



Kunal Jha

Course: GATE  
Computer Science Engineering(CS)

HOME

MY TEST

BOOKMARKS

MY PROFILE

REPORTS

BUY PACKAGE

NEWS

TEST SCHEDULE

## COMPILER DESIGN-1: (GATE - 2021) - REPORTS

OVERALL ANALYSIS

COMPARISON REPORT

SOLUTION REPORT

ALL(17)

CORRECT(0)

INCORRECT(0)

SKIPPED(17)

Q. 1

FAQ

Have any Doubt ?



Consider the following statements:

- $S_1$  : Three address code is linear representation of syntax tree.  
 $S_2$  : With triples representation optimization can change the execution order.
- Which of the above is correct?

 A Only  $S_1$ 

Correct Option

Solution :

- (a)
- $S_1$  is correct.
- With triple, optimization cannot change the execution order but with indirect triple we can.

 B Only  $S_2$  C Both  $S_1$  and  $S_2$ 

D None of these

QUESTION ANALYTICS



Q. 2

FAQ

Have any Doubt ?



Consider the following grammar:

$$X \rightarrow YZ \quad [Z.x = X.x]$$

$$X.y = Y.y]$$

$$Z \rightarrow PZ' \quad [Z.x = P.x]$$

$$Z.y = P.x + Z'.y]$$

Which of the following is true?

A Both x and y are inherited attributes.

B Both x and y are synthesized attributes.

C x is inherited and y is synthesized.

Correct Option

Solution :

- (c)
- x is inherited.
- y is synthesized.

D x is synthesized and y is inherited.

QUESTION ANALYTICS



Q. 3

Have any Doubt ?



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

A Compiler

B High level language

C Bootstrap loader

D Linker

Correct Option

Solution :

- (d)

QUESTION ANALYTICS



Q. 4

FAQ

Have any Doubt ?



Choose the correct sequence of occurrence during compilation process.

A Parse tree → Token stream → Intermediate code

B Parse tree → 3 address code → Character stream

C Character stream → Parse tree → SDT tree

Correct Option

Solution :

(c)  
Lexical analyzer → Syntax analyzer → Semantic analyzer → Intermediate code → Code optimizer.

D Token stream → SDT tree → Parse tree

QUESTION ANALYTICS



Q. 5

? FAQ Have any Doubt ?



Consider the following SDT:

X → P

P → (Print \*)P1 \* Q/Q

Q → (Print '+')Q1+R/R

R → (P)/digit(Print (digit.val))

Which of the following does by above SDT?

A Evaluating Prefix Expression

B Evaluating Infix Expression

C Infix to Postfix

D None of these

Correct Option

Solution :

(d)  
SDT not converting neither to postfix nor prefix.

QUESTION ANALYTICS



Q. 6

? FAQ Have any Doubt ?



Consider the following statements:

I. In a bottom-up evaluation of a SDT, inherited attribute can always be evaluated.

II. In a bottom-up evaluation of a SDT, inherited attribute can evaluated only if definition in L-attributed.

The number of the correct statements is/are \_\_\_\_\_.

1

Correct Option

Solution :

1  
Only II statement is true.

QUESTION ANALYTICS



Q. 7

? FAQ Have any Doubt ?



Consider the following code:

```
main()
{
    int temp = 200, 10;
    int l1, l2;
    temp = ++temp;
    l1 = l2
    printf("%d", temp + l1);
}
```

The number of tokens in the above code is \_\_\_\_\_.

34

Correct Option

Solution :

```
34
main( )
① ②③
{
④
    int temp = 200 , 10 ;
    ⑤ ⑥ ⑦ ⑧ ⑨ ⑩⑪
    int l1 , l2 ;
    ⑫⑬⑭⑮⑯
    temp = ++ temp ;
    ⑰ ⑱ ⑲ ⑳ ⑳
    l1 + = l2
```

```

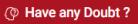
    22 23 24
    printf ("% d", temp + l1);
    25 26 27 28 29 30 31 32 33
}

```

 QUESTION ANALYTICS



Q. 8

? FAQ 



In which of the following phase(s) of the compiler FSA (Finite State Automata) is used (Choose all the options)?

A Lexical analysis

Correct Option

B Syntax analysis

C Code generation

D Code optimization

YOUR ANSWER - NA

CORRECT ANSWER - a

STATUS - SKIPPED

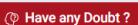
Solution :

(a) Because the lexer performs its analysis by going from one stage to another.

 QUESTION ANALYTICS



Q. 9

? FAQ 



Which of the following is/are NOT the function of the syntax phase?

A Recognize the language and to call the appropriate action routines that will generate the intermediate form or matrix for these constructs.

Correct Option

B Build a literal table and an identifier table.

Correct Option

C Build a uniform symbol table.

Correct Option

D Parse the source program into the basic elements or tokens of the language.

Correct Option

YOUR ANSWER - NA

CORRECT ANSWER - b,c,d

STATUS - SKIPPED

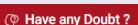
Solution :

(b, c, d)  
Syntax analysis phase generate parse tree based on grammar and call the appropriate subroutines.

 QUESTION ANALYTICS



Q. 10

? FAQ 



Consider the regular expression with the respective token number in the table.

REX	Token No.
$(a + b)^* c$	1
$ca^* b$	2
$c^*$	3

Choose the correct output when lexical analyzer scans the following input : "cabaccab" Note: The analyzer tries to output the token that matches the longest possible prefix.

A 3122

B 2132

C 1132

D Generates lexical error

Correct Option

Solution :

(d)  
 $\frac{c}{2} \frac{ab}{1} \frac{a}{3} \frac{c}{no match}$

Hence, lexical error will generate.

