x = 45; int z = 23; h( );}
void h( ) {printf("%d\n", x);}
```

A   23, 23; 23, 23

B   22, 45; 22, 45

Solution :
(c)

Static scoping means that x refers to the x declared innermost scope of declaration. Since 'h' is declared inside the global scope, the innermost x is the one in the global scope (it has no access to the x's in 'f' and 'g', since it was not declared inside them), so the program prints 23 twice. Dynamic scoping means that x refers to the x declared in the most recent frame of the call stack. 'h' will use the x from either 'f' or 'g', whichever one that called it so the program would print 22 and 45.

D 23, 23; 99, 45

Q. 13

Consider the following control flow graph given below:
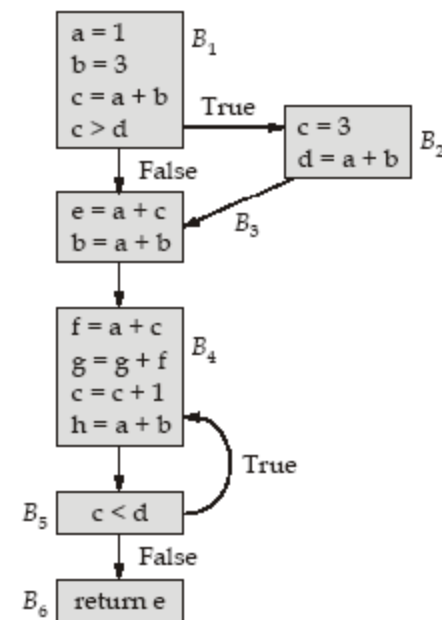
Which of the following is true for the above control flow graph?

Have any Doubt ? 🔖

A Block B_1 and B_3 can be combined together to form a basic block.

B Block B_2 and B_3 can be combined together to form a basic block.

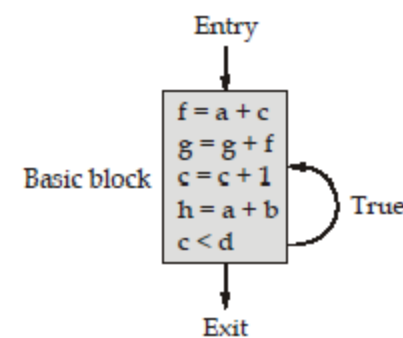**C** Block B_5 and B_6 can be combined together to form a basic block.

Solution :
(d)

Block B_4 and B_5 can be combined together to form a basic block since block B_5 is just a condition which is repeated till c < d true and B_4 must be executed.

So, B_4 and B_5 can be merged.

Q. 14

Consider the following grammar G:
S → Aa|bAc|dc|bda
A → d
Which of the following is true?

Have any Doubt ? 🔖

A G is SLR(1)

Solution :
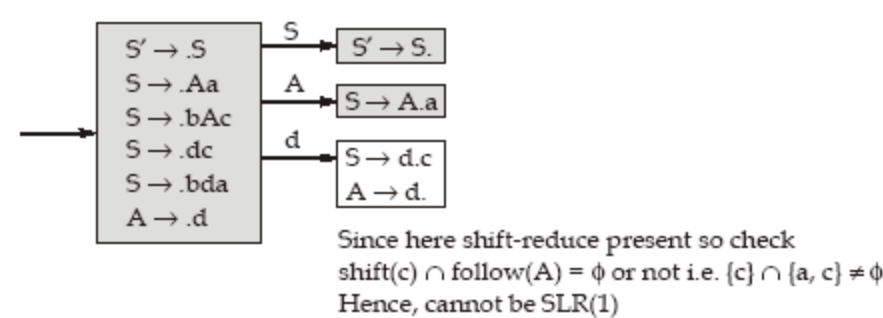(b)

Check for SLR(1):

Since here shift-reduce present so check
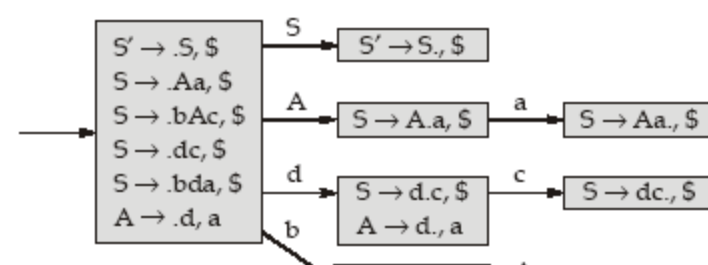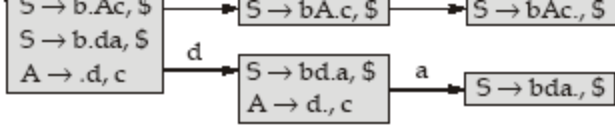shift(c) ∩ follow(A) = φ or not i.e. {c} ∩ {a, c} ≠ φ
Hence, cannot be SLR(1)

No need to design full DFA, check on each state.

