COMPILER DESIGN
FULL LENGTH TEST-1

Question-1

Which of the following is a part of a compiler that takes as input a stream of characters and produces as output a stream of words along with their associated syntactic categories?

a) Optimizer

b) Scanner

c) Parser

d) None of the mentioned

Question-2

Which of the following statement is/are true?

S1 : First statement of three address code is leader.

S2 : Every target of goto statement is leader.

S3 Every statement after goto is leader.

(a) S1 only

(b) S2 only

(c) S1 and S3 only

(d) S1 ,S2 and S3 only

Question-3

Consider the following C-program:

```
int strange (int x)
if (x <= 0) return 0;
if (x%2! = 0) return -1;
 return 1 + strange (x-1);
}
```

Find out the number of tokens?

(a) 44

(b) 42

(c) 43

(d) 75

Question-4

Which of the following is correct regarding an optimizer Compiler?

a) Optimize the code

b) Is optimized to occupy less space

c) Both of the mentioned

d) None of the mentioned

Question-5

An object module for a group of programs that were compiled separately is handed to a linker. Which of the following about an object module isn’t true?

a) Relocation bits

b) Names and locations of all external symbols denied in the object module

c) Absolute addresses of internal symbols

d) Object code

Question-6

Why Generation of intermediate code based on an abstract machine model is useful in compilers?

a) Writing for intermediate code generation

b) Portability of the front end of the compiler

c) Implementation of lexical analysis and syntax analysis is made easier

d) All of the mentioned

Question-7

Which of the following is correct?

(a) The drawback for quarduple representation of 3-address code is it requires more space compared to triple notation.

(b) The advantage for triple representation of 3- address code is it requires less space compared to quarduple notation.

(c) Both (a) and (b) are correct.

(d) Neither (a) nor (b) are correct

Download ExamPrepTool (EPT App)

https://play.google.com/store/apps/details?id=com.ept.requestteacher

For PYQ's with video solutions, Topper Notes, short notes, Toppers Test Series

Note: If any doubt in any question post with subject & question number in EPT App.

Question-8

Let G be a grammar and G has following set of productions.

S⟶ AA

A->AA |bA |Ab|a

How many strings can be generated by grammar

G with atmost four steps in the derivation. [initially start symbol is available without any step]

(a) 3

(b) 4

(c) 5

(d) 6

Question-9

Consider the following C-prog. Find the number of tokens in the output of the following program if that is passed as a string to a lexical analyzer.

```
main()
{
chars[] = {"ice", "green", "water", "hi");
char ** ptr[]= {s +3, S +2, S + 1, s}
 char ***p = ptr;
printf("in, %s", **++ P);
}
```

(a) 5

(b) 1

(c) 4

(d) 3

Question-10

Find the line number in which lexical error present in the following program.

```
1. main()
2. {
3. int x; y, z,
4. /* comment,*/
5. /*com/*ment2*/end*/
6.x = "A";
7. }
```

(a) 3

(b) 5

(c) 6

(d) No lexical error

Question-11

Consider the following grammar G:

S⟶V=E

E⟶FE+F F⟶V | int |(E)|e

V⟶ id

Which of the following will represents the FIRST (F) and FOLLOW (F) respectively?

(a) FIRST (F) = (id, int, c, e}
FOLLOW (F) = {+,), $}

(b) FIRST (F) = (id, int, c}
FOLLOW (F) = {+,), $)

(c) FIRST (F) (id, int, c, e}
FOLLOW (F) {+,)}

(d) None of these

Question-12

A system program that brings together separately compiled modules of a program into a form language that is suitable for execution.

a) Assembler

b) Linking loader

c) Cross compiler

d) None of the mentioned

Question-13

stmt ->if expr then stmt
|if expr then stmt else stmt
|other

Consider the following statements for above grammar:

1. The grammar will cause shift reduce conflict.
2. The grammar is ambiguous.
3. The grammar is left recursive.

Which of the above statements is/are true?

(a) 1 and 2 is true

(b) 1, 2 and 3 is true

(c) 2 and 3 is true

(d) all are false

Question-14

Consider the production of the grammar S->AA A->aa A->bb Describe the language specified by the production grammar.

a) L = {aaaa,aabb,bbaa,bbbb}

b) L = {abab,abaa,aaab,baaa}

c) L = {aaab,baba,bbaa,bbbb}

d) L = {aaaa,abab,bbaa,aaab}

Question-15

The string WWR is not recognized by any FSM because

____

a) An FSM cannot remember arbitrarily large amount of information
b) An FSM cannot fix the midpoint
c) An FSM cannot match W with WR
d) An FSM cannot remember first and last inputs

Question-16

Consider the grammar G shown below:

G: $E \longrightarrow EzE \mid xyE \mid w$

Which of the following conflict is present in the DFA construction of SLR(1) parser for grammar G.

(a) Reduce-Reduce conflict

(b) Shift-Reduce conflict

(c) Both (a) and (b)

(d) No conflict present

Question-17

Consider the following grammar

S⟶(L)|a

L->L, S|S

The maximum size of stack during LL(1) parsing the input string w = (a, a) is _______________

Question-18

The lexical analysis for a modern language such as Java needs the power of which one of the following machine models in a necessary and sufficient sense?

a) Finite state automata

b) Deterministic pushdown automata

c) Non-deterministic pushdown automata

d) Turing machine

Question-19

Given the language L = {ab, aa, baa}, whih of the following strings are in L*?

i) abaabaaabaa

ii) aaaabaaaa

iii) baaaaabaaaab

iv) baaaabaa

a) i, ii and iii
b) ii, iii and iv
c) i, ii and iv
d) i, iii and iv

Download ExamPrepTool (EPT App)

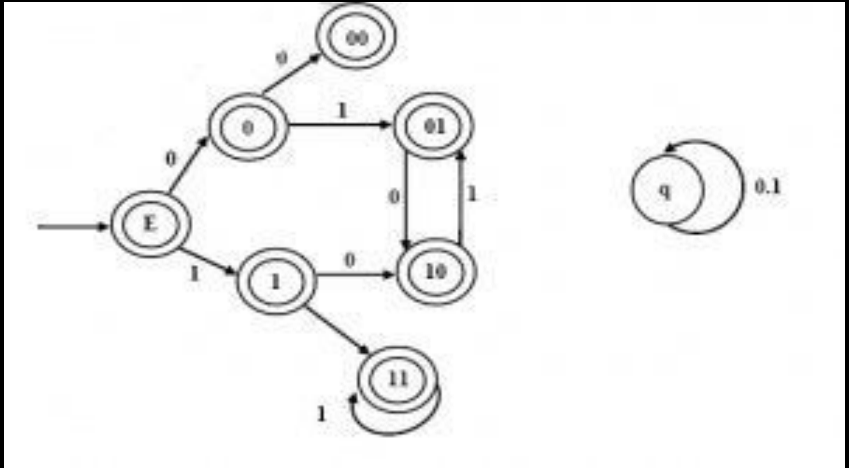
https://play.google.com/store/apps/details?id=com.ept.requestteacher

For PYQ's with video solutions, Topper Notes, short notes, Toppers Test Series

Note: If any doubt in any question post with subject & question number in EPT App.