 QUESTION ANALYTICS







Kunal Jha

 Course: GATE  
 Computer Science Engineering(CS)

[HOME](#)
[MY TEST](#)
[BOOKMARKS](#)
[MY PROFILE](#)
[REPORTS](#)
[BUY PACKAGE](#)
[NEWS](#)
[TEST SCHEDULE](#)

## COMPILER DESIGN-1: (GATE - 2021) - REPORTS

[OVERALL ANALYSIS](#)   [COMPARISON REPORT](#)   **SOLUTION REPORT**
[ALL\(17\)](#)   [CORRECT\(0\)](#)   [INCORRECT\(0\)](#)   [SKIPPED\(17\)](#)
**Q. 11**
[FAQ](#)   [Have any Doubt?](#)


Consider the following translation scheme:

$$\begin{aligned} S &\rightarrow XY \\ X &\rightarrow X * Y \text{ [Print (*)]} \\ X &\rightarrow \text{id} \text{ [Print (id)]} \\ Y &\rightarrow +Y \text{ [Print (-)]} \\ Y &\rightarrow \text{id} \text{ [Print (id)]} \end{aligned}$$

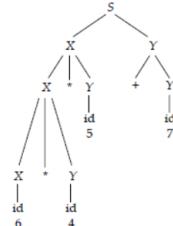
 Here id is a token which represent an integer and id represent the value of that integer. For an input  $6 * 4 * 5 + 7$ , this translation scheme prints

**A**  $64 * 5 * 7 +$ 
**B**  $6 * 4 * 5 - 7$ 
**C**  $64 * 5 * 7 -$ 

Correct Option

**Solution :**

(c)


 Output :  $64 * 5 * 7 -$ 
**D**  $64 * 5 - * 7 -$ 
[QUESTION ANALYTICS](#)

**Q. 12**
[FAQ](#)   [Have any Doubt?](#)


Consider the following C program:

**Program 1:**  

```
#include <stdio.h>
int main()
{
    float x;
    while (x ≥
        printf ("%d",x);
}
```

**Program 2:**  

```
#include <stdio.h>
int main()
{
    int temp, _temp2;
    @_//Hello World
    return 0;
}
```

Which of the above gives lexical error?

**A** Only Program 1

**B** Only Program 2

Correct Option

**Solution :**

(b)

- Only Program 2 generates lexical error.
- Program1 generates semantic error by syntax analyzer.

**C** Both Program 1 and Program 2

**D** None of these

[QUESTION ANALYTICS](#)


Q. 13

FAQ Have any Doubt ?

Consider the following statements:

- S<sub>1</sub>: While program in execution, access to heap memory is slower as compared to accessing variables allocated on stack.
- S<sub>2</sub>: While program in execution, in a multithread situation, each thread has its own stack and share a common heap memory.
- S<sub>3</sub>: During a program execution, heap is stored in main memory and stack is present in secondary memory.

Which of the above is incorrect?

A Only S<sub>1</sub> and S<sub>2</sub>

B Only S<sub>2</sub> and S<sub>3</sub>

C All of these

D Only S<sub>3</sub>

Correct Option

**Solution :**

- (d)
- Statement S<sub>1</sub> and S<sub>2</sub> are correct.
  - Statement S<sub>3</sub> is incorrect. Heap and stack both are present in main memory.

QUESTION ANALYTICS



Q. 14

FAQ Have any Doubt ?

Consider the basic block given below:

$$X \rightarrow X * Y$$

$$Z \rightarrow X + Z$$

$$P \rightarrow Z/P$$

$$X \rightarrow Z + P$$

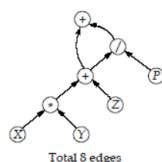
Minimum number of edges present in the DAG representation of the above block is \_\_\_\_\_.

8

Correct Option

**Solution :**

8



QUESTION ANALYTICS



Q. 15

FAQ Have any Doubt ?

Consider the following statements:

- I. Three address code is a linearized representation of syntax tree.
- II. Type checking is done during all the phases especially in syntax analysis phase.
- III. Target code generation phase is machine independent code generation.
- IV. Symbol table is accessed during lexical, syntax and semantic analysis phase.

The number of the correct statements is/are \_\_\_\_\_.

2

Correct Option

**Solution :**

2

- Statement I and IV is correct.
- Type checking is done at semantic analysis phase.
- Target code generation is dependent based on the machine.
- Symbol table is accessed during lexical, syntax and semantic analysis phase.

QUESTION ANALYTICS



Q. 16

FAQ Have any Doubt ?

Which of the following functions is/are performed by loader?

A Allocate memory for the programs and resolve symbolic references between objects decks.

Correct Option

B Address dependent locations, such as address constants, to correspond to the allocated space.

Correct Option

C Physically place the machine instructions and data into memory.

Correct Option

D Linking in the program.

VALID ANSWER

CORRECT ANSWER

STATUS

SKIPPED

YOUR ANSWER - NA

CORRECT ANSWER - a,b,c

STATUS - SKIPPED

**Solution :**

(a, b, c)  
A loader is the part of an operating system that is responsible for loading programs and libraries.

 QUESTION ANALYTICS



Q. 17

? FAQ

Have any Doubt ?



Which of the following statement(s) is/are true?

A Token is a pair consisting of a token name and an optional attribute value.

Correct Option

B A pattern is a description of the form that the lexemes of a token may take.

Correct Option

C A lexeme is a sequence of characters in the source program that matches the pattern for a token.

Correct Option

D Lexemes are identified by the semantic analyzer as an instance of that token.

YOUR ANSWER - NA

CORRECT ANSWER - a,b,c

STATUS - SKIPPED

**Solution :**

(a, b, c)  
Lexemes are identified by the lexical analyzer as an instance of that token.  
Hence only statement (d) is false.

 QUESTION ANALYTICS



Item 11-17 of 17 [« previous](#) [1](#) [2](#) [next »](#)



Kunal Jha  
 Course: GATE  
 Computer Science Engineering(CS)

HOME  
 MY TEST  
 BOOKMARKS  
 MY PROFILE

REPORTS

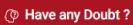
BUY PACKAGE  
 NEWS  
 TEST SCHEDULE

## COMPILER DESIGN-2: (GATE - 2021) - REPORTS

OVERALL ANALYSIS    COMPARISON REPORT    **SOLUTION REPORT**

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

Q. 1

? FAQ    

In SLR parsing for the grammar:

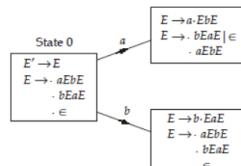
$E \rightarrow E$   
 $E \rightarrow aEbE \mid bEaE \mid \epsilon$

In state 0, for inputs 'a' and 'b'

- A** Both will have shift-reduce conflict.