Check for LALR:

Since no state present, which only differs is look ahead symbols, hence grammar has LALR(1) and LR(1) DFA with same state. So grammar is LR(1), LALR(1) but not SLR(1).

C G is LR(1) but not LALR(1)

D G is SLR(1), LALR(1) and LR(1)

Q. 15

Consider the CFG G = {S, {S, U, V, W}, {a, b, c, d}, P} with LL(1) table given below:
S → UVW
U → (S)|aSb|d
V → aV|∈
W → cW|∈
LL(1) Table:

M[T, t]	a	b	c	d	(	)	$
S	S → UVW			S → UVW			
U	U → aSb			U → d	U → (S)		
V	V → aV						V → ∈
W			W → cW				W → ∈

The number of entries missing in LL(1) table for given grammar _____.

Have any Doubt ? 🔖

● 6   Correct Option

Solution :
6

First(S) = {(, a, d}   Follow(S) = {), b, $}
First(U) = {(, a, d}   Follow(U) = {a, c,), b, $}
First(V) = {a, ∈}   Follow(V) = {c,), b, $}
First(W) = {c, ∈}   Follow(W) = {), b, $}

LL(1) Table:

M[T, t]	a	b	c	d	(	)	$
S	S → UVW			S → UVW	S → UVW		
U	U → aSb			U → d	U → (S)		
V	V → aV	V → ∈	V → ∈			V → ∈	V → ∈
W		W → ∈	W → cW			W → ∈	W → ∈

Six entries are missing.

Q. 16

Consider the following grammar:
S → Aa|bAc|dc|bda
A → d
If given grammar is LALR(1), then the minimum number of states needed for DFA construction is _____.

Have any Doubt ? 🔖

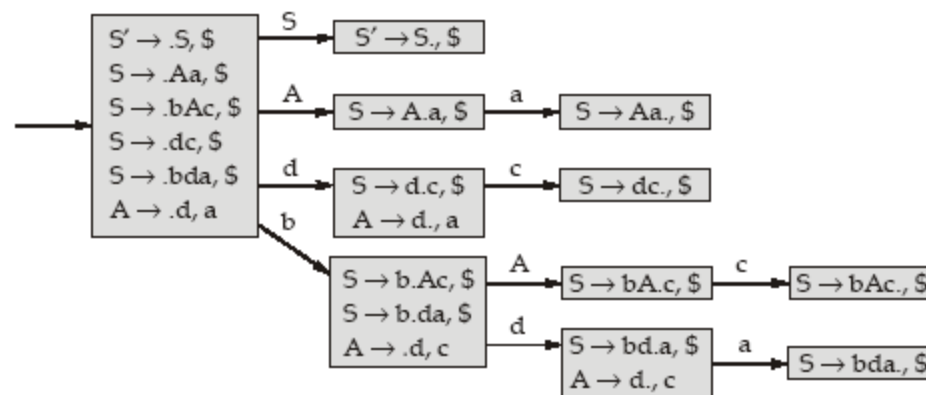● 11   Correct Option

Solution :
11

LR(1):

Since their is not state which only differs in look ahead symbol, hence given LR(1) DFA is LALR(1) DFA. Hence 11 states are needed.

Q. 17

Consider the grammar given below:
S → aXab|Y
X → bYa|∈
Y → Sc
The minimum number of states required to construct LR(1) DFA are _____.

