OVERALL ANALYSIS Solution Report

All Correct Answers Wrong Answers Not Attempted Questions

Q.1) Max Marks: 1

Which of the following statements is false?

A **Unambiguous grammar has both kinds of derivation** Correct Option

Solution: (A)
Answer: A
Explanation:
Unambiguous grammar has both kinds of derivation: False
In Unambiguous grammar both LMD and RMD generates the unique parse tree for the given input string

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)**

Q.2) Max Marks: 1

Assume that the CLR parser for a grammar G has n_1 states and the LALR parser for G has n_2 states. The relationship between n_1 and n_2 is

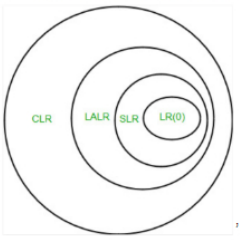
A n_1 is necessarily less than n_2

B n_1 is necessarily equal to n_2

C n_1 is necessarily greater than n_2

D n_1 is necessarily greater than or equal to n_2 Correct Option

Solution: (D)
Answer: D
Explanation:
Number of states in a CLR parser table greater than equal to LALR(1) parse table.



Q.3) Max Marks: 1

Consider the following grammar G = {bexpr, {bexpr, bterm, bfactor},{not, or, and, (,), true, false}, P } with P given below.
bexpr → bexpr or bterm | bterm
bterm → bterm and bfactor | bfactor
bfactor → not bfactor | (bexpr) | true | false
The equivalent non-left recursive grammar for the given grammar is

A bexpr→ bterm E'
E' → or bterm E' | ε
bterm → bfactor F'
F' → and bfactor F'
bfactor → not bfactor | (bexpr) | true | false

B bexpr→ bterm E'
E' → or bterm E' | ε
bterm → bfactor and F'
F' → bfactor F' | ε
bfactor → not bfactor | (bexpr) | true | false

C bexpr→ bterm E'
E' → or bterm E' | ε
bterm → bfactor F' Correct Option

F′ ⟶ and bfactor F′ | ε
bfactor ⟶ not bfactor | (bexpr) | true | false

Solution: (C)

Answer: C
Explanation:
Non-left recursive grammar for the given grammar is
bexpr⟶ bterm E'
E' ⟶ or bterm E' | ε
bterm ⟶ bfactor F'
F' ⟶ and bfactor F' | ε
bfactor ⟶ not bfactor | (bexpr) | true | false

D

bexpr⟶ bterm E'
E' ⟶ or bterm E'
bterm ⟶ bfactor F'
F' ⟶ and bfactor F' | ε
bfactor ⟶ not bfactor | (bexpr) | true | false

Q.4) Max Marks: 1

Consider the following Code fragment

```
int main()
{
  int x, y, total;
  x = 10, y = 20;
  total = x + y;
  printf ("Total = %d \n", total);
}
```

Number of tokens in the given code fragment is_____

Correct Answer

Solution: (34)

```
int main()                  -4
{                           -1
 int x, y, total;           -7
  x = 10, y = 20;           -8
  total = x + y;            -6
  printf ("Total = %d \n", total);    -7
}                                     -1
```

Total number of tokens are 4+1+7+8+6+7+1 = 34

Q.5) Max Marks: 1

Match the description of several parts of a classic optimizing compiler in **List - I**, with the names of those parts in **List - II**:

	List - I		**List - II**
(a)	A part of a compiler that is responsible for recognizing syntax.	(i)	Optimizer
(b)	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.	(ii)	Semantic Analysis
(c)	A part of a compiler that understand the meanings of variable names and other symbols and checks that they are used in ways consistent with their definitions.	(iii)	Parser
(d)	An IR-to-IR transformer that tries to improve the IR program in some way (Intermediate Representation).	(iv)	Scanner

Code :

	(a)	**(b)**	**(c)**	**(d)**
(1)	(iii)	(iv)	(ii)	(i)
(2)	(iv)	(iii)	(ii)	(i)
(3)	(ii)	(iv)	(i)	(iii)
(4)	(ii)	(iv)	(iii)	(i)

A **(1)** Correct Option

Solution: (A)
Answer: A
Parser is a part of compiler and responsible for syntax recognition. Scanner (or tokenization) used by the lexical analyzer. In Semantic analysis consistency and definition of syntax is checked. An optimizer is used improve the IR program. So option (A) is correct.

B **(2)**

C **(3)**

D (4)

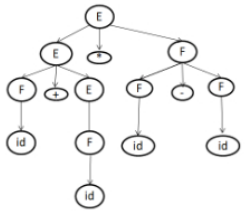
Q.6) Max Marks: 1

Given the following expression grammar:
E→E∗F | F+E | F
F→F−F | id
The output produced by the expression grammar after evaluating the expression
2+3*4-4

A 0 Correct Option

Solution: (A)

Answer:A
Explanation:
The parse tree for the given string is 2+3*4-4



The output is (2+3)*(4-4) = 5*0 =0

B 10

C 4

D None of these

Q.7) Max Marks: 1

In a bottom up evaluation of a syntax direction definition ,inherited attributes can

A Always be evaluated

B Be evaluated only if the definition is L -attributed Correct Option

Solution: (B)
A Syntax Directed Definition (SDD) is called S Attributed if it has only synthesized attributes.L-Attributed Definitions contain both synthesized and inherited attributes but do not need to build a dependency graph to evaluate them

C Evaluation only done if the definition has synthesized attributes

D None of these

Q.8) Max Marks: 1

Match the following items

(i) Backus-Naur form	(x) Regular expressions
(ii) Lexical Analysis	(y) LALR(1) grammar
(iii) YACC	(z) LL(1) grammars
(iv) Recursive descent parsing	(w) General context-free grammar