Correct Option

Solution :  
 (a)



In state 0, there is reduce  $E \rightarrow \epsilon$  which will go under follow of ' $E$ ' which { $a, b$ } and also at state 0. There is shift at ' $a$ ' and ' $b$ '. Hence, there is shift-reduce conflict.

- B** Only 'a' will have shift-reduce conflict.

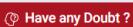
- C** Only 'b' will have shift-reduce conflict.

- D** Neither of the other options

QUESTION ANALYTICS

+

Q. 2

? FAQ    

In SLR passing to get a shift-reduce conflict for state-I on terminal symbol 'a'.

- A**  $A \rightarrow a.\beta$  with First ( $\beta$ ) containing 'a' should be in  $I$ .

- B**  $A \rightarrow \delta.$  in  $I$  with 'Follow A having 'a''.

- C**  $A \rightarrow a.\beta$  with first ( $\beta$ ) containing 'a' should be in  $I$  and  $A \rightarrow \delta.$  be in  $I$  with follow A having 'a'.

Correct Option

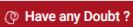
Solution :  
 (c)

- D** None of these

QUESTION ANALYTICS

+

Q. 3

? FAQ    

Construction of parsing table in which strategies do not need the follow set?

- A** SLR and Canonical LR

- B** Canonical LR and LALR

Correct Option

Solution :  
 (b)

- C** SLR and LALR

- D** None of these

QUESTION ANALYTICS

+

Q. 4

FAQ Have any Doubt ?

Which of the following statements is true?

- A  $S \rightarrow aabc/ab$ , this grammar is not LL(1) but it is LL(2).
- B Every regular language is LL(1).
- C Every regular grammar is LL(1).
- D Both (a) and (b)

Correct Option

**Solution :**

(d)

- $S \rightarrow aabc \mid ab$
- There is left factoring in LL(1). Hence, not LL(1), but it is LL(2).
- Every regular language is LL(1) is true. There exist a regular grammar which is LL(1).
- Every regular grammar is LL(1) is false, because regular grammar may contain left recursion, left factoring, ambiguity.



Q. 5

FAQ Have any Doubt ?

Consider the following grammar G1 and G2 with S, A, B, C as non-terminals and a, b, c,  $\epsilon$  as terminals.

$$G_1 : S \rightarrow A + B \mid A \mid B \mid AB$$

$$\begin{aligned} A &\rightarrow A^*C \mid a \\ B &\rightarrow B + C \mid b \\ C &\rightarrow c \end{aligned}$$

$$G_2 : S \rightarrow A^*B \mid \epsilon$$

$$\begin{aligned} S &\rightarrow B - C \\ A &\rightarrow a \\ B &\rightarrow b \\ C &\rightarrow c \end{aligned}$$

Which of the above grammar is operator grammar?

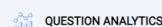
- A Only  $G_1$
- B Only  $G_2$
- C Both  $G_1$  and  $G_2$
- D None of these

Correct Option

**Solution :**

(d)

- A Grammar G is said to be operator grammar if
- (a) it does not contain null production.
  - (b) it does not contain 2 adjacent variable on right hand side.
- So, both  $G_1$  and  $G_2$  are not operator grammar.



Q. 6

FAQ Have any Doubt ?

Which of the following statements:

$S_1$  : Grammar which is not parsed by LALR(1) parser cannot be parsed by LL(1) parser.

$S_2$  : Grammar parsed by LL(1) parser must be parsed by CLR(1) parser.

$S_3$  : There is only 1 lookahead symbol in LALR.

The number of the correct statements is/are \_\_\_\_\_.

 2

Correct Option

**Solution :**

(2)

- Statement  $S_2$  and  $S_3$  are correct only.
- $S_1$  is incorrect. It is not always that a grammar is not parsed by LALR(1) parser cannot be parsed by LL(1) parser.



Q. 7

FAQ Have any Doubt ?

Consider the following grammar:

$$X \rightarrow YY \mid aXb$$

$$Y \rightarrow dMf$$

Variables X, Y are non-terminals and a, b, d, f are terminals. Consider the below strings.

- (i) affb
- (ii) ff
- (iii) df

The number of the above strings which are the viable prefixes is/are \_\_\_\_\_.

 1

Correct Option

...  
...

**SOLUTION :**

1  
A handle  $Y \rightarrow f$  is present. So, any string having ' $f$ ' followed by any character can not be viable prefixes.  
So, clearly (i) and (ii) are not viable prefixes.  
So, total 1 prefixes possible which is (iii).

**QUESTION ANALYTICS****Q. 8****? FAQ** **Have any Doubt ?**

Which of the following statements are CORRECT?

**A** Static allocation of all data areas by a compiler makes it impossible to implement recursion.**Correct Option****B** Automatic garbage collection is essential to implement recursion.**C** Dynamic allocation of activation records is essential to implement recursion.**Correct Option****D** Both heap and stack are essential to implement recursion.**YOUR ANSWER - NA****CORRECT ANSWER - a,c****STATUS - SKIPPED****Solution :**

(a, c)  
To implement recursion, activation record should be implemented by providing dynamic memory allocation. This dynamic allocation is done from run-time stack. Heap is essential to allocate memory for data structures at run-time, not for recursion.  
So, statement (a) and (c) are correct.

**QUESTION ANALYTICS****Q. 9****? FAQ** **Have any Doubt ?**

Which of the following statement is correct?

**A** Lexical analysis phase is responsible for eliminating void space character.**Correct Option****B** Semantic analyzer help in giving error message by giving row number and column number.**C** Even though CLR(1) don't have R-R conflict but LALR(1) may contain R-R conflict.**Correct Option****D** Even though CLR(1) don't have S-R conflict but LALR(1) may contain S-R conflict.**YOUR ANSWER - NA****CORRECT ANSWER - a,c****STATUS - SKIPPED****Solution :**

(a, c)  
As lexical analysis phase help in eliminating void space character like blank, tab, new line etc. So (a) is true. Lexical analyzer is responsible for giving error message by providing row and column no and not the semantic analyzer. So (b) is false.  
(c) is True and (d) is false.

**QUESTION ANALYTICS****Q. 10****? FAQ** **Have any Doubt ?**

Consider the following grammar:

 $S \rightarrow aAbB \mid bAaB \mid \epsilon$  $A \rightarrow S$  $B \rightarrow S$ 

Which of the following option is correct?