Have any Doubt ? 🔖

● 17   Correct Option

Solution :
17

S_4
$S \rightarrow aX.ab, \$, c$

S_5
$S \rightarrow aXa.b, \$, c$

S_6
$S \rightarrow aXab., \$, c$

S_0
$S' \rightarrow .S, \$$
$S \rightarrow .aXab, \$, c$
$S \rightarrow .Y, \$, c$
$Y \rightarrow .Sc, \$, c$

S_3
$S \rightarrow a.Xab, \$, c$
$X \rightarrow .bYa, a$
$X \rightarrow .\epsilon, a$

S_7
$X \rightarrow b.Ya.a$
$Y \rightarrow .Sc, a, c$
$S \rightarrow .aXab, c$
$S \rightarrow .Y, c$

S_8
$Y \rightarrow S.c, a, c$

S_9
$Y \rightarrow Sc., a, c$

S_{10}
$S \rightarrow Y., c$
$S \rightarrow bY.a, a$

S_{12}
$S \rightarrow bYa., a$

S_{16}
$S \rightarrow Y., \$, c$

S_{12}
$S \rightarrow a.Xab, c$
$X \rightarrow .bYa, a$
$X \rightarrow .\epsilon, a$

S_{13}
$S \rightarrow aX.ab, c$

S_{14}
$S \rightarrow aXa.b, c$

S_{15}
$S \rightarrow aXab., c$
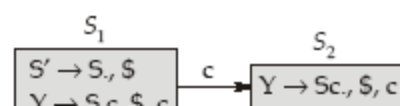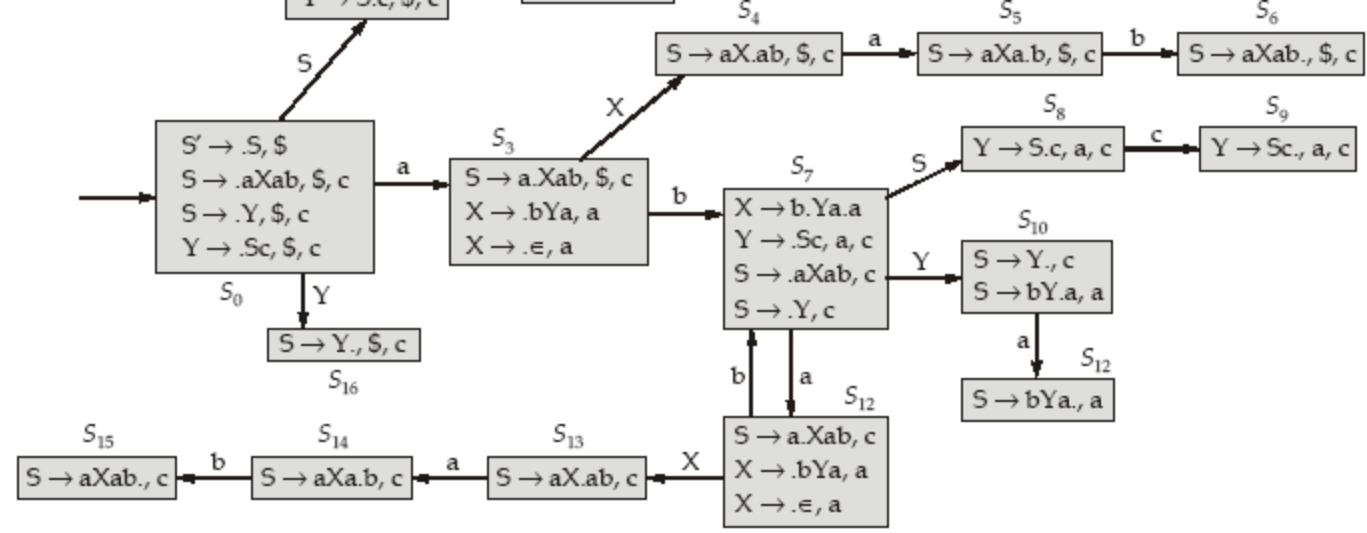
So total 17 states are needed for LR(1) DFA.

QUESTION ANALYTICS