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

TOPICWISE : COMPILER DESIGN-1 (GATE - 2019) - REPORTS

OVERALL ANALYSIS | COMPARISON REPORT | SOLUTION REPORT

ALL(17) | CORRECT(3) | INCORRECT(8) | SKIPPED(6)

Q. 1

In a compiler the module that checks the token arrangement against the source code grammar is called _____.

Have any Doubt ?

A Lexical analyzer — Your answer is **Wrong**

B Syntax analyzer — Correct Option

Solution :
(b)
- Lexical analyzer scan the source code as a stream of characters and counts it into meaning full lexemes.
- Syntax analyzer checks the token arrangement against the source code grammar.
- Semantic analyzer check whether the parse tree constructed follows the rules of language.
- Code optimizer do code optimization of the intermediate code.

C Semantic analyzer

D Code optimizer

QUESTION ANALYTICS +

Q. 2

Consider the following grammar:

$S \rightarrow S \times E \mid E$

$E \rightarrow F + E \mid F$

$F \rightarrow id$

Which of the following is true?

Have any Doubt ?

A '×' is right associative but '+' is left associative

B '+' is right associative but '×' is left associative — Your answer is **Correct**

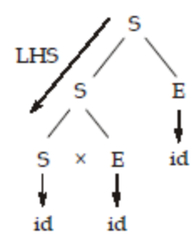
Solution :
(b)

$S \rightarrow S \times E \mid E$

$E \rightarrow F + E \mid F$

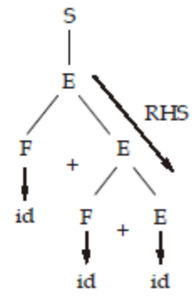
$F \rightarrow id$

1. For expression "id × id × id".



So, '×' is left associative.

2. For expression "id + id + id".



So, '+' is left associative.

C Both '+' and '×' are right associative

D Both '+' and '×' re left associative

QUESTION ANALYTICS +

Q. 3

Which of the following is false?

Have any Doubt ?

A Live variable analysis used in control flow graph for register allocation.

B Basic block does not contain jump into the middle of the block.

C Three address code is linear representation of syntax tree. — Your answer is **Wrong**

D With triples representation optimization can change the execution order. — Correct Option

Solution :
(d)
- With triple, optimization cannot change the execution order but with indirect triple we can.
- Live variable analysis needed in register allocation and deallocation.
- Basic block does not contain jump into middle of the block i.e. sequence of instruction where control enter the sequence at begin and exist at end.
- Three address code is linear representation of syntax tree.

QUESTION ANALYTICS

Q. 4

Consider the following grammar:
$S \rightarrow ZZ$
$Z \rightarrow xZ \mid y$
Which of the following is represent "handle" in the generation of string "$xxxyxy$"?

Have any Doubt ?

A ZxZ

B Zxy

C $xZxy$

D xZ Correct Option

Solution :
(d)
String given: "$xxxyxy$"

$$S \rightarrow ZZ \rightarrow \overset{\text{Handle } \{Z \rightarrow xZ\}}{\overbrace{ZxZ}} \rightarrow Zxy \rightarrow xZxy \rightarrow xxZxy \rightarrow xxxZxy \rightarrow xxxyxy$$

- ZxZ is not handle i.e. cannot reduce to any variable.
- Zxy is not handle i.e. cannot reduce to any variable.
- $xZxy$ is not handle i.e. cannot reduce to any variable.
- xZ is handle since xZ reduce to Z in next step.

QUESTION ANALYTICS

Q. 5

Consider the following statements:
S_1 : Grammar parsed by LL(1) parser must be parsed by SLR(1) parser.
S_2 : Grammar parsed by LL(1) parser must be parsed by CLR(1) parser.
S_3 : Grammar which is not parsed by LALR(1) parser cannot be parsed by LL(1) parser.
Which of the following is true?

FAQ Have any Doubt ?

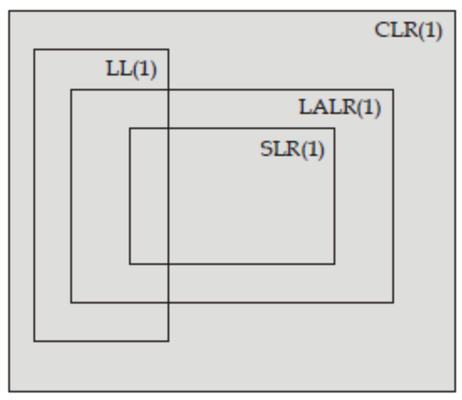
A Only S_1

B Only S_1 and S_2

C Only S_2 and S_3

D Only S_2 Correct Option

Solution :
(d)
Relation between LL(1), SLR(1) and CLR(1) and LALR(1) given below:



S_1 is false, S_2 is true and S_3 is false.