**A** Follow ( $S$ ) =  $\{\$$   
Follow ( $A$ ) =  $\{a, b, \$\}$   
Follow ( $B$ ) =  $\{a, b, \$\}$ **B** First ( $S$ ) =  $\{\epsilon\}$   
First ( $A$ ) =  $\{a, b\}$   
First ( $B$ ) =  $\{a, b\}$ **C** Follow ( $S$ ) =  $\{\$, a, b\}$   
Follow ( $A$ ) =  $\{a, b\}$   
First ( $B$ ) =  $\{a, b, \epsilon\}$ **Correct Option****Solution :**

(c)

Follow ( $S$ ) =  $\{\$, a, b\}$   
Follow ( $A$ ) =  $\{a, b\}$   
First ( $B$ ) =  $\{a, b\}$   
First ( $S$ ) =  $\{a, b, \epsilon\}$  = First ( $A$ ) = First ( $B$ )

**D** Both (a) and (b)





Kunal Jha  
 Course: GATE  
 Computer Science Engineering(CS)

HOME  
 MY TEST  
 BOOKMARKS  
 MY PROFILE

REPORTS

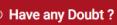
BUY PACKAGE  
 NEWS  
 TEST SCHEDULE

## COMPILER DESIGN-2: (GATE - 2021) - REPORTS

OVERALL ANALYSIS   COMPARISON REPORT   **SOLUTION REPORT**

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

Q. 11

? FAQ   



Consider the following grammar:

$$S \rightarrow Px \mid yPz \mid mz \mid ymx$$

$$P \rightarrow n$$

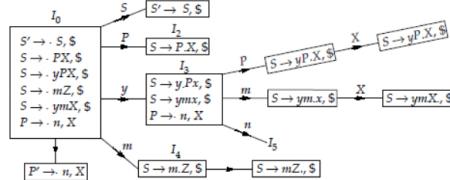
Variables S, P are non-terminals and rest are terminals. Which of the following is correct about the above grammar?

**A** SLR (1) and LALR (1)

Correct Option

Solution :

(a)



There is no conflict at any state. Hence, it is LR(0), SLR(1), LR(1), CLR(1) and LALR(1).

**B** CLR (1) but not LR (0)

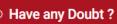
**C** CLR (1) but not LALR (1)

**D** LALR (1) but not SLR (1)

QUESTION ANALYTICS



Q. 12

? FAQ   



Consider the grammar:

$$S \rightarrow P^* Q$$

$$P \rightarrow R - P / R$$

$$R \rightarrow P + R / P$$

$$P \rightarrow a$$

$$Q \rightarrow b$$

$$R \rightarrow c$$

Which of the following is true?

**A** '+' is left associative and has higher precedence than '-'.

**B** '+' is right associative and has higher precedence than '\*'.

Correct Option

Solution :

- (b)  
 + → Right associative  
 - → Right associative  
 Precedence → + > - > \*

**C** '-' is right associative and has higher precedence than '+'.

**D** Both '+' and '-' are left associative and have same precedence order.

QUESTION ANALYTICS



Q. 13

? FAQ   



Consider the following grammar:

$$S \rightarrow SABC$$

$$A \rightarrow aAd \mid \epsilon$$

$$B \rightarrow bB \mid \epsilon$$

$$C \rightarrow Cc \mid C\epsilon \mid \epsilon$$

Variables S, A, B, C are non-terminals and a, b, c, d, e are terminals.

What is the number of elements in FOLLOW(S)?

**A** 2

**B** 3

—

C 4

D 5

Correct Option

**Solution :**  
(d)

$$\text{FOLLOW}(S) = \{a, b, c, e, \$\}$$

Total 5 elements are there.

QUESTION ANALYTICS



Q. 14

FAQ

Have any Doubt ?



Consider the following C statements:

- $S_1$  : Whi(x);
- $S_2$  : for(x, y, z)
- $S_3$  : float f = x;

The number of correct statements which will give syntax error is/are \_\_\_\_\_?

1

Correct Option

**Solution :**

- 1
- $S_1$  is correct as it can be treated as function.
- $S_2$  gives syntax error as for() is used for loop and should be properly defined.
- $S_3$  is syntactically correct.

QUESTION ANALYTICS



Q. 15

FAQ

Have any Doubt ?



Consider the following CFG:

$$\begin{aligned} S &\rightarrow PxQ \mid xP \\ P &\rightarrow yQ \mid Q \\ Q &\rightarrow xQ \mid x \end{aligned}$$

Variables S, P, Q are non-terminals and rest are terminals. Consider the following statements:

- (i) The grammar G is parsed by LL(1).
- (ii) The grammar G is parsed by SLR(1).
- (iii) The grammar G is parsed by CLR(1).

The number of the correct statements is/are \_\_\_\_\_.

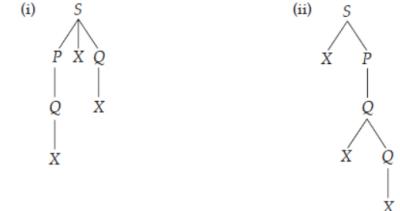
0

Correct Option

**Solution :**

0

Grammar G is ambiguous as string "xxx" is generated with 2-tree.



Two different parse tree can be generated. Hence, it cannot be LL(1), SLR(1) and CLR(1). Therefore, no statements are correct.

QUESTION ANALYTICS



Q. 16

FAQ

Have any Doubt ?



Which one of the following statement is/are true for the SLR(1) and LALR(1) parsing tables for a context free grammar?

A The reduce entries in both the tables may be different.

Correct Option

B The error entries in both the tables may be different.

Correct Option

C The go to part of both tables may be different.

D The shift entries in both the tables are always identical.

Correct Option

YOUR ANSWER - NA

CORRECT ANSWER - a,b,d

STATUS - SKIPPED

**Solution :**

- (a, b, d)
- Only option (c) is false since the GoTo part remains same.

QUESTION ANALYTICS





Which one of the following statement is/are false?