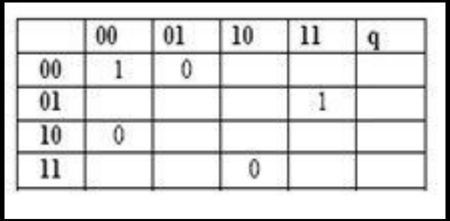
Question-20

Consider the set of strings on {0,1} in which, every substring of 3 symbols has at most two zeros. Complete the partially completed DFA that accepts this language is shown below.



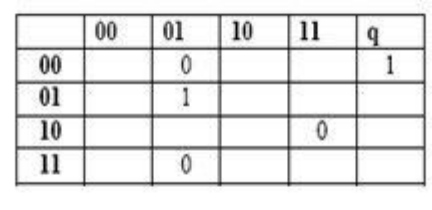
The missing arcs in the DFA are

a)



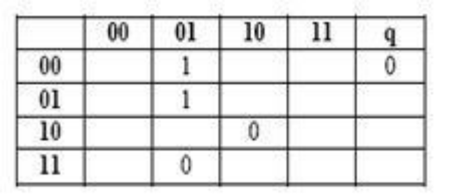
	00	01	10	11	q
00	1	0			
01				1	
10	0				
11			0		

b)



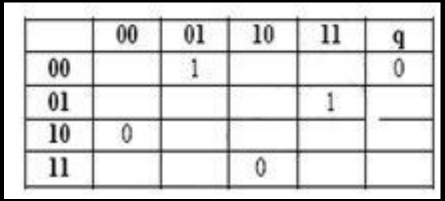
	00	01	10	11	q
00		0			1
01		1			
10				0	
11		0			

c)



	00	01	10	11	q
00		1			0
01		1			
10			0		
11		0			

d)



	00	01	10	11	q
00		1			0
01				1	
10	0				
11			0		

Question-21

Which of the following problems occur?

i) Does a given program ever produce an output?

ii) If L is a CFL, then is L’ is also context-free?

iii) L’ is regular only if L is regular?

iv) If L is a recursive language, then, L’ is also recursive?

a) i, ii, iii, iv

b) i, ii

c) ii, iii, iv

d) iii, iv

Question-22

Consider the grammar:

S⟶ aABe

B⟶d

A⟶ Abclb For which the string abbcde is right most derivation

(a) S->aABe=>aAde⇒ aAbcde=> abbcde

(b) S->aABe=>aAbcBe⇒ abbcBe⇒ abbcde

(c) Both (a) and (b)

(d) None of these

Question-23

Match the following.

Group 1	Group 2
P. Regular expression	1. Syntax analysis
Q. Pushdown automata	2. Code generation
R. Dataflow analysis	3. Lexical analysis
S. Register allocation	4. Code optimization

a) P-4. Q-1, R-2, S-3

b) P-3, Q-1, R-4, S-2

c) P-3, Q-4, R-1, S-2

d) P-2, Q-1, R-4, S-3

Question-24

Let L = L1 ∩ L2, where L1 and L2 are languages as defined below.

L1 = {ambmcanbn | m, n >= 0 }

L2 = {aibjck | i, j, k >= 0 } Then L is?

a) Not recursive

b) Regular

c) Context free but not regular

d) None of the mentioned

Question-25

Consider the following statements:

S_1: Viable prefixes of grammar G are those

prefixes of right-sentenial form that can appear on top of stack of a shift reduce parsing.

S_2: Handle is sub-string that matches right side of production whose reduction to non terminal on left side of production represents one step along the reverse of leftmost derivation.

Which of the above statements is/are true?

(a) S_1, is true and S_2 is false

(b) S_2, is true and S_1, is false

(c) Both are true

(d) Both are false

Question-26

Which of the following is a correct statement?

a) { If an bn | n = 0,1, 2, 3 ..} is regular language

b) Strings with equal number of a’s and b’s denies a regular language

c) L (A* B*)∩ B gives the set A

d) None of the mentioned

Question-27

Consider the following grammar:

E-> E / X |X

X -> T - X|X*T| T

T ->T+F|F

F-> (E)| id

The above grammar is used to generate all valid arithmatic expressions in a hypothetical language in which

(a) / associates from the left

(b) * associates from the left

(c) + associates from the left

(d) all of the above

Question-28

Here is a context-free grammar G: S ⟶ AB A ⟶ 0A1 | 2 B ⟶ 1B | 3A which of the following strings are in L (G)?

a) 021300211

b) 022111300211

c) None of the mentioned

d) 021300211 & 022111300211

Question-29

For the following grammar: S → A | B | 2 A → C0 | D B → C1 | E C → D | E | 3 D → E0 | S E → D1 | S Identify all the unit pairs.

a) D,C

b) A,B

c) B,C

d) A,C

Question-30

Consider the operator precedence relation

	id	+	*	$
Id		. >	.>	. >
+	< .	. >	.>	.>
*	< .	< .	.>	.>
$	< .	< .	< .	

Suppose we are evaluating string id1+ id2*id3 give the order in which id1,id2 and id3 will be evaluated

(a) id1 ,id 3 ,id2

(b) id2,id1 ,id3

(c) id1 , id2, id3

(d) id3, id2,id1

Question-31

In operator precedence parsing, for the input id *(id - id) - id / id, where id is an identifier (terminal), the following grammar is given.

$E \longrightarrow EAE \mid (E) \mid - E \mid id$

$A \longrightarrow + \mid - \mid / \mid \uparrow$

Which symbol has the higher precedence than other?

(a) $, +

(b) id, *

(c) +, *

(d) -, *

Question-32

Let L1 = {w ∈ {0,1}∗ | w has at least as many occurrences of (110)’s as (011)’s}.
Let L2 = { ∈ {0,1}∗ | w has at least as many occurrences of (000)’s as (111)’s}.
Which one of the following is TRUE?
a) L1 is regular but not L2
b) L2 is regular
c) L1 and L2 are regular
d) Neither of them are regular

Question-33

Consider alphabet ∑ = {0, 1}, the null/empty string λ and the sets of strings X0, X1 and X0. How are X1 and X2 are related?

X0 = 1 X1 X1 = 0 X1 + 1 X2 X2 = 0 X1 + {λ} Which one of the following represents the strings in X0?

a) 10 (0* + (10)*)1
b) 10 (0* + (10)*)*1
c) 10 (0* + (10)*)*1
d) 10 (0 + 10)*1 + 110 (0 + 10)*1

Question-34

S -> AB

S -> CA

B -> BC

B -> AB

A -> a

C aB |b

Check which of the following is correct for above grammar?

(a) the above grammar is in reduce form

(b) the above grammar is not in reduce form

(c) from given non-terminal we get the non- terminal only thus above grammer is in reduce form

(d) none of the above

Question-35

Consider the LR(0), LR(K), LR(1) and SLR(1) grammars then which of the following about these grammar is true?