QUESTION ANALYTICS

Q. 6

The number of tokens in the following C-code _____.

```
int main()
{
    int m = 10;
    int n, n1;
    n = ++m;
    n1 = m++;
    n --;
    -- n1;
    n - = n1;
    printf("%d", n);
    return 0;
}
```

FAQ Have any Doubt ?

46 Correct Option

Solution :
46

```
int main ( )
①  ②  ③④
{
⑤
    int m = 10 ;
    ⑥ ⑦⑧⑨⑩
    int n , n1 ;
    ⑪⑫⑬⑭⑮
    n = ++m ;
    ⑯⑰⑱⑲⑳
    n1 = m ++ ;
    ㉑㉒㉓㉔㉕
    n -- ;
    ㉖㉗㉘
    -- n1 ;
    ㉙㉚㉛
    n -= n1 ;
    ㉜㉝㉞㉟
    printf ( "%d" , n ) ;
      ㊱ ㊲ ㊳ ㊴㊵㊶㊷
    return 0 ;
      ㊸ ㊹㊺
}
㊻
```

Number of tokens are 46.

Your Answer is 51

QUESTION ANALYTICS

Q. 7

Consider the following grammar which is not LL(1) because LL(1) table contain multiple entry for same production.

$S \rightarrow aAbB \mid bAaB \mid \in$

$A \rightarrow S$

$B \rightarrow S$

The number of entries have multiple productions in LL(1) table are _____.

Have any Doubt ?

2 — Correct Option

Solution :
2

	FIRST	FOLLOW
S	$\{a, b, \in\}$	$\{a, b, \$\}$
A	$\{a, b, \in\}$	$\{a, b\}$
B	$\{a, b, \in\}$	$\{a, b, \$\}$

LL(1) Parsing table:

	a	b	$\$$
S	$S \rightarrow aAbB$ $S \rightarrow \in$	$S \rightarrow bAbB$ $S \rightarrow \in$	$S \rightarrow \in$
A	$A \rightarrow S$	$A \rightarrow S$	
B	$B \rightarrow S$	$B \rightarrow S$	$B \rightarrow S$

Your Answer is 0

QUESTION ANALYTICS

Q. 8

Assume ×, –, + and/are operators. Precedences and associativity given for those operators as following:
1. × has highest precedence among all operators and it is left associative
2. –, + and/are having equal precedence and they are right associative.
Using × as Multiplication, – as Subtraction, + as Addition and/as Division.
The output of the given expression 3 × 2 – 10 + 5 – 7 × 6 / 3 is _____.

Have any Doubt ?

5 — Correct Option

Solution :
5

Given expression: ((3 × 2) – (10 + (5 – ((7 × 6) / 3))))
= (6 – (10 + (5 – (42/3))))
= (6 – (10 + (5 –14)))
= (6 – (10 – 9))
= (6 – (1))
= 5

Your Answer is 15

QUESTION ANALYTICS

Q. 9

The minimum number of temporary variables are created in 3-address code for the following expression _____.

$a + b \times c + d - e - a + b \times c$

Assume order of precedence from highest to lowest as: ×, + and –. Consider associativity for + and × are not important but – is left associative.

Have any Doubt ?

2 — Correct Option

Solution :
2

Here × is highest and + is next highest.

Associativity does not matter.

Select the best way so that less number of temporary variables will be created.

$a + b \times c + d - e - a + b \times c$

$$= ((a + (b \times c)) + d) - e - (a + (b \times c))$$
$$= (((a + (b \times c)) + d) - e) - (a + (b \times c))$$

Equivalent 3-address code is:

$$t_1 = b \times c$$
$$t_2 = a + t_1$$
$$t_1 = t_2 + d$$
$$t_1 = t_1 - e$$
$$t_1 = t_1 - t_2$$

∴ Only two temporary variables are used.