- A Canonical LR parser is more powerful than LALR parser.
- B SLR parser is more powerful than LALR. Correct Option
- C LALR parser is more powerful than canonical LR parser. Correct Option
- D SLR parser, canonical LR parser and LALR parser all have the same power. Correct Option

YOUR ANSWER - NA

CORRECT ANSWER - b,c,d

STATUS - SKIPPED

Solution :

(b, c, d)  
LALR parser is more powerful thus option (b) is false.  
CLR parser is more powerful than LALR so option (c) is also false.  
Option (d) is also false with same reason which described above.

QUESTION ANALYTICS





Kunal Jha

Course: GATE  
Computer Science Engineering(CS)

HOME

MY TEST

BOOKMARKS

MY PROFILE

REPORTS

BUY PACKAGE

NEWS

TEST SCHEDULE

## COMPILER DESIGN (GATE - 2021) - REPORTS

OVERALL ANALYSIS

COMPARISON REPORT

SOLUTION REPORT

ALL(33)

CORRECT(0)

INCORRECT(0)

SKIPPED(33)

Q. 1

FAQ

Have any Doubt?



The identification of common sub-expression and replacement of run time computations by compile time computations is

 A Local optimization B Constant folding

Correct Option

Solution :

(b)  
By common sub expression we mean- the expression which is already computed but during compilation it again appears for computation. In constant folding - the expression which is having a constant value at the time of compilation it can be referred to its respective value or we can say that it can be replaced with its similar value. It is constant folding where the replacement is done during compilation time. C Loop optimization D Data flow analysis

QUESTION ANALYTICS



Q. 2

FAQ

Have any Doubt?



The graph that shows basic blocks and their successor relationship is called:

 A DAG B Control graph C Flow graph

Correct Option

Solution :

(c)

 D Hamiltonian graph

QUESTION ANALYTICS



Q. 3

Have any Doubt?



A top down parser generates

 A Left most derivation

Correct Option

Solution :

(a)  
"Parsing is classified into two categories, i.e. Top Down Parsing and Bottom-Up Parsing. Top- Down Parsing is based on Left Most Derivation whereas Bottom Up Parsing is dependent on Reverse Right Most Derivation". B Right most derivation C Left most derivation in reverse D Right most derivation in reverse

QUESTION ANALYTICS



Q. 4

Have any Doubt?



The output of lexical analyzer is

 A A set of regular expressions B Strings of character

**C** Syntax tree**D** Set of tokens

Correct Option

**Solution :**

(d)

Lexical analyser uses the regular expressions as defined by the language specification to output a set of tokens for the syntax analyser whose output is a syntax tree after matching the syntax.

## QUESTION ANALYTICS +

Q. 5

Have any Doubt ?



Which data structure is used by the compiler for managing variables and their attributes?

**A** Binary tree**B** Link list**C** Symbol table

Correct Option

**Solution :**

(c)

Symbol tables are data structures that are used by compilers to hold information about source program constructs. The information is collected incrementally by the analysis phases of a compiler and used by the synthesis phases to generate the target code. Entries in the symbol table contain information about an identifier such as its character string (or lexeme), its type, its position in storage, and any other relevant information. It can be implemented by using an array, hash table, tree and even some time with the help of the linked list.

**D** Parse table

## QUESTION ANALYTICS +

Q. 6

Have any Doubt ?



What would be the output of the following program:

```
#include <stdio.h>
#define a (x + 1)
int x = 2;
void b( )
{
    x = a;
    printf("%d\n", x);
}
void c( )
{
    int x = 1;
    printf("%d\n", a);
}
void main() {b(); c();}
```

**A** 3  
2

Correct Option

**Solution :**

(a)

Compiler will remove all a's and replace it with (x + 1) Thus, program would become like given below:

```
#include <stdio.h>
int x = 2;
void b( )
{
    x = (x + 1);
    printf("%d\n", x);
}
void c( )
{
    int x = 1;
    printf("%d\n", (x + 1));
}
void main() {b(); c();}
Thus, output will be:
```

3  
2**B** 3  
1**C** 3  
Garbage value**D** 1  
3

## QUESTION ANALYTICS +

Q. 7

[Have any Doubt ?](#)

Replacing the expression  $4 * 2.14$  by  $8.56$  is known as

 A Constant folding

Correct Option

Solution :

- (a) It's constant folding.

Constant folding: Replacing the value of expression before compile time.

Here the value of  $4 * 2.14$  is replaced by  $8.56$ . So it's constant folding.

 B Induction variable C Strength reduction D Code reduction

+

Q. 8

[FAQ](#) [Have any Doubt ?](#)

Which one of the following string can definitely said to be a token without looking at the next input:

 A + B return C \* D Yellow

Correct Option

Solution :

- (d)

(a)  $(++, +=)$  Thus need to look at the next input since there are many choices.

(b) (return a) Thus need to look at the next input because can be a keyword or identifier in C language.

(c)  $(^=)$

(d) Yellow is not a keyword and must be a identifier thus need not to look at the next input.

+

Q. 9

[FAQ](#) [Have any Doubt ?](#)

In any Chomsky Normal Form grammar G, how many nodes will parse tree have including leaf nodes for terminal strings of length 4? \_\_\_\_\_.

 11

Correct Option

Solution :

11

CNF form:

 $S \rightarrow SA | a$  $A \rightarrow a$ Total number of nodes for string length  $n = 3n - 1$ .Here given  $n = 4$ , so total nodes =  $3 \times 4 - 1 = 11$ .

+

Q. 10

[FAQ](#) [Have any Doubt ?](#)

Consider following grammar:

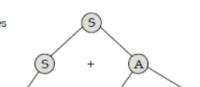
 $S \rightarrow S + A | A$  $A \rightarrow D - A | D$  $D \rightarrow D^* B | B$  $B \rightarrow \text{num}$ The number of internal nodes for the parse tree for  $5 \times 4 + 10 \times 6 - 7 - 8$  \_\_\_\_\_. 18

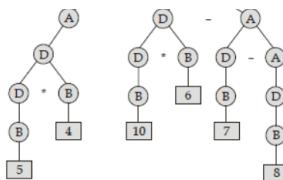
Correct Option

Solution :

18

Internal nodes  
 Leaf nodes





Parse tree for  $5 \times 4 + 10 \times 6 - 7 - 8$  based on above grammar will contain 18 internal nodes and 6 leaf nodes.