(a) LR(0) ⊂SLR(1)⊂ CLR(1) ⊂ CLR(K)

(b) LR(0) ⊂ CLR(k) ⊂ LR(1) ⊂ SLR(1)

(c) SLR(1) ⊂ CLR(0) ⊂ LR(1) ⊂ LR(K)

(d) LR(K) ⊂ CLR(0) ⊂ CLR(1) ⊂ SLR(1)

Question-36

Consider the grammar given below E? E+E | E*E | E-E | E/E | E^E | (E) | id Assume that + and ^ have the same but least precedence, * and / have the next higher precedence but the same precedence and finally ^ has the highest precedence. Assume + and ^ associate to the left like * and / and that ^ associates to the right. Choose the correct for the ordered pairs (^,^), (-,-), (+,+), (*,*) in the operator precedence table constructed for the grammar.

a) All <

b) All >

c) < >, =

d) < > > >

Question-37

A context free language is called ambiguous if __________

a) It has 2 or more left derivations for some terminal string w є L (G)

b) It has 2 or more right derivations for some terminal string w є L (G)

c) It has 2 or more left & right derivations for some terminal string w є L (G)

d) None of the mentioned

Question-38

Fill in the blanks in the following statements:

S_1: Merging states with a common core may produce __I______conflicts but does not produce ___II_______conflicts in LALR parser.

S_2: The LR parsing method is the most general ___III______ ___IV_____ parsing method.

(a) I-reduce-reduce; II-shift reduce III non backtracking IV-predictive

(b) I-shift-reduce; II-reduce-reduce; III back tracking IV-shift-reduce

(c) I-shift-reduce; II-reduce-reduce; III back tracking; IV-predictive

(d) I-reduce-reduce; II-shift-reduce; III-non back tracking; IV shift-reduce

Question-39

Consider a program P that consists of two source modules M1(contains reference to a function defined in M2) and M2 contained in two different files.

a) Edit time

b) Compile time

c) Link time

d) Load time

Question-40

Which of the following describes a handle (as applicable to LR-parsing) appropriately?

a) Position where next reduce or shift operation will occur

b) The next step has use of Non-terminal for reduction

c) Used for reduction in a coming-up step along with a position in the sentential form where the next shift or reduce operation will occur

d) Used in the next step for reduction along with a position in the sentential form where the right hand side of the production may be found

Question-41

How many conflicts are there in the following grammar

$$S \longrightarrow SS \mid a \mid \in$$

(a) 3

(b) 4

(c) 5

(d) 6

Question-42

What is terminal table?

a) Contains all constants in the program

b) Is a permanent table of decision rules in the form of patterns for matching with the uniform symbol table to discover syntactic structure

c) Consist of a full or partial list of the token is as they appear in the program created by lexical analysis and used for syntax analysis and interpretation

d) Is a permanent table which lists all keywords and special symbols of the language in symbolic form

Question-43

Advantage of incorporating the macro-processor into pass 1 is that

____

a) Many functions have to be implemented twice
b) Functions are combined not necessarily creating intermediate files as output from the macro-processor and input to the assembler
c) More flexibility is provided to the programmer in that he may use all the features of the assembler in conjunction with macros
d) All of the mentioned

Question-44

A linker is given object module for a set of programs that were compiled separately. What is not true about an object module?

a) Object code

b) Relocation bits

c) Names and locations of all external symbols denied in the object module

d) Absolute addresses of internal symbols

Question-45

Consider the following translation scheme.

S > ER

R > * E{print{’ * ’);

R | f

E > F + E{print(’ + ’); | F F > (S) | id{print(id.value);}

Here id is a taken that represents an integer and id. value represents the corresponding integer value. For an input ‘2 * 3 + 4’, this translation scheme prints?

a) 2 * 3 + 4
b) 2 * + 3 4
c) 2 3 * 4 +
d) 2 3 4 + *

Download ExamPrepTool (EPT App)

https://play.google.com/store/apps/details?id=com.ept.requestteacher

For PYQ's with video solutions, Topper Notes, short notes, Toppers Test Series

Note: If any doubt in any question post with subject & question number in EPT App.

Question-46

Consider the following grammar:

S > FR

R > * S | ε

F > id

In the predictive parser table, M , of the grammar the entries M [S, id] and M [R,$] respectively.

a) {S ” FR} and {R ” ε}
b) {S ” FR} and {}
c) {S ” FR} and {R ” * S}
d) {F ” id} and {R ” ε}

Question-47

Which of the following describes a handle (as applicable to LR-parsing) appropriately?

a) It is the position in a sentential form where the next shift or reduce operation will occur

b) It is a non-terminal whose production will be used for reduction in the next step

c) It is a production that may be used for reduction in a future step along with a position in the sentential form where the next shift or reduce operation will occur.

d) It is the production p that will be used for reduction in the next step along with a position in the sentential form where the right-hand side of the production may be found

Question-48

Some code optimizations are carried out on the intermediate code because _________________

a) They enhance the portability of the complier to other target processors

b) Program analysis is name accurate on intermediate code than on machine code

c) The information from data flow analysis cannot otherwise be used for optimization

d) The information from the front end cannot otherwise be used for optimization

Question-49

For the above CFG, LL(1) parsing table constructed using production numbers are as follows:

	c	s	e	$
S		1	1	
X	4	E1	E2	4
Y			5	

Find the missing entries E_{1} and E_{2} in the above LL(1) table.

(a) E1 = {2} ,E2 = {3, 4}

(b) E1 = {3, 4}, E2 = {2}

(c) E1 = {2, 4} ,E2 = {3, 4}

(d) E1 = {3, 4} , E2 = {2, 4}

(v) Languages which permit a function to return a function as its result cannot be implemented with a stack-based storage allocation scheme for activation records

a) (ii) and (v) only
b) (i), (iii) and (iv) only
c) (i), (ii) and (v)
d) (ii), (iii) and (v) only

Question-50

Let G be any arbitary grammar and G known to be LR(1). Assume that the number of error entries in the LR(0) table is 'a' and the number of error entries in the SLR(1) table is 'b'. What is the relation over a and b? [Assume an error is SR conflict or RR conflict]

(a) a>b>0

(b) a>=b>=0

(c) a<=b<=0

(d) a <b<0

Question-51

In a two pass assembler, adding literals to literal table and address resolution of local symbols are done using?

a) First pass and second respectively

b) Both second pass

c) Second pass and first respectively

d) Both first pass

Question-52

Consider the following statements.

I. Symbol table is accessed only during lexical analysis and syntax analysis.

II. Compilers for programming languages that support recursion necessarily need heap storage for memory allocation in the run-time environment.

III. Errors violating the condition 'any variable must be declared before its use' are detected during syntax analysis.

Which of the above statements is/are TRUE?

a) I only

b) I and III only

c) II only

d) None of I, II, and III

Question-53

Consider the following syntax directed definition

A⟶BC

C.i= c(A.i)…..1

B.i=b(A.s) ... 2

A.s = f(B.s)….3

i for inherited, s for synthesized attribute

Which of the following is true?

