Nitish Kumar Gupta
Course: GATE
Computer Science Engineering(CS)

- HOME
- MY TEST
- BOOKMARKS
- MY PROFILE
- REPORTS
- BUY PACKAGE
- ASK AN EXPERT
- OFFER
- EXCLUSIVE OFFER FOR OTS STUDENTS ONLY ON BOOK PACKAGES

SINGLE SUBJECT : COMPILER DESIGN (GATE - 2019) - REPORTS

OVERALL ANALYSIS   COMPARISON REPORT   SOLUTION REPORT

ALL(33)   CORRECT(15)   INCORRECT(10)   SKIPPED(8)

Q. 1

Which of the following statements is true?

Have any Doubt ?

A Every regular grammar is LL(1).

B Every regular language is LL(1).

C S → abc|ad, this grammar is not LL(1) but it is LL(2).

D Both (b) and (c) Correct Option

Solution :
(d)

- Every regular grammar need not to be LL(1) because regular grammar may contain left recursion, left factoring, ambiguity.
- Every regular language is LL(1), is true. Because at least there exist a regular grammar which is LL(1).
- S → abc | ad
  IN LL(1), First (S) = {a}
  Both production will enter in the same entry because left factoring in LL(1).
  Hence not LL(1).
  In LL(2), both production will enter into differ entry because no left factoring is present.
  Hence it is LL(2).

QUESTION ANALYTICS +

Q. 2

The resolution of externally defined symbols and functions associated with the program is performed by

Have any Doubt ?

A Loader

B Linker Your answer is **Correct**

Solution :
(b)

C Compiler

D Assembler

QUESTION ANALYTICS +

Q. 3

Consider the following grammar G. Where X, Y and Z are non-terminals and remaining are terminals used:

X → Y | Z
Y→ a | b
Z → c | b

Which of the following statements are correct?
S_1 : SLR(1) can parse all the strings that are generated by grammar G.
S_2 : LR(1) can parse all the strings that are generated by grammar G.

Have any Doubt ?

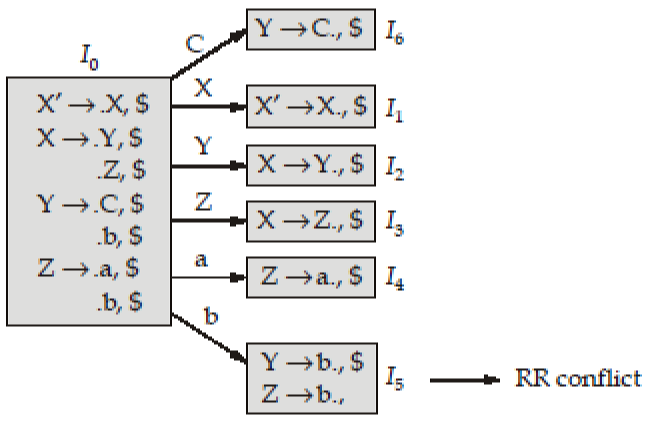
A Only S_1

B Only S_2

C Both S_1 and S_2

D None of these Correct Option

Solution :
(d)



In the above LR(1) parsers there is RR conflict in I_5 state.

It is not LR(1), it means it will also not be SLR(1).

QUESTION ANALYTICS +

Q. 4

Assume input is scanned from left to right of the given expression during parsing. Which of the following derivations for a top down parser uses while parsing.

Have any Doubt ?

A Right most derivation

B Left most derivation — Your answer is **Correct**

Solution :
(b)
Left most derivation is used.

C Reverse of right most derivation

D Any of the above

QUESTION ANALYTICS +

Q. 5

Consider the following SDT:

$L \rightarrow E$
$E \rightarrow \{\text{print}('+')\}\ E_1 + T/T$
$T \rightarrow \{\text{print}('*')\}\ T_1 * F/F$
$F \rightarrow (E)$
$F \rightarrow \text{digit}\ \{\text{print (digit.val)}\}$

Which of the following options is true?