 QUESTION ANALYTICS

Item 1-10 of 33 [« previous](#) [1](#) [2](#) [3](#) [4](#) [next »](#)



Kunal Jha

 Course: GATE  
 Computer Science Engineering(CS)

[HOME](#)
[MY TEST](#)
[BOOKMARKS](#)
[MY PROFILE](#)
[REPORTS](#)
[BUY PACKAGE](#)
[NEWS](#)
[TEST SCHEDULE](#)

## COMPILER DESIGN (GATE - 2021) - REPORTS

[OVERALL ANALYSIS](#)
[COMPARISON REPORT](#)
[SOLUTION REPORT](#)
[ALL\(33\)](#)
[CORRECT\(0\)](#)
[INCORRECT\(0\)](#)
[SKIPPED\(33\)](#)
**Q. 11**
[FAQ](#)
[Have any Doubt ?](#)


Consider the following grammar:

 $S \rightarrow aSBId$ 
 $B \rightarrow b$ 

The number of reduction steps taken by a bottom-up parser while accepting the string aaadbdbb is \_\_\_\_\_.

**7**

Correct Option

**Solution :**

7

Given string: aaadbdbb

Bottom-up parsers: rightmost derivation in reverse.

 $S \rightarrow aSB$ 
 $S \rightarrow aSb$ 
 $S \rightarrow aaSb$ 
 $S \rightarrow aaSbb$ 
 $S \rightarrow aaaSBbb$ 
 $S \rightarrow aaaSbbb$ 
 $S \rightarrow aaadbdbb$ 

So, there are 7 steps.

[QUESTION ANALYTICS](#)

**Q. 12**
[FAQ](#)
[Have any Doubt ?](#)


The number of tokens in the following C code segment is

switch (inputvalue)

{

case 1 : b = c\*d;

break;

default : b = b++;

break;

}

**26**

Correct Option

**Solution :**

26

The number of tokens are:

1. switch

2.

3. inputvalue

4.

5.

6. case

7. 1

8. :

9. b

10. =

11. c

12. \*

13. d

14. ;

15. break

16. ;

17. default

18. :

19. b

20. =

21. b

22. ++

23. ;

24. break

25. ;

26. }

[QUESTION ANALYTICS](#)

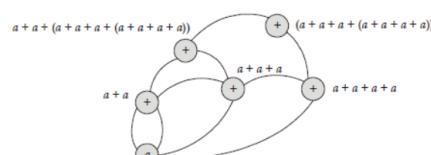
**Q. 13**
[FAQ](#)
[Have any Doubt ?](#)

 Construct the DAG in the expression given below and find out the number of nodes in it, assuming + associates from the left.  
 $a + a + ((a + a + a + (a + a + a + a)))$ 
**6**

Correct Option

**Solution :**

6



Total number of nodes = 6.

#### QUESTION ANALYTICS

Q. 14

FAQ Have any Doubt ?



Which of the following statement(s) is/are true?

A A function of a linker is to combine several object modules into a single load module.

Correct Option

B A function of a linker is to replace absolute references in an object module by symbolic references to locations in other modules.

C Every regular grammar is LL(1).

D Every regular set has a LR(1) grammar.

Correct Option

YOUR ANSWER - NA

CORRECT ANSWER - a,d

STATUS - SKIPPED

Solution :

(a, d)

A linker is a computer software that combine 2 or more file generated by compiler into a single executable file. So its function is to combine several object modules into a single load module. Hence, (a) is true and (b) is false.

There is a difference between Left linear grammar and left recursion. Regular languages are either Left linear or Right linear. LL(1) does not accept left recursion. Left recursion is a kind of left-linear grammar. Every left linear grammar can be converted into equivalent right linear grammar. Hence, (d) is true and (c) is false.

#### QUESTION ANALYTICS

Q. 15

FAQ Have any Doubt ?



Consider the grammar shown below:

$S \rightarrow SS + SS * a$

Then which of the following statement is/are true with respect to the above grammar?

A First (s) = {+}

Correct Option

B First (s) = {a}

C Follow set (s) = {+, \*, \$}

D Follow set (s) = {+, \*, \$, a}

Correct Option

YOUR ANSWER - NA

CORRECT ANSWER - b,d

STATUS - SKIPPED

Solution :

(b, d)

First (s) = {a}

Follow set (s) = {+, \*, \$, a}

#### QUESTION ANALYTICS

Q. 16

FAQ Have any Doubt ?



In a string of length  $n$ , Which of the following is/are True?

A Number of Prefixes are  $n$ .

Correct Option

B Number of Suffixes are  $n + 1$

C Number of Proper prefixes are  $n - 1$ .

D Number of Subsequences are  $2^n$ .

Correct Option

YOUR ANSWER - NA

CORRECT ANSWER - b,d

STATUS - SKIPPED

Solution :

(b, d)

(a)  $n + 1$  (n prefixes for n characters, plus the empty string).

(b)  $n + 1$  (similar to (a)).

(c)  $n(n + 1)$  prefixes for n characters, minus the string itself).

(d)  $\sum_{i=0}^n {}^n C_i = 2^n$ .

Q. 17

? FAQ | Have any Doubt ?



Consider the following CFG:

 $S \rightarrow Aa|ca$  $A \rightarrow cd$ 

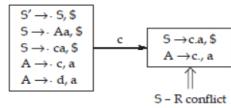
Which of the following statement is true?

- A Above grammar is not CLR(1).

Correct Option

Solution :

(a)

There is S-R conflict at  $I_1$ . Hence it can not be CLR(1).

Since it is not CLR(1), so it can not be LALR(1), SLR(1), LR(0) as well.

- B NOT CLR(1) but LALR(1).

- C SLR(1) but not LR(1).

- D SLR(1), LALR(1) but not CLR(1).

Q. 18

? FAQ | Have any Doubt ?



What kind of compilation error is this?

`fro(int I = 0; I < 5; I++);`

- A Syntax error

Correct Option

Solution :

(a)

Syntax error: Compiler assume fro as a function but as semicolon can not be used in a function (parameters) due to which compiler generates syntactical error.

- B Logical error

- C Semantic error

- D None of these

Q. 19

? FAQ | Have any Doubt ?