(a) 1 is violating L attributed definition

(b) 2 is violating L attributed definition

(c) 3 is violating L attributed definition

(d) None of the above

Question-54

A shift-reduce parser carries out the actions specified within braces immediately after reducing with the corresponding rule of grammar

S⟶AS (print "1")

S⟶ AB (print "2"}

A->a (print "3"}

B⟶ bC (print "4"}

B⟶ dB (print "5"}

C⟶c{print "6"}

This syntax directed translation scheme translates a language whose terminal symbols are a, b, c, and d into another language whose terminal symbols are 1, 23, 4, 5 and 6. What is the translation of "aaadbc"?

(a) 333546 (b) 654211

(c) 333645211 (d) 64523331

Question-55

A lexical analyzer uses the following patterns to recognize three tokens T1, T2, and T3 over the alphabet (a,b,c}.

T_1: a? (b|c)*a

T2: b? (ac)*b

T3: c? (b|a) *c

Note that 'x?' means 0 or 1 occurrence of the symbol x. Note also that the analyzer outputs the token that matches the longest possible prefix. If the string bbaacabc is processed by the analyzer, which one of the following is the sequence of tokens it outputs?

a) T_1T2T3

b) T_1T1T3

c) T_2T1T3

d) T3T3

Question-56

Consider the following grammar

S→m | mn | mno

Choose correct statement from the following:

1. The grammar is LL (2)
2. The grammar is LL (4)
3. The grammar is LL (1)
4. The grammar is LL (3)

Question-57

Given the grammar

s⟶T*SIT

T⟶ U + T | U

U ⟶ a|b

Which of the following statements is wrong?

a) Grammar is not ambiguous

b) Priority of + over * is ensured

c) Right to left evaluation of* and + happens

d) None of these

Question-58

A shift reduce parser carries out the actions specified with in braces immediately after reducing the corresponding rule of grammar.

A->bbB (print "+"}

A⟶ a {print "x"}

B⟶ Ac {print "-"}

What is the translation of bbbbacc using the syntax directed translation scheme described by the above rules?

(a) * + - + *

(b) * * - + -

(c) + * - + *

(d) * - + - +

Question-59

Which one of the following is a top-down parser?

a) Recursive descent parser

b) Operator precedence parser

c) An LR(k) parser

d) An LALR(k) parse

Question-60

Let G be a grammar is used for arithmetic expressions. The grammar G is shown below with semantic actions and attribute "sign" can contain either 0 or 1.

E->E1+E2 (E.sign = E2 sign)

E->E1 x (E2) (Esign =E1. sign x E2.sign)

E->E1/E2, (if (E1.sign == 0) then

Esign= 1 else Esign = 0)

E->+E1 (E.sign = 0)

E->-E1 (Esign = 1}

E->id (E.sign=0)

[Note: E->E1+E2 is same as E⟶ E+E]

Find the attribute value at the root E for the given input: -id x (-id + id)

(a) 0

(b) 1

(c) 2

(d) None of these

Question-61

Consider the following production, along with the syntax directed definition.

A->QR,

Q.in=f.(As) || 1st rule

R.in=f.(Q.s) || 2nd rule

Where .in and .s have their regular meaning i.e., inherited and synthesized attributes respectively. Which of the following option is true?

(a) Syntax directed definition are L-attributed.

(b) Syntax directed definition is not L-attributed because of 2nd rule.

(c) Syntax directed definition is not L-attributed because of 1st rule

(d) Syntax directed definition is not L-attributed because of both rule

Download ExamPrepTool (EPT App)

https://play.google.com/store/apps/details?id=com.ept.requestteacher

For PYQ's with video solutions, Topper Notes, short notes, Toppers Test Series

Note: If any doubt in any question post with subject & question number in EPT App.

Question-62

Consider the basic block given below:

b = (b+c)

d = b*d

b = d-b

The minimum number of nodes and edges present in the DAG representation of above basic

block respectively are

(a) 4 and 5

(b) 5 and 4

(c) 6 and 6

(d) 6 and 7

Question-63

It has encoded within it information on the possible sequences of characters that can be contained within any of the tokens it handles. The mentioned function is performed by?

a) Scanner

b) Parser

c) Syntactic Analyser

d) All of the mentioned

Question-64

Which of the following statements is false?

a) Left as well as right most derivations can be in Unambiguous grammar

b) An LL (1) parser is a top-down parser

c) LALR is more powerful than SLR

d) Ambiguous grammar can’t be LR (k)

Question-65

Consider the following statements:

Statement I: LALR parser is more powerful than canonical LR Parser.

Statement II: SLR parser is more powerful than LALR

Which of the following is correct?

a) Statement I true and Statement II false

b) Statement I false and Statement II true

c) Both Statement I and Statement II false

d) Both Statement I and Statement II true

SOLUTION

Solution-1

Answer: b

Explanation: A compiler’s scanner scans a character-based input stream and creates a word-based output stream, with each word identified with its Syntactic category.

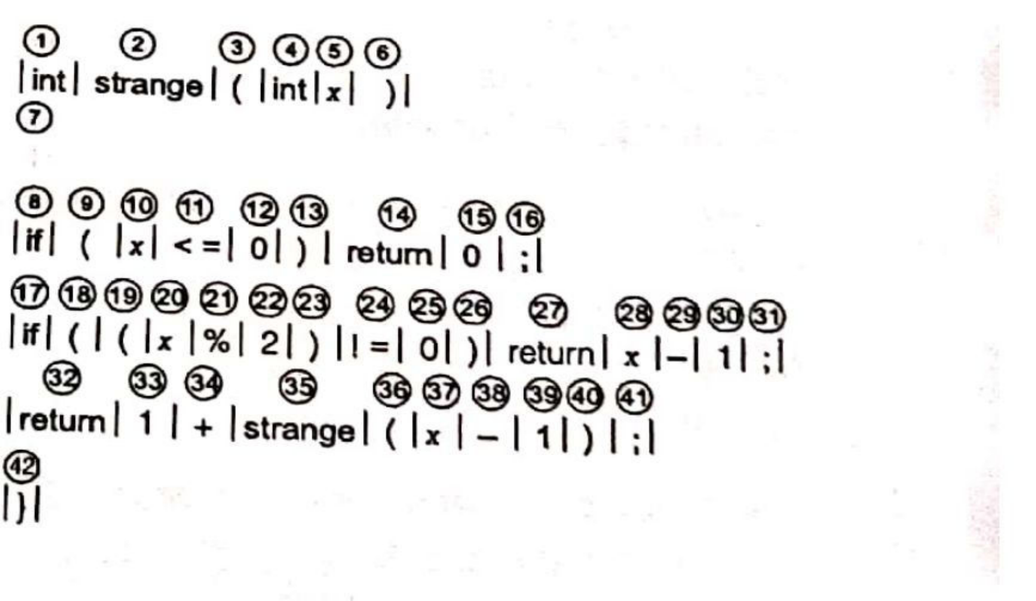
Solution-2

Answer: (a)

The all of three-statement is true about the leader of basic block in flow graph of code optimization technique

Solution-3

Answer b



```
 ①     ②      ③ ④⑤ ⑥
|int| strange| ( |int|x|  )|
⑦

⑧ ⑨ ⑩ ⑪ ⑫⑬    ⑭   ⑮⑯
|if| ( |x| <=| 0| ) | return| 0 | ;|
⑰⑱⑲⑳㉑㉒㉓ ㉔㉕㉖   ㉗     ㉘㉙㉚㉛
|if| ( | ( |x |%| 2| ) |!=| 0| )| return| x |–| 1| ;|
  ㉜   ㉝ ㉞    ㉟     ㊱㊲㊳㊴㊵㊶
|return| 1 | + |strange| ( |x | – | 1| ) | ;|
㊷
|}|
```

Solution-4

Answer: d

Explanation: An optimising compiler is a computer programme that strives to minimise or maximise specific characteristics of an executable programme.