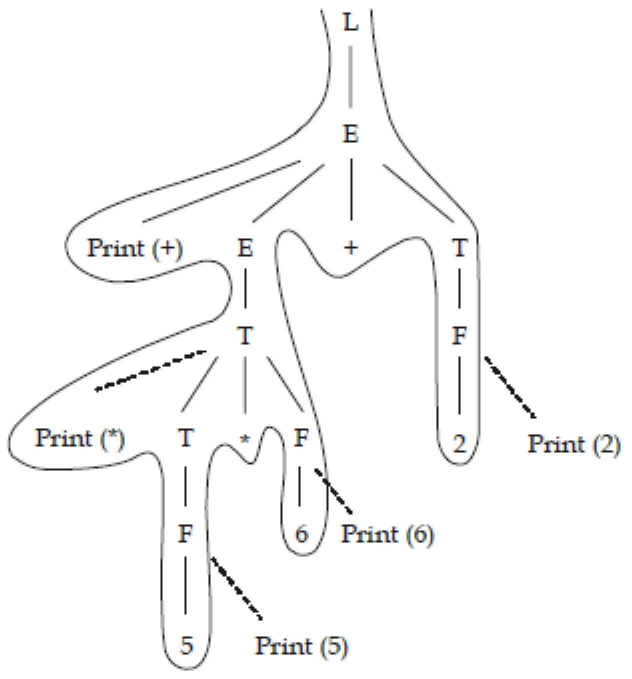
Have any Doubt ?

A Evaluating prefix expression

B Infix to prefix — Your answer is **Correct**

Solution :
(b)

Let us take an expression 5 * 6 + 2 and parse it.



Output: + * 5 6 2
It converts an infix expression to prefix expression.

C Infix to postfix

D Evaluating postfix expression

QUESTION ANALYTICS +

Q. 6

Consider the following CFG:

$X \rightarrow (Y) \mid Z$
$Y \rightarrow XY'$
$Y' \rightarrow \in \mid * XY$

Where X, Y, Y' are non-terminals and *, Z are terminals. What is the FIRST and FOLLOW of Y?

Have any Doubt ?

A FIRST = {), ∈}
FOLLOW = {)}

B FIRST = {(, ∈}
FOLLOW = {), ∈}

C FIRST = {), *}
FOLLOW = {), ∈}

D FIRST = {(, Z}
FOLLOW = {)} — Your answer is **Correct**

Solution :

(d)

$$\begin{aligned} \text{FIRST (Y)} &= \{(, Z\} \\ \text{FOLLOW (Y)} &= \{)\} \cup \text{FOLLOW (Y')} \\ &= \{)\} \cup \{)\} \\ &= \{)\} \end{aligned}$$

QUESTION ANALYTICS +

Q. 7

Which of the following sufficient to convert an arbitrary CFG to an LL(1) grammar?

Have any Doubt ?

A Only left recursion

B Only ambiguity

C Both left recursion and left factoring

D None of the above Your answer is **Correct**

Solution :
(d)
A grammar is LL(1) if it does not contain
(a) Left recursion
(b) Ambiguity
(c) Lcft factoring
Hence to convert an arbitrary CFG to LL(1) grammar all three should be eliminated.

QUESTION ANALYTICS +

Q. 8

Consider the grammar defined by the following production rules with 2 operators + and –

$X \rightarrow X + Y \mid Z$
$Y \rightarrow Y - Z \mid Z$
$Z \rightarrow id$

Have any Doubt ?

A '+' is left associative while '–' is right associative.

B '–' is right associative and '+' has left associative.

C Both have not fixed associativity.

D Both are left associative. Your answer is **Correct**

Solution :
(d)
Both X and Y have left recursion.
So +, – both are left associative.

QUESTION ANALYTICS +

Q. 9

Which of the following grammar is operator grammar with variables E, T, A, B are non-terminals and a, b, $\in$, id are terminals

Have any Doubt ?

A
$E \rightarrow E + E$
$E \rightarrow E * E$
$E \rightarrow id$
$E \rightarrow \in$

B
$E \rightarrow EA$
$A \rightarrow b$
$A \rightarrow a$
$E \rightarrow id$

C Correct Option
$E \rightarrow E + E$
$E \rightarrow E * E$
$E \rightarrow id$
$E \rightarrow a$

Solution :
(c)
A grammar G is said to be operator grammar if
1. It does not contain null production.
2. It does not contain 2 adjacent variable on right hand side.
So, clearly option (a), (b) and (d) are incorrect.

D
$E \rightarrow T + B$
$B \rightarrow E + A$
$A \rightarrow a$
$B \rightarrow b$
$T \rightarrow \in$

QUESTION ANALYTICS +

Q. 10

In a bottom-up evaluation of a syntax directed definition, inherited attribute can

Have any Doubt ?

A Always be evaluated.

B Be evaluated only if definition in L-attributed. Your answer is **Correct**

Solution :
(b)
A SDT is called S attributed if it has only synthesized attributed. L-attributed definitions contain both synthesized and inherited attributes.

C Be evaluated only if the definition has synthesized attributes.

D Never be evaluated.