Which of the following is not a requirement for LL(1) grammar?

- A Unambiguity

- B No left recursion

- C If  $A \rightarrow \alpha|\beta$  are two productions, then FIRST( $\alpha$ ) and FIRST( $\beta$ ) are disjoint.

- D If  $A \rightarrow \alpha|\beta$  are two productions, then FIRST( $\alpha$ ) and FIRST( $\beta$ ) must have atleast one common term.

Correct Option

Solution :

(d)

LL(1) grammar are free from left recursion, ambiguity left factoring.  
So option (a), (b) and (c) are the pre requisite to be LL(1) grammar.

Q. 20

? FAQ | Have any Doubt ?



Which of the following statements regarding LR(0) parser is FALSE?

- A A LR(0) configurating set cannot have multiple reduce items.

**B** A LR(U) configurating set cannot have both shift as well as reduce items.

**C** If a reduce item is present in a LR(0) configurating set it may have other item.

**D** A LR(0) parser can parse any regular grammar.

Correct Option

Solution :

(d)

No grammar with empty production can be LR(0). But empty rules are allowed in regular grammar as the only condition for a grammar to be regular is to be either left linear or right linear.

 QUESTION ANALYTICS

+

Item 11-20 of 33 « previous 1 2 3 4 next »



Kunal Jha  
 Course: GATE  
 Computer Science Engineering(CS)

HOME  
 MY TEST  
 BOOKMARKS  
 MY PROFILE

**REPORTS**

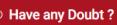
BUY PACKAGE  
 NEWS  
 TEST SCHEDULE

## COMPILER DESIGN (GATE - 2021) - REPORTS

OVERALL ANALYSIS    COMPARISON REPORT    **SOLUTION REPORT**

ALL(33)    CORRECT(0)    INCORRECT(0)    SKIPPED(33)

Q. 21

? FAQ    

Which of the following statements is FALSE?

**A** Any DCFL has an equivalent grammar that can be parsed by a SLR(1) parser with end string delimiter.

**B** Languages of grammars parsed by LR(2) parsers is a strict super set of the languages of grammars parsed by LR(1) parsers.

Correct Option

**Solution :**

(b)

$LR(0) \subset LR(1) = LR(2) \dots LR(k) = LR(k+1)$ ; for  $k \geq 1$ :  $LR(k) = LR(k+1)$  so, languages of grammars parsed by LR(2) parsers is not a strict super set of the languages of grammars parsed by LR(1) parsers although they both are same; if you compare their grammars:

$$LR(0) \subset LR(1) \subset LR(2) \dots LR(k) \subset LR(k+1)$$

**C** Languages of grammars parsed by LL(2) parsers is a strict super set of the languages of grammars parsed by LL(1) parsers.

**D** There is no DCFL which is not having a grammar that can be parsed by a LR(1) parser.



+

Q. 22

? FAQ    

If we merge states in LR(1) parser to form a LALR(1) parser, we may introduce

**A** Shift-reduce conflict

**B** Reduce-reduce conflict

Correct Option

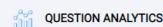
**Solution :**

(b)

It is because we construct LALR parsing table by merging states of CLR(1) which are only separated by look ahead heads. In doing so we may merge states which introduce R-R conflicts.

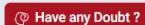
**C** No extra conflict

**D** Both shift-reduce as well as reduce-reduce



+

Q. 23



Consider the following Translation rules for the Grammar G:

$S \rightarrow a[\text{print "C"}] A$   
 $A \rightarrow b[\text{print "B"}] B$   
 $C \rightarrow c[\text{print "R"}]$   
 $A \rightarrow \epsilon[\text{print "B"}]$   
 $B \rightarrow e[\text{print "P"}] A$   
 $B \rightarrow \epsilon[\text{print "A"}]$

What will be the output for input string abebebe for bottom up parser?

**A** CBBPBPPB

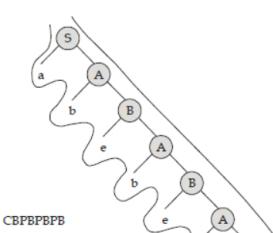
**B** CBBBPPBB

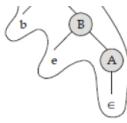
**C** CBPBPBPB

Correct Option

**Solution :**

(c)





**D** CBPBPPB

QUESTION ANALYTICS



Q. 24

? FAQ



Consider the following grammar:

$S \rightarrow XX$

$X \rightarrow b$

$X \rightarrow aX$

Which of the following can be the viable prefixes?

**A** baab

**B** aab

Correct Option

**Solution :**

(b)

- Viable prefix is nothing but stack content in LR Parsing. In this question just check option if it is visible in stack while doing parsing or not.
- A short trick to solve such question is to check, if there is already reduce handle in stack then stack can't store at any symbol after that reduce handle symbol.

Option (b) is only viable prefixes.

**C** aaabab

**D** bbbaX

QUESTION ANALYTICS



Q. 25

? FAQ



On translating the expression given below into quadruple representation, how many operations are required?

$$(i * j) + (e + f) * (a * b + c)$$

**6**

Correct Option

**Solution :**

6

$$\begin{aligned} T_1 &= (i * j) \\ T_2 &= (e + f) \\ T_3 &= (a * b) \\ T_4 &= (T_3 + c) \\ T_5 &= T_2 * T_4 \\ T_6 &= T_1 + T_5 \end{aligned}$$

Hence 6 operations are required.

QUESTION ANALYTICS



Q. 26

? FAQ



Construct the DAG for the expression given below and find out the number of leaf nodes in the graph:  
 $((x + y) - ((x + y) \times (x - y))) + ((x + y) \times (x - y))$

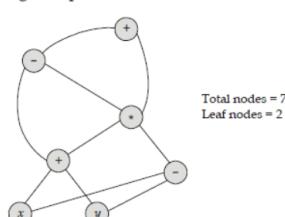
**2**

Correct Option

**Solution :**

2

Even if you don't draw the DAG, it's still ok and you can give the answer of this question.  
 Clearly, leaf nodes are operands thus 2 nodes ( $x, y$ ).  
 Below is the DAG for the give expression.



Q. 27

FAQ Have any Doubt ?