Solution-5

Answer: c

Explanation: A linker, sometimes known as a link editor, is a computer program that merges one or more object files generated by a compiler into a single executable, library, or another object file

Solution-6

Answer: c

Explanation: Intermediate code generator receives input from its predecessor phase, semantic analyzer, in the form of an annotated syntax tree.

Solution-7

Answer:(c)

The drawback in quarduple representation is one extra field required to store the result. In triple representation there is no need of extra field to store the result, so it requires less space. Both (a) and (b) are correct.

Solution-8

Answer: (b)

S -> AA -> aA -> aa

S -> AA -> aA -> abA -> aba

S -> AA -> aA -> aAb -> aab

S -> AA -> Aa -> bAa -> baa

.. {aa, aba, aab, baa} can be generated within 4 steps.

Solution-9

Answer: b

We need not to find out the actual output of the given C-program. Just by seeing the C-code we can understand that the output is a string and that is passed to lexical analyzer and that would be treated as a single valid lexeme for lexical analyzer. For instance we can see the output of the program will be 'water'.

|Water| is considered as a single token

Solution-10

Answer: d

The given program contain no lexical error even though it contains syntax errors. In line number "5", comment started and searches for the first close comment pattern when it finds, it consider a comment. There is no start comment pattern (*) but there is end comment at last in line 5, hence it is not lexical error but it is syntax error.

/*com/ment2 */ end */

Comment Identifier Operator⇒ No lexical error

Solution-11

Answer: a

FOLLOW (F) = Contain terminal after F on RHS of grammar, if nothing present right hand side of 'F' then find FOLLOW (LHS) i.e. FOLLOW (E) = {+,)) and again FOLLOW (S) = {$}

So FOLLOW (F) = {+,), & }

FIRST (F)= Contain first terminal on RHS i.e. (id, int, c, ∈}

Solution-12

Answer: b

Explanation: A loader which brings together the functions of a relocating loader with the ability to combine a number of program segments that have been independently compiled into an executable program.

Solution-13

Answer: (a)

The grammar

```
stmt ->if expr then stmt
          |if expr then stmt else stmt
          |other
```

When we construct parsing table it will cause shift reduce conflict because at else we will not be able to decide whether shift or reduce and also the given grammar is ambiguous.

Solution-14

Answer: a

Explanation: S->AA (substitute A->aa)

S->aaaa

S->AA (substitute A->aa)

S->aaA (substitute A->bb)

S->aabb

S->AA (substitute A->bb the A->aa)

S->bbaa

S->AA (substitute A->bb)

S->bbbb.

Solution-15

Answer: b

Explanation: Palindromes cannot be recognized by FSM.

Solution-16

Answer: (b)

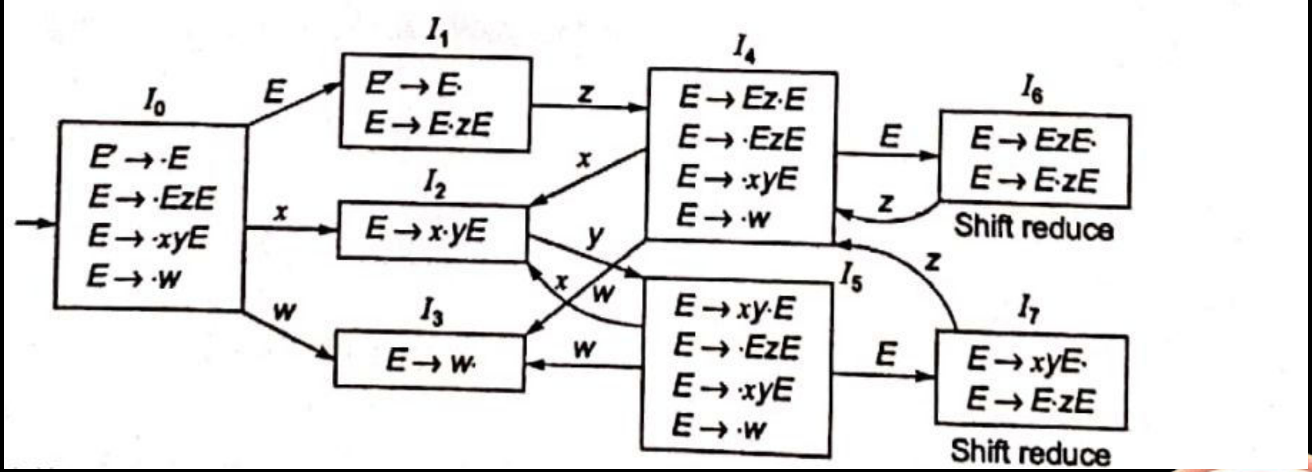
Augmented grammar G:

E->.E

E->.EzE

E->.xyE

E->.w



In state, I_6 and I_7, shift reduce conflict is present. The element to be shifted is reduced so they will come in same column.

Solution-17

Answer: (4)

Since given grammar is left recursive, so first convert into non-left recursive.

S->(L)|a

L->SL'

L '-> S L' | ∈

First (S)={(. a}

Follow (S) ={, ,), $)

First (L) = {(. a}

Follow (L) = {)}

First (L'),={, , follow (L')}

= {L , ,)}

	(	)	a	,	$
S	S->(L)		S->a		
L	S->SL’		L->SL’		
L’		L->∈		L’->SL’	

LL(1) Table

Maximum size of stack is 4.

Stack	Input	Output
$S	(a,a)$	
$)L(	(a,a)$	S->(L)
$)L’	a,a)$	
$)L’S	a,a)$	S->SL’
$)L’a	a,a)$	S->a
$)L’	,a)$	
$)L’S	,a)$	L’->,SL’
$)L’S	a)$	
$)L’a	a)$	S->a
$)L’	)$	
$)	)$	S->∈
$)	$	

Solution-18

Answer: a

Explanation: Initially in lexical analysis the program is divided into tokens. Tokens can be expressed as regular expressions: [a-zA-Z] [a-zA-Z0-9]*

the keyword if is given by if.

Integers are given by [+-]? [0-9]+.

Solution-19

Answer: c

Explanation: Any combination of strings in set {ab, aa, baa} will be in L*.

i) “abaabaaabaa” can be partitioned as a combination of strings in set {ab, aa, baa}. The partitions are “ab aa baa ab aa”

ii) “aaaabaaaa” can be partitioned as a combination of strings in set {ab, aa, baa}. The partitions are “aa ab aa aa”

iii) “baaaaabaaaab” cannot be partitioned as a combination of strings in set {ab, aa, baa}

iv) “baaaabaa” can be partitioned as a combination of strings in set {ab, aa, baa}. The partitions are “baa aa ab aa”.