QUESTION ANALYTICS +

Q. 11

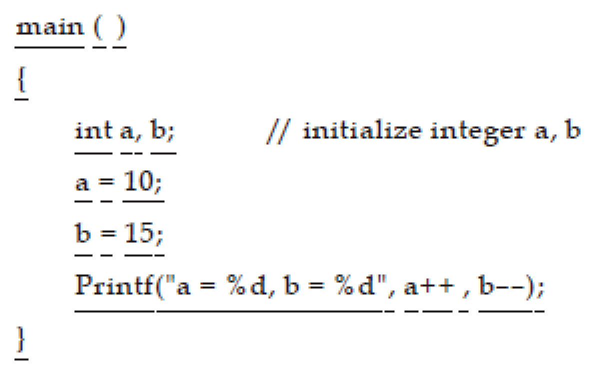
```
int main ( )
 {
     int a, b; // initialize integer a, b
     a = 10;
     b = 15;
     printf("a = %d, b = %d", a++, b--);
 }
```

The number of tokens in the above C program is ________.

Have any Doubt ?

● 29 Correct Option

Solution :
29



```
main ( )
{
     int a, b;        // initialize integer a, b
     a = 10;
     b = 15;
     Printf("a = %d, b = %d", a++ , b--);
}
```

Total 29 tokens are available in the above C program.

● Your Answer is 30

QUESTION ANALYTICS +

Q. 12

Let G be any grammar with the following productions:

$X \rightarrow X + Y | Y$
$Y \rightarrow Y * Z | Z$
$Z \rightarrow (X)$
$Z \rightarrow id$

If LR(1) parser is used to parse the above grammar, then total how many look-a-heads are present for the item X → .Y and Z → .id in the initial state ________.

Have any Doubt ?

● 3 Correct Option

Solution :
3

$X' \rightarrow .X, \{\$\}$
$X \rightarrow .X + Y, \{\$, +\}$
$X \rightarrow .Y, \{\$\}$ ...(i)
$Y \rightarrow .Y * Z, \{\$, *\}$
$Y \rightarrow .Z, \{\$, *\}$
$Z \rightarrow .(X)\ \{\$, *\}$
$Z \rightarrow .id, \{\$, *\}$ ...(ii)

So, total (3) look-a-heads are there.

QUESTION ANALYTICS +

Q. 13

Consider the following grammar G:

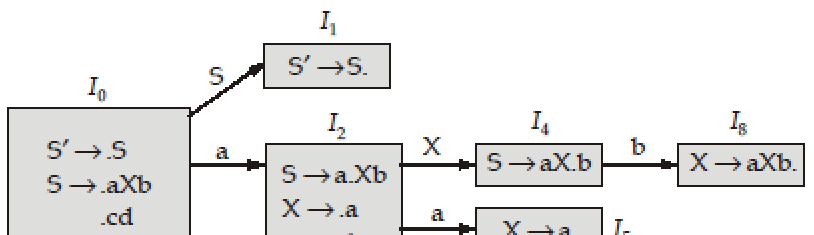
$S \rightarrow aXb | cd$
$X \rightarrow a | b$

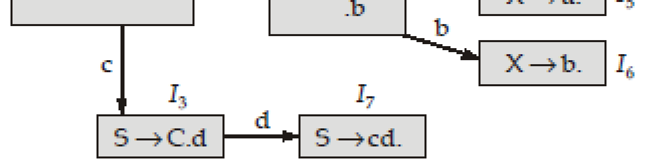
The number of DFA states in LR(0) construction is ________.

Have any Doubt ?

● 9 Correct Option

Solution :
9





Total 9 states are there.

● Your Answer is 8

QUESTION ANALYTICS +

Q. 14

Consider the following statements:
S_1 : Static allocation binding do not change at runtime.
S_2 : Heap allocation allocates and de-allocates memory at runtime.
S_3 : Non-local data such as global constants are known during compile time and placed in static area.
The number of correct statements are ________.

Have any Doubt ?

● 3 Your answer is **Correct**3

Solution :
3
All the above statements are correct.

QUESTION ANALYTICS +

Q. 15

Consider the following statements:
S_1 : Three address code is a linearized representation of syntax tree.
S_2 : The syntax tree not depicts the natural hierarchical structure of source program.
How many number of statements are correct ________.

Have any Doubt ?

● 1 Correct Option

Solution :
1

- Three address code is linear representation of syntax tree in which each statement can have at most 3-address. So, S_1 is true.
- The syntax tree depicts the natural hierarchical representation of source program. So, S_2 is false.

Hence only one statement is correct.

● Your Answer is 2

QUESTION ANALYTICS +

Q. 16

Let G be the following grammar:
$S \rightarrow XYfg$
$X \rightarrow XSm \mid \epsilon$
$Y \rightarrow XhS \mid \epsilon$
Variables S, X and Y are non-terminals and f, g, h, m are terminals. The number of productions will be in G after elimination of all the NULL productions are ________.