Number of temporary variable required to create 3 address code in static single assignment form for the expression  $P + Q * R - S / (Q * R)$ .

4

Correct Option

Solution :

4

$$P + Q * R - S / (Q * R)$$

$$\begin{aligned}T_1 &= Q * R \\T_2 &= P + T_1 \\T_3 &= S / T_1 \\T_4 &= T_2 - T_3\end{aligned}$$

So, total 8 variables [ $P, Q, R, S, T_1, T_2, T_3, T_4$ ].  
Here 4 temporary variables.

Q. 28

FAQ Have any Doubt ?



Consider the following CFG:

$$S \rightarrow aSbSb|ab$$

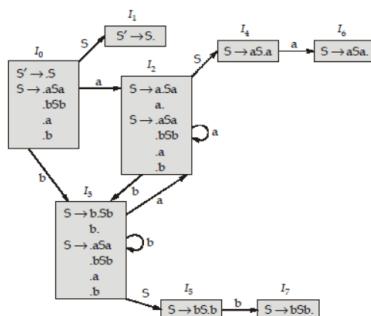
How many states having conflicts in LR(0) State Diagram? \_\_\_\_\_

2

Correct Option

Solution :

2



Total states in LR(0) diagram = 8.

Conflicting states having SR conflict = 2[I<sub>1</sub> and I<sub>3</sub>].

Q. 29

FAQ Have any Doubt ?



Consider the basic block given below:

$$u = u + v$$

$$v = v + w$$

$$x = v - w$$

$$y = v - x$$

$$z = u + v$$

The sum of minimum number of nodes and edges present in the DAG representations of the above basic block respectively are \_\_\_\_\_.

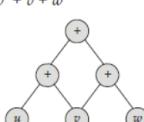
12

Correct Option

Solution :

12

$$\begin{aligned}z &= u + v \\z &= u + v + v + w\end{aligned}$$



Total number of nodes = 6

Total number of edges = 6

So, nodes + edges = 6 + 6 = 12

Let G be any grammar with the following productions:

$X \rightarrow X + Y \mid Y$

$Y \rightarrow Y * Z \mid Z$

$Z \rightarrow (X)$

$Z \rightarrow \text{id}$

If LR(1) parser is used to parse the above grammar, then total how many look-a-heads are present for the item  $X \rightarrow Y$  and  $Z \rightarrow \cdot \text{id}$  in the initial state \_\_\_\_\_.

5

Correct Option

Solution :

5

$I_0$  [Initial state]  
 $X' \rightarrow X, \$$   
 $X \rightarrow X + Y, \{ \$, + \}$   
   $.Y, \{ \$, + \}$   
 $Y \rightarrow Y * Z, \{ \$, *, * \}$   
   $.Z, \{ \$, *, * \}$   
 $Z \rightarrow (X), \{ \$, *, * \}$   
   $.id, \{ \$, *, * \}$

Look-ahead for  $X \rightarrow .Y, \{ \$, + \} = 2$   
Look-ahead for  $X \rightarrow .id, \{ \$, *, * \} = 3$   
Total look-ahead =  $2 + 3 = 5$

QUESTION ANALYTICS

+



Kunal Jha  
 Course: GATE  
 Computer Science Engineering(CS)

HOME

MY TEST

BOOKMARKS

MY PROFILE

REPORTS

BUY PACKAGE

NEWS

TEST SCHEDULE

## COMPILER DESIGN (GATE - 2021) - REPORTS

OVERALL ANALYSIS

COMPARISON REPORT

SOLUTION REPORT

ALL(33)

CORRECT(0)

INCORRECT(0)

SKIPPED(33)

Q. 31

? FAQ

Have any Doubt ?



Choose the correct statement(s) from the following:

$S \rightarrow AB$   
 $A \rightarrow Ca | \epsilon$   
 $B \rightarrow BaAC | c$   
 $C \rightarrow b | \epsilon$

A First (A) = {b, a, ε}

Correct Option

B Follow (B) = {S, a}

Correct Option

C Follow (A) = {c, b, S, a}

Correct Option

D First (B) = {b, a, ε}

YOUR ANSWER - NA

CORRECT ANSWER - a,b,c

STATUS - SKIPPED

**Solution :**

(a, b, c)

First remove left recursion from the grammar:

$S \rightarrow AB$   
 $A \rightarrow Ca | \epsilon$   
 $B \rightarrow aACB' | \epsilon$   
 $C \rightarrow b | \epsilon$

First (S) = {b, c, a}  
 First (A) = {b, a, ε}  
 First (B) = {c}  
 First (C) = {b, ε}  
 Follow (S) = {S}  
 Follow (A) = {c, b, S, a}  
 Follow (B) = {S, a} [a because first of B' is subset of follow of B]  
 Follow (C) = {S, a}

Yes all are correct.

QUESTION ANALYTICS



Q. 32

Have any Doubt ?



Which of the following is (are) machine independent optimization?

A Loop optimization

Correct Option

B Redundancy elimination

Correct Option

C Folding

Correct Option

D Register allocation

YOUR ANSWER - NA

CORRECT ANSWER - a,b,c

STATUS - SKIPPED

**Solution :**

(a, b, c)

Machine Independent optimization: Loop optimization, Redundancy elimination, folding, Strength reduction, Deadlock elimination.

Machine Dependent optimization: Register allocation, Use of addressing modes, Peephole optimization.

QUESTION ANALYTICS



Q. 33

? FAQ

Have any Doubt ?



Given the grammar:

$S \rightarrow T * SIT$   
 $T \rightarrow U + TIU$   
 $U \rightarrow b$

Which of the following statements is/are true?

A Grammar is not ambiguous.

Correct Option

B Priority of + over \* is ensured.

Correct Option

**C** Right to left evaluation of \* and + happens.

Correct Option

**D** Priority of \* over + is ensured.

YOUR ANSWER - NA

CORRECT ANSWER - a,b,c

STATUS - SKIPPED

**Solution :**

- (a, b, c)
- (a) Grammar is not ambiguous - true
- (b) Priority of + is > than \*, since + comes lower in parse tree - true.
- (c) Right associativity - true.

 QUESTION ANALYTICS

+

Item 31-33 of 33 « previous 1 2 3 4 next »