Solution-20

Answer: d

Explanation: State ‘q’ is trap state. All other states are accepting states. In state 00, DFA must move to ‘q’ for input symbol 0. All (non-trap) states indicate names indicate the characters seen before reaching that particular state.

Solution-21

Answer: d

Explanation: i) Is a variation of Turing Machine Halting problem and it is undecidable.

….ii) Context Free Languages are not closed under intersection and complement.

….iii) Complement of Regular languages is also regular.

….iv) Recursive Languages is closed under complement.

Solution-22

Answer: (a)

The string abbcde can be obtained using RMD by following steps

S->aABe->aAde→aAbcde→abbcde.

Solution-23

Answer: b

Explanation: Regular grammar relates to lexical analysis

Pushdown automata relates to Syntax analysis

Data flow analysis is Code optimization

Register allocation is code generation.

Solution-24

Answer: c

Explanation: The language L1 accept strings {c, abc, abcab, aabbcab, aabbcaabb,} and L2 accept strings {a, b, c, ab, abc, aabc, aabbc, … }. Intersection of these two languages is L1 ∩ L2 = {akbkc | k >= 0} which is not regular but context free.

Solution-25

Answer: (a)

"Handle" is a substring that matches the right side of a production, and whose reduction to the nonterminal on the left side of the production represents one step along the reverse of a right most derivation.

Solution-26

Answer: c

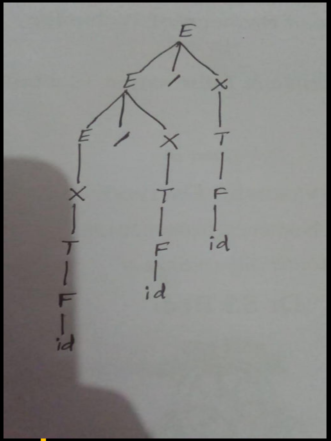
Explanation: If we include A and B in a set and if we write A* it means except then A i.e. B same as B* means except then B i.e. so if we intersect (A*B*) and B then get A because in any regular language. If we write A-B then A-B=A intersection B’ so if we intersect A and B means A-B So the intersection of (A*B*) and B = (BA). intersection B means (BA)-B’ and B’=A so (BA) intersection(A)=A

Solution-27

Answer: (d)

For checking associativity of /, consider the string

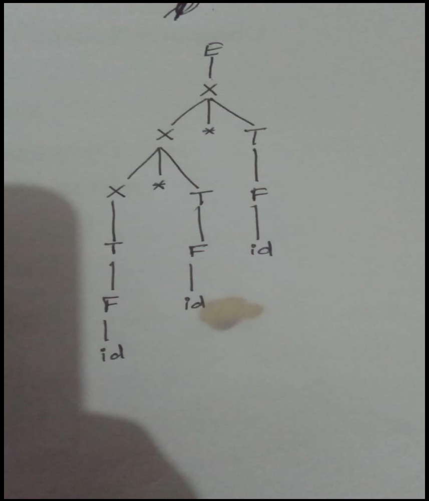
id/id/id



The parse tree grows down towards the left, hence is left associative.

For checking associativity of

*, consider the string id id id



The parse tree grows down towards the left, hence *is left associative.

For checking associativity of +, consider the string id+id+id

Solution-28

Answer: d

Explanation: First, notice that A generates strings of the form 021, where n is 0 or more. Also, B gives zero or more 1’s, which is followed by one 3, and then A gives something. Since S generates something an A can generate followed by something a B can generate, the strings in L (G) are of the form 0 21 30 21.

Solution-29

Answer: a

Explanation: The cycle of unit-productions S ⟶ A ⟶ D ⟶ S says that any pair involving only S, A, and D is a unit pair. Similarly, the cycle S ⟶ B ⟶ E ⟶ S tells us that any pair involving S, B, and E is a unit pair.

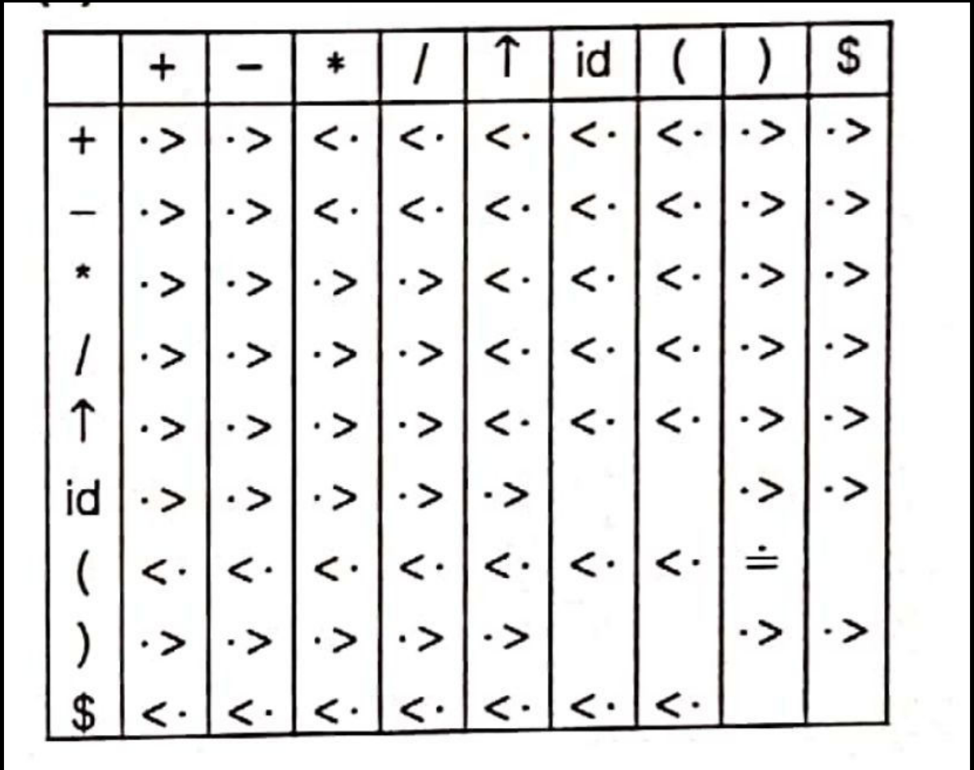
Solution-30

Answer: (c)

Here,+ has higher precedence than *. So the order of evaluation will be id1, id2 and id3

Solution-31

Answer: b



	+	−	*	/	↑	id	(	)	$
+	·>	·>	<·	<·	<·	<·	<·	·>	·>
−	·>	·>	<·	<·	<·	<·	<·	·>	·>
*	·>	·>	·>	·>	<·	<·	<·	·>	·>
/	·>	·>	·>	·>	<·	<·	<·	·>	·>
↑	·>	·>	·>	·>	<·	<·	<·	·>	·>
id	·>	·>	·>	·>	·>			·>	·>
(	<·	<·	<·	<·	<·	<·	<·	≐	
)	·>	·>	·>	·>	·>			·>	·>
$	<·	<·	<·	<·	<·	<·	<·		

Solution-32

Answer: A

Explanation: L1 is regular let us considering the string 011011011011 . Number of times 011 has occurred is 4 but also its occurrence is 3. Also if the string is ending with 011 we can make a 110 . Now the next string: 110110110110 in this 110 has occurred 4 times and 011 3 times which already satisfy the .