Have any Doubt ?

● 8 Correct Option

Solution :
8
Remove $X \rightarrow \epsilon$, then we get

$S \rightarrow XYfg \mid Yfg$
$X \rightarrow XSm \mid Sm$
$Y \rightarrow XhS \mid \epsilon \mid hS$

Now remove $Y \rightarrow \epsilon$, then we get

$S \rightarrow XYfg \mid Yfg \mid Xfg \mid fg$
$X \rightarrow XSm \mid Sm$
$Y \rightarrow XhS \mid hS$

So, total 8 productions will be there after eliminating null productions.

● Your Answer is 7

QUESTION ANALYTICS +

Q. 17

Consider the following grammar:
$S \rightarrow XX$
$X \rightarrow b$
$X \rightarrow aX$
Where S and X are non-terminals and a, b are terminals. Which of the following can be the viable prefixes?

Have any Doubt ?

A baab

B aab Correct Option

Solution :

(b)

Viable prefixes: Combination of non-terminal and terminal which can be in the stack during parsing is called viable prefix.
A handle X→ b is available. So whenever terminal b is encounter it is popped out before inserting new element into the stack.
Option (a) is wrong because before popping 'b' element 'aab' is inserted.
Option (b) is correct after 'b' no element has inserted.
So, option (c) and (d) also incorrect.

C aaabab

D bbbaX

QUESTION ANALYTICS +

Q. 18

Consider the following C program:

Program 1:
```
int main ( )
{
    int a = 3;
    @ _ /* hello world */
    return 0;
}
```

Program 2:
```
int main ( )
{
    int x = 10;
    if (x > 20
    x = 20;
}
```

Program 3:
```
int main ( )
{
    int x = 20;
    /* hello world
    if (x > 20)
    x = 10;
}
```

Program 4:
```
int main ( )
{
    int a = 09;
    /* hello world */
}
```

Assume programs are compiled same as give above. How many programs results in lexical error?

Have any Doubt ?

A 1

B 2 Your answer is **Wrong**

C 3 Correct Option

Solution :
(c)

Any symbols $, ', @ like that gives lexical error in C if put any where outside of string and the comment lines. They can not pass the lexer because lexer can not recognize them as a valid token.

- Program 1 gives lexical error because "@_" written outside comment.
- Program 2 having no lexical error but producing semantic error by syntax analyzer.
- Program 3 gives lexical error closing comment lines */ is missing.
- Program 4 gives lexical error because integer has been assigned an invalid octal number "09".

D 4

QUESTION ANALYTICS +

Q. 19

Which of the following statements holds true?

Have any Doubt ?

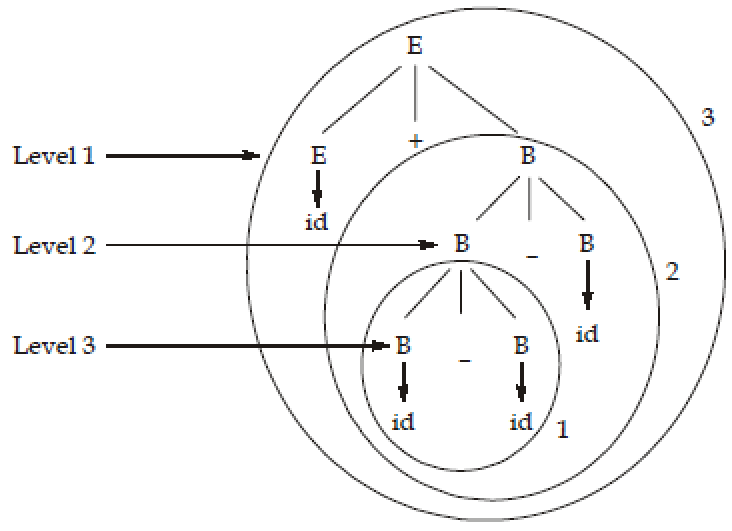
A The precedence of operator always depends on the level at which they are defined. Your answer is **Correct**

Solution :
(a)

Let us take a grammar

$$E \rightarrow E + B \mid id$$
$$B \rightarrow B - B \mid id$$

Syntax tree:



- Statement (a) correct. In this grammar '–' operator is computed first because they are at higher level then '+' operator.
- Statement (b) is incorrect because precedence will be different at different level.
- Statement (c) is also wrong because, when the '–' operator encountered as show in circle (1) and (2), operator inside circle (2) will be computed first because they are at higher level than circle (1). Here left associativity has computed first.

B Precedence of operator '+' and '–' will be same, when they are at level 1 and level 2 respectively of syntax tree.