Your Answer is 3

QUESTION ANALYTICS

Q. 10

Consider the following grammar:

C → PF class id XY
P → public | ∈
F → final | ∈
X → extends id | ∈
Y → implements I | ∈
I → id J
J → , I | ∈

Which of the following is true?

Have any Doubt ?

A FIRST (C) = {public, final}
FOLLOW (X) = {implements}

B FIRST (Y) = {implements, ∈}
FOLLOW (P) = {final}

C FIRST (C) = {public, final, class}
FOLLOW (X) = {implements, $}

Your answer is **Correct**

Solution :
(c)

FIRST (C) = FIRST (PF class id XY)
= {public} ∪ FIRST (F class id XY)
= {public} ∪ {final} ∪ FIRST (class id XY)
= {public} ∪ {final} ∪ {class}
= {public, final, class}
FIRST (X) = FIRST (Y)
= {implements} ∪ FOLLOW (C)
= {implements} ∪ {$}
= {implements, $}
FIRST (Y) = FIRST (implements I) ∪ FIRST (∈)
= {implements, ∈}
FOLLOW (P) = FIRST (F)
= {final} ∪ FIRST (class) = {final, class}

D FIRST (Y) = {implements}
FOLLOW (P) = {final, class}

QUESTION ANALYTICS

Q. 11

Consider the following expression grammar 'G':
A → B | a | CBD
B → C | b
C → A | c
D → d
Which of the following grammar is non-left recursive but is equivalent to G?

FAQ | Have any Doubt ?

A A → aA′ | bA′ | cA′ | cBDA′
A′ → BDA′ | ∈ | BA′
B → C | b
C → A | c
D → d

Your answer is **Wrong**

B A → aA′ | bA′ | cA′
A′ → cBDA′ | BDA′ | ∈
B → C | b
C → A | c
D → d

C A → aA′ | bA′ | cA′
A′ → BDA′ | ∈
B → C | b
C → A | c
D → d

D A → aA′ | bA′ | cA′ | cBDA′

Correct Option

$A' \to BDA' \mid \in$
$B \to C \mid b$
$C \to A \mid c$
$D \to d$

Solution :
(d)

Given grammar:

$A \to B \mid a \mid CBD$ $\quad\quad$ $A \to A \mid a \mid ABD \mid c \mid b \mid cBD$
$B \to C \mid b$ $\quad\quad \equiv \quad\quad$ $B \to C \mid b$
$C \to A \mid c$ $\quad\quad$ $C \to A \mid c$
$D \to d$ $\quad\quad$ $D \to d$

Removing left recursion from $A \to A \mid a \mid b \mid c \mid ABD \mid cBD$
$A \to aA' \mid bA' \mid cA' \mid cBDA'$
$A' \to BDA' \mid \in$
$B \to C \mid b$
$C \to A \mid c$
$D \to d$

QUESTION ANALYTICS

Q. 12

Consider the following SDT:

$C \to C + S \quad \{C.val = \xrightarrow{(1)}\}$
$C \to S \quad \{C.val = S.val\}$
$S \to S \times E \quad \{S.val = \xrightarrow{(2)}\}$
$S \to E \quad \{S.val = E.val\}$
$E \to id \quad \{E.val = id.num\}$

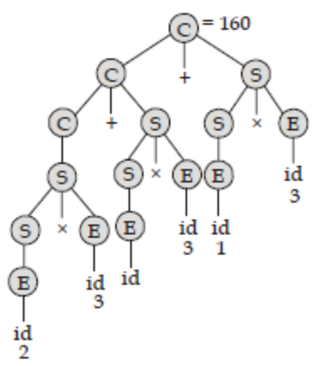
What is the missing translation (1) and (2), if the string "$2 \times 3 + 5 \times 3 + 1 \times 3$" produces 160 instead of 24?

Have any Doubt ?

A $(1): C_1.val + S.val$ **and** $(2): S_1.val - E.val$

B $(1): C_1.val * S.val$ **and** $(2): S_1.val + E.val$ Correct Option

Solution :
(b)



C $(1): C_1.val * S.val$ **and** $(2): S_1.val - E.val$

D $C_1.val + S.val$ **and** $(2): S_1.val * E.val$

QUESTION ANALYTICS

Q. 13

Consider the following grammar to generate binary fractions:

$F \to 0.B \quad \{F.val = B.val\}$
$B_0 \to 0B_1 \quad \{S_1\}$
$B_0 \to 1B_1 \quad \{S_2\}$
$B \to 0 \quad \{B.val = 0\}$
$B \to 1 \quad \{S_3\}$

If the above grammar with semantic rules calculate $\sum_{i=1}^{n} bi2^{-i}$ and each non-terminal has synthesized attribute 'val' to store its value. Then the missing semantic rules will be _____.