Solution-33

Answer: c

Explanation: The smallest possible string by given grammar is “11”.

X0 = 1X1

= 11X2 [Replacing X1 with 1X2]

= 11 [Replacing X2 with λ]

The string “11” is only possible with option 10 (0* + (10)*)*1.

Solution-34

Answer:(b)

S -> AB

S -> CA

B -> BC

B -> AB

A -> a

C -> aB | b

The above grammar is not in reduce form and reduced form is

S -> CA

A -> a

C -> aB

C -> b

Solution-35

Answer:(a)

The grammar's recognizing powers are as follows

LR(0) subset SLR(1) subset LR(1) subset LR(K)

more is the look ahead symbol number, more accurate is the recognizing power of the parser

Solution-36

Answer: d

Explanation: This relation is established of basis of the precedence of operators.

Solution-37

Answer: b

Explanation: A context-free grammar (CFG) is a set of recursive rewriting rules (or productions) used to generate patterns of strings.

Solution-38

Answer: d

Merging states with a common care may produce reduce-reduce conflicts but does not produce shift-reduce conflicts in LALR parser

The LR parsing method is the most general non backtracking shift reduce parsing method.

Solution-39

Answer: c

Explanation: Compiler transforms source code into the machine language which is in binary.

Kinds of object codes:

i. Defined symbols, which allow it to be called by other modules,

ii. Undefined symbols, which call the other modules where these symbols are defined, and

iii. Symbols which are used internally within object file for relocation.

Solution-40

Answer: d

Explanation: the next step in LR parsing shall have a Reduction

Solution-41

Answer:(b)

In this, first we draw the goto graph with LR(0) item then find the number of inadequate state so we get

	a
0	S_3/r_3

	action
2	$S_1/r_1/r_3 \mid r_1/r_3 \mid r_1/r_3$

So there are 3-shift reduce and 1-reduce-reduce conflicts. So that 4 conflict.

Solution-42

Answer: d
Explanation: A permanent database that has entry for each terminal symbols such as arithmetic operators, keywords, punctuation characters such as ‘;’, ‘,’etc Fields: Name of the symbol.

Solution-43

Answer: d

Explanation: A general-purpose macro processor or general purpose pre-processor is a macro designed primarily for string manipulation, macro definition.

Solution-44

Answer: d
Explanation: A linker or link editor is a computer program that takes one or more object files generated by a compiler and combines them into a single executable file, library file, or another object file.

Solution-45

Answer: b

Explanation: Input string 2) 3 + 4 S ” ER FR idR {print(2)} id)ER {print())} id) F+ER {print(+)}id) id + ER {print(3)} id) id) id +id So 2)+ 3 4 are printed.

Solution-46

Answer: a

Explanation: The predictive parser table is given as. Non Terminal) id

$ S S -> FR F

F -> id R

R ->) S

R ->! R ->!

So at M [S, id] = { S ” FR} M [R,$] = {R “!}

Solution-47

Answer: d

Explanation: Handles are the part of sentential form, & they are identified as the right side of any given production which will be used for reduction in the next step.

Solution-48

Answer: b

Explanation: Code optimizations are carried out on the intermediate code because program analysis is more accurate on intermediate code than on machine code.

Solution-49

Answer: . (d)

FIRST (X) = {s, e, ε}

FOLLOW (X) = {e, c, s, $}

	c	s	e	$
X	X->∈(4)	X->sX(3) X->∈(4) =E1	X->Yc(2) X->∈(4) =E2	X->∈(4)

E1 = {3, 4} and E2 = {2,4}

Solution-50

Answer:(b)

#error entries in LR(0) >= # error entries in SLR(1)

>= # error entries inLR(1)

G is LR(1) so no error in LR(1)

a >= b >= 0

Solution-51

Answer: d

Explanation: A two pass assembler does two passes over the source file (the second pass can be over a file generated in the first pass).

Solution-52

Answer .d

Statement I: FALSE

Symbol table is the data structure, which is used in all phase, that is, from lexical analysis till code generation and optimization.

Statement II: FALSE

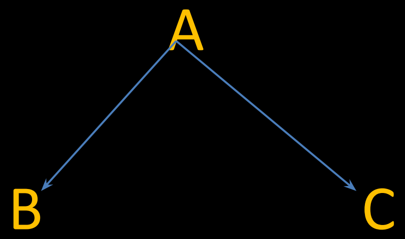
Recursion mandatorily requires stack memory during the runtime environment, not heap memory.

Statement III: FALSE

Error such as any variable must be declared before its use is semantic error and thus cannot be detected during syntax analysis.

Solution-53

Answer . (b)

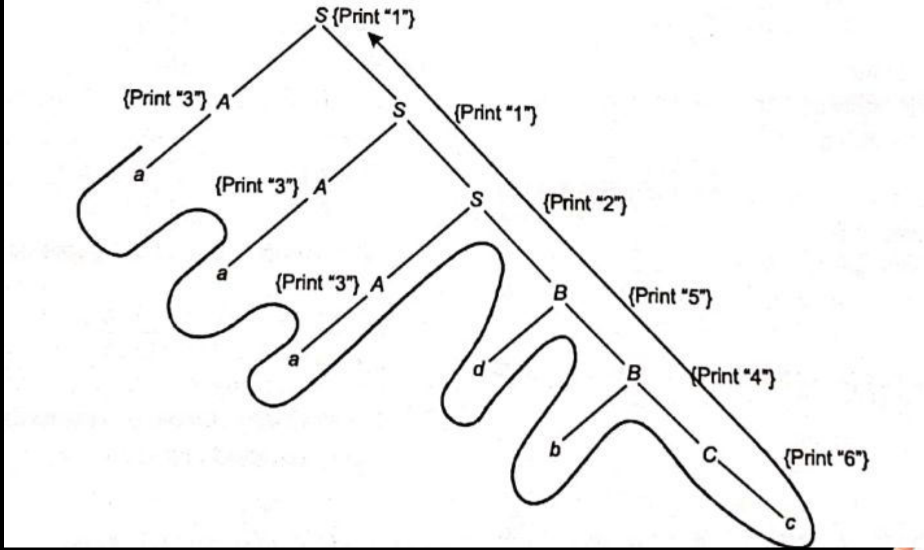


According to L attributed definition, B's attributes can depend on inherited attributes of A but not on synthesized attributes of A

Solution-54

Answer . (c)

The syntax directed tree for the given grammar can be represented as for given input "aaadbc“



Output printed will be: 3 3 3 6 4 5 2 1 1.

Solution-55

Answer .d

Concept:

Take relational algebra which generates the longest subsequence.

With T3 we get a subsequence of 5, and T_1 we get a subsequence of only 4, T_2 we get a subsequence of only 3.