C Associativity does not play role in higher precedence

C Associativity does not play role in higher precedence.

D Both (a) and (b)

QUESTION ANALYTICS +

Q. 20

Assume X, Y, Z represents the number of unfilled entries in the LR(0), SLR(1) and LR(1) table respectively. A common grammar G have been successfully parsed by all the three mentioned parsers. What is the relation over X, Y and Z?

Have any Doubt ?

A $X \leq Y \leq Z$ Your answer is **Correct**

Solution :
(a)
In general, number of unfilled entries in LR(0) ≤ number of unfilled entries in SLR(1) ≤ number of unfilled entries in LR(1).
So option (a) holds true.

B $X = Y = Z$

C $X \geq Y \leq Z$

D $X < Y < Z$

QUESTION ANALYTICS +

Q. 21

Consider the following code segment:

$X = A/b$
$Y = C + D$
$Y = Y - X$
$X = D + Y$
$Z = F + Y$
$Z = Z + A$

The minimum number of total variables required to convert the above code segment to static single assignment form is

Have any Doubt ?

A 9

B 10

C 11 Your answer is **Correct**

Solution :
(c)
In static single assignment, each assignment to variable should be specified with distinct names.

$$\begin{aligned} t_1 &= t_2 / t_3 \\ t_4 &= t_5 + t_6 \\ t_7 &= t_4 - t_1 \\ t_8 &= t_6 + t_7 \\ t_9 &= t_{10} + t_7 \\ t_{11} &= t_9 + t_2 \end{aligned}$$

So total 11 variables.

D 12

QUESTION ANALYTICS +

Q. 22

Consider the following statements:
(i) Type checking is done during all the phases especially in syntax analysis phase.
(ii) Target code generation phase is machine independent code generation.
(iii) SDT with only synthesized attribute, always have a order of evaluation.
(iv) Symbol table is constructed during analysis part of compiler
Which of the above is correct?

Have any Doubt ?

A (i) and (ii) only

B (iii) and (iv) only Your answer is **Correct**

Solution :
(b)
• Type checking is done during semantic analysis phase.
• Target code generation is machine specific.
• SDT with only synthesized attribute does not have any cyclic dependency. So always have order of evaluation.
• Symbol table is constructed during the lexical, syntax and semantic analysis phase.

C (ii) and (iv)

D (i) and (ii) only

QUESTION ANALYTICS +

Q. 23

Consider the following two sets of LR(1) items of LR(1) grammar:

A → d., c	A → d., b | f
B → d., b	B → d., d | $
X → .fx, g	X → .fx, $
Y → .g, a	Y → .g, $

Which of the following statement related to merging of the two sets in the corresponding parser is true?

I. Can be merged but will result in S-R conflicts.

II. Can be merged but will result in R-R conflicts.

III. Can not be merged since go to of f is leading to SR conflict with A → d.,

IV. Can not be merged since look ahead are different.

Have any Doubt ?

A I only

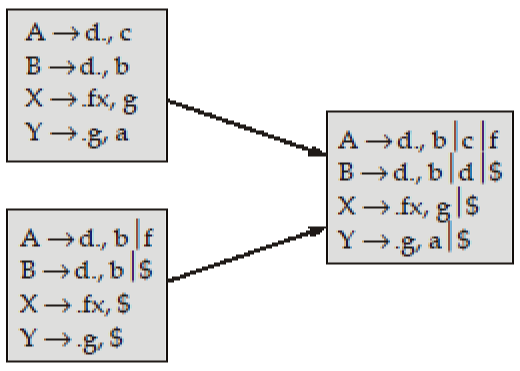
B III and IV only

C I, II and III only

Correct Option

Solution :

(c)



I. Can be merged but will results in S-R conflict is true because of productions.

$$\boxed{\begin{array}{l} A \to d., b \mid c \mid \textcircled{f} \\ X \to .\textcircled{f}x, g \mid \$ \end{array}}$$

II. Can be merged but will results in R-R conflict is true because of productions.

$$\boxed{\begin{array}{l} A \to d., \boxed{b} \mid c \mid f \\ B \to d., \boxed{b} \mid d \mid \$ \end{array}}$$

2-different production will be in the same entry.

III. Goto of f leading to SR conflict is true.

IV. Cannot be merged since look-a-heads are different false, because merging does not depend on look-a-head.

D I and II only

QUESTION ANALYTICS +

Q. 24

Consider the given SDT having left recursion. Which of the following options is correct after elimination of left recursion from the SDT.

X → XY $\{X.x = f(X.x, Y.y)\}$

X → Z $\{X.x = g(Z.z)\}$

Have any Doubt ?