Have any Doubt ?

A $S_1 : \{B_0 \cdot val = B_1 \cdot val / 2\}$
$S_2 : \{B_0 \cdot val = B_1 \cdot val / 2^{B_1 \cdot val}\}$
$S_3 : \{B \cdot val = 1\}$

B $S_1 : \{B_0 \cdot val = B_1 \cdot val / 2\}$
$S_2 : \{B_0 \cdot val = B_1 \cdot val + 1 / 2\}$
$S_3 : \{B \cdot val = 1 / 2\}$

C $S_1 : \{B_0 \cdot val = B \cdot val\}$
$S_2 : \{B_0 \cdot val = B_1 \cdot val / 2 + 1 / 2\}$
$S_3 : \{B \cdot val = 1 / 2\}$

D $S_1 : \{B_0 \cdot val = B_1 \cdot val / 2\}$
$S_2 : \{B_0 \cdot val = B_1 \cdot val / 2 + 1 / 2\}$
$S_3 : \{B \cdot val = 1 / 2\}$
Correct Option

Solution :
(d)

Since for every 1 after fractional point represented by $1/2^i$. So, $\{B.val = 1/2\}$, then, for B_0=1B_1 lower bit from fractional side added to $B.val$ i.e., $\{B_0.val = B_1.val/2+1/2\}$.

Finally $B_0 \rightarrow$ old value divide by 2 i.e., $\{B_0.\text{val} = B_1.\text{val}/2\}$

QUESTION ANALYTICS +

Q. 14

Consider the intermediate code given below:

1. $i = 1$
2. $j = 1$
3. $t_1 = 10 \times i$
4. $t_2 = t_1 + j$
5. $t_3 = 8 \times t_2$
6. $t_4 = t_3 - 88$
7. $a[t_4] = 0.0$
8. $i = j + 1$
9. If $j < = 10$ goto (3)
10. $i = i + 1$
11. If $i < = 10$ goto (2)
12. $i = 1$
13. $t_5 = i - 1$
14. $t_6 = 88 \times t_5$
15. $a[t_6] = 1.0$
16. $i = i + 1$
17. If $i < = 10$ goto (13)

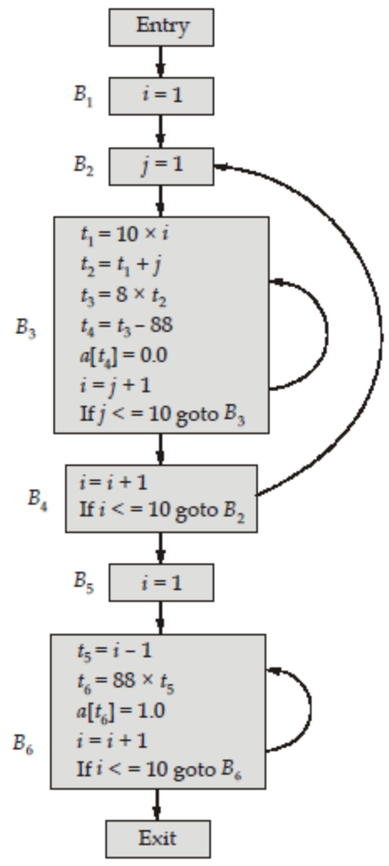
How many nodes and edges in the control flow graph constructed for above three address code?

FAQ | Have any Doubt ?

A 8 and 8

B 8 and 10 — Your answer is **Correct**

Solution :
(b)
Control flow graph will be:



C 7 and 10

D 7 and 9

QUESTION ANALYTICS +

Q. 15

Consider the following code segment:

$a = b + c;$
$c = a + x;$
$d = b + c;$
$b = a + x;$

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

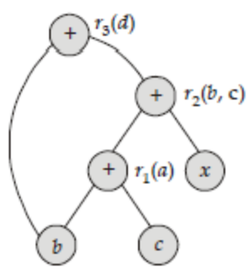
Have any Doubt ?