Hence, T3 preferred over T2 and T1

Explanation

String = bbaacabc

(bla)* = (a + b)*

c? = (e + c)

T3: c? (ba)+c

T3 = bbaac (longest prefix match)

T3= abc

T3T3 = bbaac abc

Therefore option d is correct

Solution-56

Answer d

Concept: In LL(K) grammar, first L means scanning input from left to right and second L stands for left most derivation where k means k symbols are used at each step.

If first(production 1) n first(production 2) = empty, then it is LL(1)., it must be true for any two prductions.

For LL(2) we will check intersection of second (production) means second element of that production. Moving forward one step with symbol in bracket.

Explanation: Given grammar:

S→m|mn| mno

Case 1: For LL(1)

First(m) n first (mn) = m (which is not empty).So, grammar is not LL(1).

Case 2: For LL(2)

Second (mn) n second (mno) = n (which is not empty)So, grammar is not LL(2).

Case 3: For LL(3)

Third (mn) n third (mno) = empty.So, grammar is LL(3).

Download ExamPrepTool (EPT App)

https://play.google.com/store/apps/details?id=com.ept.requestteacher

For PYQ's with video solutions, Topper Notes, short notes, Toppers Test Series

Note: If any doubt in any question post with subject & question number in EPT App.

Solution-57

Answer d

Option a: Grammar is not ambiguous - TRUE

Because preferences of the above grammar are clearly defined.

Option b: Priority of + over* is ensured - TRUE

Because priority is decided by how farther a symbol appears from start symbol. Here '+' appears after *** hence, its priority is greater than *.

Option c: Right to left evaluation of* and + happens - TRUE

Because both are right associative as their generating non-terminals appear to be at the right side in both the rules.

Hence none of the solution is wrong.

Solution-58

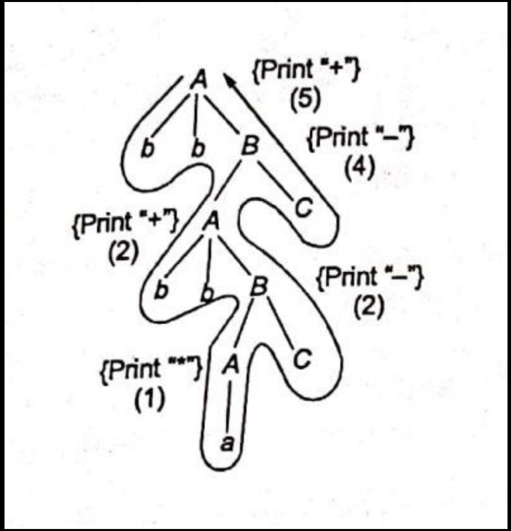
Answer: d

A⟶ bbB (Print "+"}

A⟶ a {Print "*"}

B⟶ Ac {Print "-"}

INPUT:bbbbacc



The above syntax tree prints *-+-+

Hence, (d) is correct option.

Solution-59

Answer: a

Explanation: Recursive Descent also known as top down parsing also known to be LL(1).

Consider the following two statements:

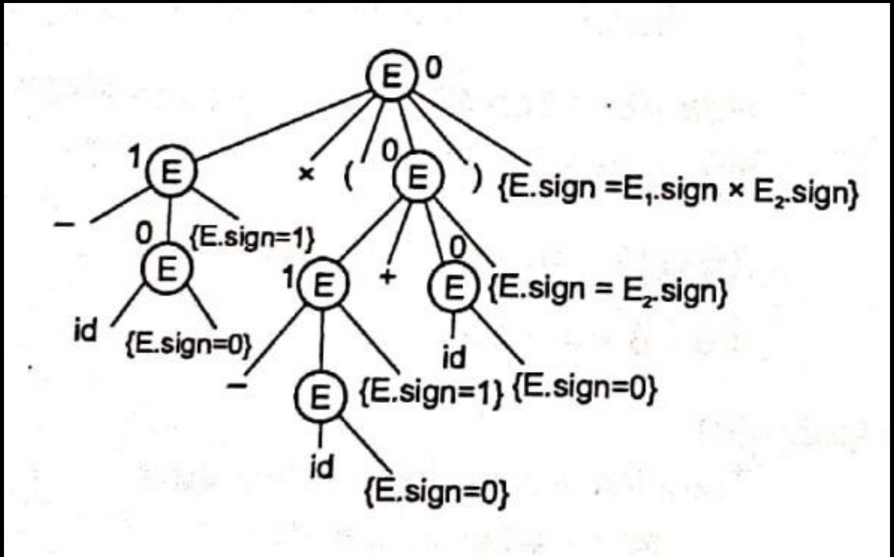
P: Every regular grammar is LL(1)

Q: Regular is LR(1) grammar

Solution-60

Answer: (a)

Input: -id x (-id + id)



The value computed at root is "0".

Solution-61

Answer: (c)

SDD is L-attributed iff:

The inherited attributes of Y, in the production.

$X \longrightarrow Y_1$ Y2... Yi Yi + 1 ... Y p

- Should only be derived from only attributes

  Y_1 to Y+1

-Should only be derived from inherited attribute of X.

Hence in first production rule Q is being derived from synthesized attribute of A. Hence SDD is not L-attribute because of 1's rule

Solution-62

Answer: (c)

b = (b+c)

d = (b+c) * d

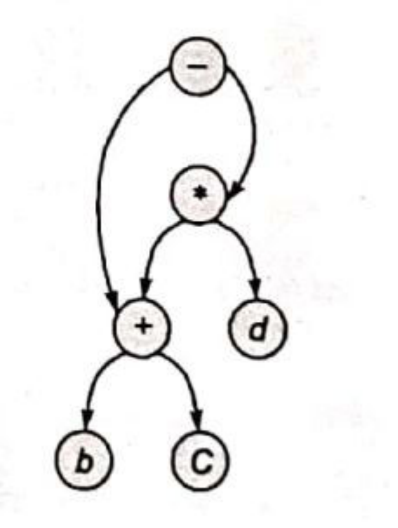
d = b* d

b= (b+c) * d-(b+c)

Final expresssion is

=>b= (b+c)* d- (b+c)

So, DAG representation for above expression is:



Number of nodes = 6

Number of edges = 6

Solution-63

Answer: a

Explanation: The first stage, the scanner, is FSM. It has encoded information on the possible sequences of characters that can be contained within any of the tokens it handles.

Solution-64

Answer: a

Explanation: If a grammar has more than one leftmost (or rightmost) derivation the grammar is ambiguous. Sometimes in unambiguous grammar the rightmost derivation and leftmost derivations may differ.

Solution-65

Answer. C

LR parsing is one sort of bottom up parsing. It is used to parse a broad category of grammars: "L" denotes left-to-right scanning of the input in LR parsing, "R" signifies the reverse construction of a right-most derivation and "K" is the number of input symbols used to determine the number of parsing decisions

LR parsing consists of four different parts: LR (0) parsing, SLR parsing, CLR parsing, and LALR parsing.

LR parsers in term of power:

CLR > LALR > SLR > LR(0)

Therefore, Option c) is correct option.

Download ExamPrepTool (EPT App)

https://play.google.com/store/apps/details?id=com.ept.requestteacher

For PYQ's with video solutions, Topper Notes, short notes, Toppers Test Series

Note: If any doubt in any question post with subject & question number in EPT App.