A X → Z $\{X.x = g(Z.z)\}$ X′

X′ → Y $\{X'.x = f(X.x, Y.y)\}$ X′

B X → Z $\{X.x = g(Z.z)\}$ R

R → Y $\{R.x = f(X.x, Y.y)\}$ R

R → ∈

Correct Option

Solution :

(b)

As we know,

A → Aα | B

After eliminating LR

A → BR

R → αR | ∈

In the given case

$$\alpha = Y\ \{X.x = f(X.x, Y.y)\}$$

Final production after eliminating will be

X → Z $\{X.x = g(Z.z)\}$ R

R → Y $\{R.x = f(X.x, Y.y)\}$ R

R → ∈

So, option (b) is correct.

C X → Z $\{X.x = g(Z.z)\}$ R

R → $\{R.x = f(X.x, Y.y)\}$ Y

R → ∈

D None of the above

QUESTION ANALYTICS +

Q. 25

Consider the following grammar:

S → XaXb | YbYa

X → ∈

Y → ∈

The above grammar is:

Have any Doubt ?

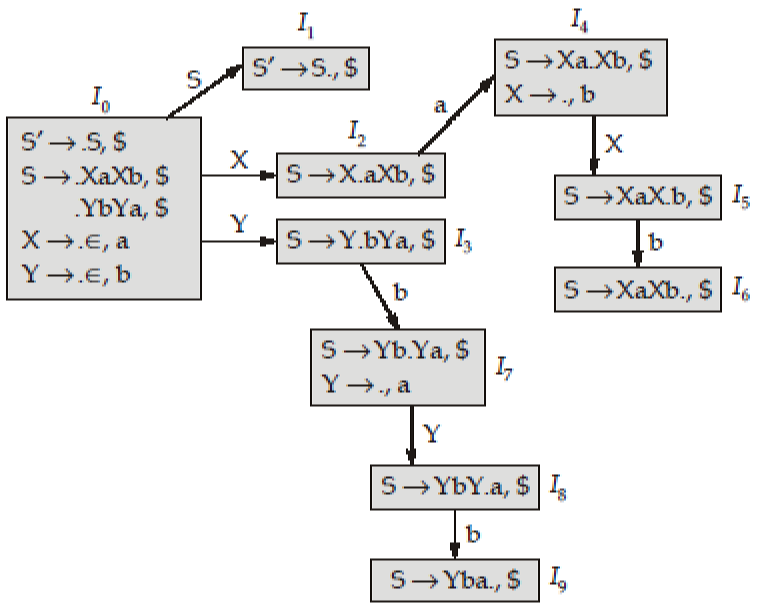
A LL(1) but not LR(1)

B LL(1) and LR(1) Correct Option

Solution :
(b)

$$\begin{aligned}\text{First (S)} &= \{a, b\}\\ \text{First (X)} &= \{\in\};\ \text{FOLLOW (X)} = \{a, b\}\\ \text{First (Y)} &= \{\in\};\ \text{FOLLOW (Y)} = \{a, b\}\end{aligned}$$

→ Grammar is LL(1).



Grammar is not SLR(1) because of conflict at the state I_0. Hence it will not be LR(0) also.
The above grammar is LR(1).
Hence option (b) is correct.

C LR(0) but not LL(1) Your answer is **Wrong**

D SLR(1) and LR(1)

QUESTION ANALYTICS +

Q. 26

Consider the following statements and choose the option which are not correct about parsers.

Have any Doubt ?

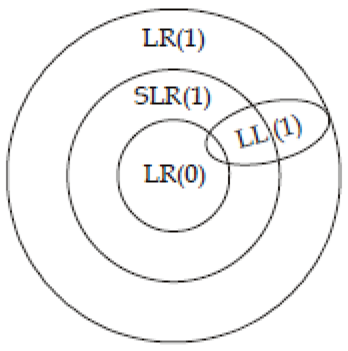
A Operator precedence parsers can parsed some ambiguous grammars.

B LR(1) grammars can not be LL(1) but can be LR(0).

C CLR(1) don't have SR conflict. Your answer is **Wrong**

D Both (b) and (c) Correct Option

Solution :
(d)
Some LR(1) grammars can also be LL(1).
So option (b) is incorrect.



Option (c) is wrong because in CLR(1) parsers SR conflicts can occur.
So, option (d) is correct.

QUESTION ANALYTICS +

Q. 27

Consider the following statements with respect to LR parsers:
S_1 : To occur a conflict minimum 2 productions is required.
S_2 : At least one of the production should to reduce.
Which of the following is correct?

Have any Doubt ?