6 — Correct Option

Solution :
6

$a = b + c;$ $\quad\quad$ $r_1 = b + c;$
$c = a + x;$ $\quad\quad$ $r_2 = r_1 + x;$
$d = b + c;$ $\quad \Rightarrow \quad$ $r_3 = b + r_2;$
$b = a + x;$ $\quad\quad$ $r_2 = r_1 + x;$

It can be verified by making DAG:



Total number of nodes = 6

Your Answer is 3

QUESTION ANALYTICS

Q. 16

A shift reduce parser carries out the actions. Specified within braces immediately after reducing with the corresponding rule of grammar:

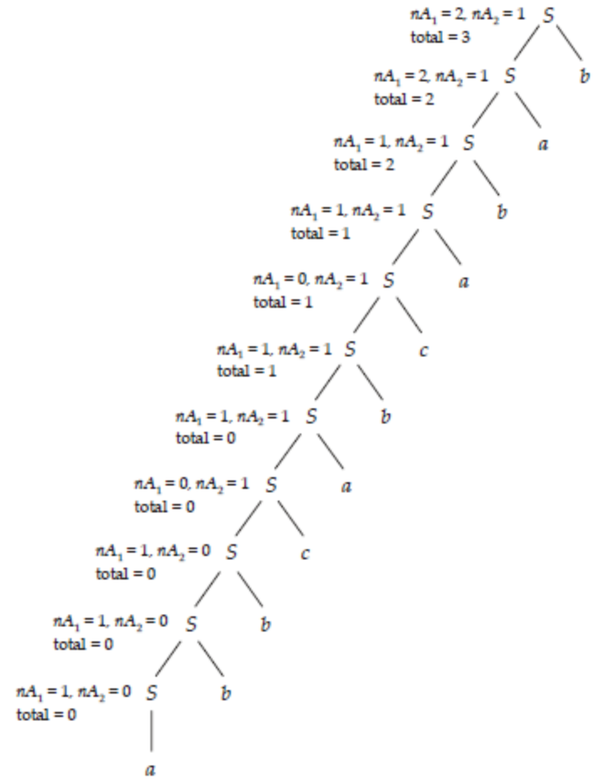
$S_1 \rightarrow S_2\, a$ {$S_1 . nA_1 = S_2 . nA_1 + 1$; $S_1 . nA_2 = S_2 . nA_2$; $S_1 .$ total $= S_2 .$ total;}
$S_1 \rightarrow S_2\, b$ {$S_1 . nA_1 = S_2 . nA_1$; $S_1 . nA_2 = S_2 . nA_2$; $S_1 .$ total $= S_2 .$ total $+ S_2 . nA_2$;}
$S_1 \rightarrow S_2\, c$ {$S_1 . nA_1 = 0$; $S_1 . nA_2 = S_2 . nA_1$; $S_1 .$ total $= S_2 .$ total;}
$S_1 \rightarrow a$ {$S_1 . nA_1 = 1$; $S_1 . nA_2 = 0$; $S_1 .$ total $= 0$;}
$S_1 \rightarrow b$ {$S_1 . nA_1 = 0$; $S_1 . nA_2 = 0$; $S_1 .$ total $= 0$;}
$S_1 \rightarrow c$ {$S_1 . nA_1 = 0$; $S_1 . nA_2 = 0$; $S_1 .$ total $= 0$;}

The final output (S_1 total for) input string "*abbcabcabab*" that corresponds to the pattern $(a(a \mid b)^* c + (a \mid b)^* b)^*$ is _____.

FAQ | Have any Doubt ?

3 Correct Option

Solution :
3



So final value of S_1 . total is 3.

QUESTION ANALYTICS

Q. 17

The following program uses six temporary variables *p, q, r, s, t* and *u*. The code is:

$p = 6$
$q = 7$
$t = p * q$
$s = t + p$
$u = 8$
$u = s * p$
$s = p + u$
$r = r * q$
$t = t + p$
return t

Assuming that all operations take their operands from registers. The minimum number of registers needed to execute this program without spilling is _____.

FAQ | Have any Doubt ?

5 Correct Option

Solution :
5

$r_1 = 6$
$r_2 = 7$
$r_3 = r_1 * r_2$
$r_4 = r_3 + r_1$
$r_5 = 8$
$r_4 = r_4 * r_1$
$r_4 = r_1 + r_4$
$r_4 = r_4 * r_2$
$r_3 = r_3 + r_1$
return r_3

So, total 5 registers are required to execute this program without spilling.

QUESTION ANALYTICS