A S_1 and S_2 sufficient for conflict

B S_1 or S_2 alone sufficient for conflict

C S_1 and S_2 necessary to conflict Your answer is **Correct**

Solution :
(c)
Statements S_1 and S_2 are the necessary conditions but not sufficient to occur conflict during parsing.
So, option (c) is correct.
For eg: Consider the grammar

$$\begin{aligned}S &\to AA\\ A &\to aA \mid b\end{aligned}$$

Both statements S_1 and S_2 are satisfying but still it has not any conflict and it is LR(0).

D None of the above

QUESTION ANALYTICS +

Q. 28

Consider the following SDT, with non-terminals {X, Y, Z, W, M} and {*, id} as terminals.

X → Y * Z {Print 3}
Y → Y * W {Print 3}
Y → W {Print 2}
Z → N + Z {Print 4}
Z → N {Print 2}
N → id {Print 1}
W → id {Print 1}

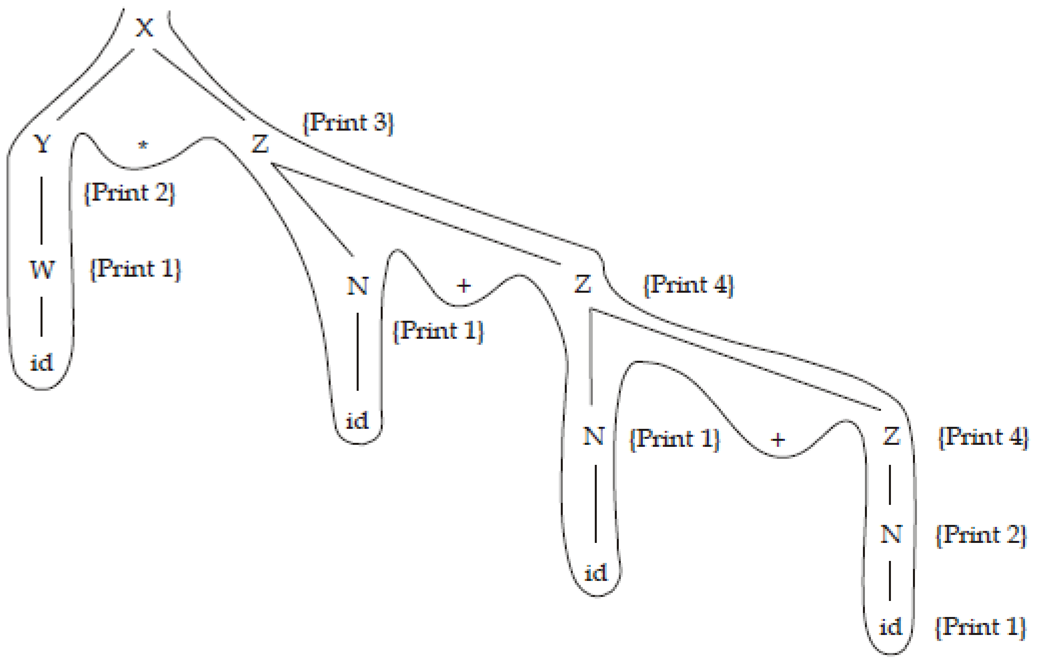
Using the above SDT, the output printed by a bottom-up parser for the input id * id + id + id is:

Have any Doubt ?

A 121112443 Correct Option

Solution :
(a)

Given input id * id + id + id



Output → 121112443

B 211112443

C 421144331

D 122114433 Your answer is **Wrong**

QUESTION ANALYTICS +

Q. 29

Consider the intermediate code given below:

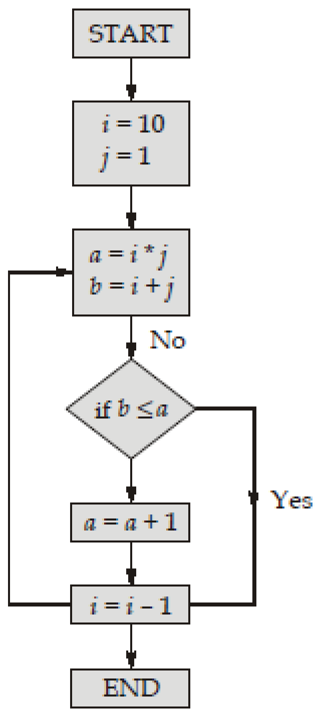
1. $i = 10$
2. $j = 1$
3. $a = i * j$
4. $b = i + j$
5. If $b \le a$ goto 7
6. $a = a + 1$
7. $i = i - 1$ goto 3

X and Y are the number of nodes and edges in the control flow-graph constructed for the above code then the value of (X + Y) is ________.

Have any Doubt ?

15 Correct Option

Solution :
15



Number of nodes X = 7
Number of edges Y = 8
= X + Y = 15

Your Answer is 13

QUESTION ANALYTICS +

Q. 30

The attributes of four arithmetic operators with precedence and associativity is show in the table:

Operator	Precedence	Associativity
*	3	Left
–	4	Left
+	2	Right
↑	1	Right

Note: Higher number has higher precedence.
The value of the expression 10 – 6 – 2 ↑ 8 – 6 + 1 * 2 ↑ 1 + 1 is ________.

Have any Doubt ?

● 65536

65536
Your answer is **Correct**

Solution :
65536

Given expression:

$$\overset{\textcircled{1}}{\underbrace{((10-6)-2)}_{\textcircled{2}}} \uparrow \underbrace{(8-6)}_{\textcircled{3}} + 1 * 2 \uparrow 1 + 1$$

$$\Rightarrow \quad 2 \uparrow 2 + \underbrace{(1 * 2)}_{\textcircled{4}} \uparrow 1 + 1$$

$$\Rightarrow \quad 2 \uparrow \underbrace{(2 + 2)}_{\textcircled{6}} \uparrow \underbrace{(1 + 1)}_{\textcircled{5}}$$

$$\Rightarrow \quad \underbrace{(2 \uparrow \underbrace{(4 \uparrow 2)}_{\textcircled{7}})}_{\textcircled{8}}$$

$$\Rightarrow \quad 2 \uparrow 16$$

$$\Rightarrow \quad 65536$$

QUESTION ANALYTICS +

Q. 31

Consider the given 3-address table for the set of instructions of a basic blocks.

Number	Instruction	Meaning
1	Load a, T_1	$T_1 \leftarrow a$
2	Load b, T_2	$T_2 \leftarrow b$
3	Add T_1, T_2, T_3	$T_3 \leftarrow T_1 + T_2$
4	Load c, T_4	$T_4 \leftarrow c$
5	MUL T_3, T_4, T_5	$T_5 \leftarrow T_3 \times T_4$
6	Store T_1, a	$a \leftarrow T_5$

The minimum number of register are required for the above basic block instructions with no memory spills is ________.

Have any Doubt ?

● 2

Correct Option

Solution :
2

$$\begin{aligned} R_1 &\leftarrow a \\ R_2 &\leftarrow b \\ R_2 &\leftarrow R_1 + R_2 \\ R_1 &\leftarrow c \\ R_1 &\leftarrow R_1 * R_2 \\ a &\leftarrow R_1 \end{aligned}$$

So, 2 registers are required.

● Your Answer is 3

QUESTION ANALYTICS +

Q. 32

A grammar that has not any epsilon productions and also free from unit productions. The maximum number of reduce moves that can be taken during bottom up evaluation of 25 token string by bottom up parsers is ________.

Have any Doubt ?

● 49

Correct Option

Solution :
49

Maximum number of reduce moves for n token = $2n - 1$

So, for 25 tokens $= 2 \times 25 - 1 = 49$

QUESTION ANALYTICS +

Q. 33

Consider the following grammar:

$X \rightarrow YY \mid aXb$
$Y \rightarrow dY \mid f$

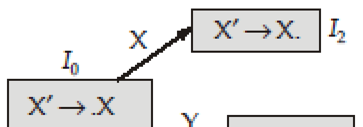
The total number of inadequate states in SLR(1) parsing table of the above grammar are ________.

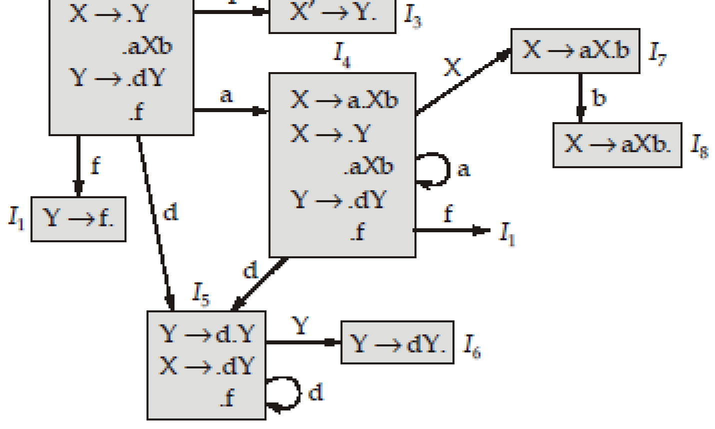
Have any Doubt ?

● 0

Your answer is **Correct**0

Solution :
0





Inadequate state = Number of S-R conflict or RR conflict states

Number of inadequate state = 0

QUESTION ANALYTICS +