A i-w, ii-x, iii-y, iv-z Correct Option

Solution: (A)
Answer:A
Explanation:
Backus normal form (BNF) is a notation technique for context-free grammar, often used to describe the syntax of languages used in computing
Yacc (Yet Another Compiler-Compiler) is a computer program for the Unix operating system. It is a Look Ahead Left-to-Right (LALR) parser generator, generating a parser, the part of a compiler that tries to make syntactic sense of the source code, specifically a LALR parser, based on an analytic grammar.
Lexical Analysis with the help of Regular expressions identifies the number of tokens in the source program.
Recursive descent parsing and LL(1) parsers top down parsers without backtracking

B i-w, ii-x, iii-z, iv-y

C i-x, ii-w, iii-y, iv-z

D i-y, ii-x, iii-w, iv-z

Q.9) Max Marks: 1

Uniform symbol table

A Has all constants in the program

B Permanent table of rules in the form of patterns for matching with the uniform symbol table to discover syntactic structure

C Consists of full or partial list of the tokens as they appear in the program created by Lexical analysis and used for syntax analysis and interpretation — Correct Option

Solution: (C)
Answer:C
Explanation:
Uniform Symbols Table consists of a full or partial list of the token's as they appear in the program. Created by Lexical analysis and used for syntax analysis and interpretation

D A permanent table which has all key words and special symbols of the language in symbolic form

Q.10) Max Marks: 1

Consider the following statements related to compiler construction :
I. Lexical Analysis is specified by context-free grammar and implemented by pushdown automata.
II. Syntax Analysis is specified by regular expressions and implemented by finite-state machine.
Which of the above statement(s) is/are correct ?

A Only I

B Only II

C Both I and II

D Neither I nor II — Correct Option

Solution: (D)
Answer: D
Both the statements are incorrect. Lexical Analysis is specified by the Regular Expression and implemented by the finite state-machine and Syntax Analysis is specified by the CFG and implemented by the PDA.

Q.11) Max Marks: 2

Consider the following Grammar
S→ X
X → Yb | aa
Y→ a | bYa
The given grammar is

A LR(0) and SLR(1)

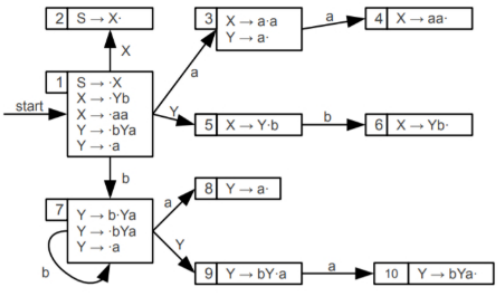
B Not LR(0) but SLR(1)

C LR(0) but not SLR(1)

D Neither LR(0) nor SLR(1) — Correct Option

Solution: (D)

Answer: D



State I3 contains S/R conflict in LR(0). Applying reduction operation on state I3 and shift operation for the input symbol 'a'.
Need to check for the S/R conflict in SLR(1)
First(a) intersection Follow(Y) = {a} intersection {a,b} = {a} ≠ Φ

S/R conflict in SLR(1).
The given grammar is not LR(0) and SLR(1).

Q.12) Max Marks: 2

Consider the following syntax directed translation scheme.
$E \rightarrow E_1 * T \{E.val = E_1.val * T.val\}$
$E \rightarrow T \{E.val = T.val\}$
$T \rightarrow F - T_1\{T.val = F.val - T_1.val\}$
$T \rightarrow F \{T.val = F.val\}$
$F \rightarrow 3 \{F.val = 2\}$
$F \rightarrow 5 \{F.val = 4\}$
The output produced by the SDTS after evaluating the given expression is 5-3*5*3.
Assume attribute evaluation with bottom-up parsing, i.e., attributes are evaluated immediately after a reduction.

A 15

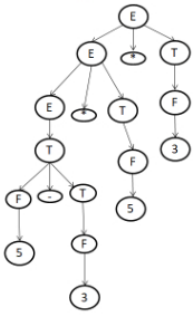
B -42

C 30

D None of these — Correct Option

Solution: (D)

Given string is 5-3*5*3



(((5-3)*5)*3) ⇒
$F \rightarrow 3 \{F.val = 2\}$
$F \rightarrow 5 \{F.val = 4\}$
(((4-2)*4)*2) ⇒ 16

Q.13) Max Marks: 2

Consider the following context free grammar
$S \rightarrow P$
$P \rightarrow (P)P$
$P \rightarrow \varepsilon$
In the LL(1) parse table M, the entries for M[S, $] M[P,)] are

A $S \rightarrow P, P \rightarrow \varepsilon$ — Correct Option

Solution: (A)

Answer: A
Explanation:
First(S) = {(, ε}
First(P) = {(, ε}
Follow(S) = {$}
Follow(P) = {), $}

	(	)	$
S	S→ P		S→ P
P	P→ (P)P	P→ ε	P→ ε

B $S \rightarrow \varepsilon, P \rightarrow \varepsilon$

C $P \rightarrow \varepsilon, P \rightarrow (P)P$

D None of these

Q.14) Max Marks: 2

Given the CFG G = {S, {S, U,V, W}, {a, b,c,d}, P } with P given as shown below:
S → UVW
U → (S) | aSb | d
V→ aV | ε
W → cW | ε
Then Follow(U) =

A {a,c}

B {a,c,b,)}

C { a,c,$,b}

D {a,c,$,b,)} Correct Option

Solution: (D)

Answer:D
Explanation:
Follow(U) = First(V) = {a, ε}
⇒ {a} u First(W) ={a} u {c, ε}
Substitute {ε} in place of W
⇒ {a} u {c} u Follow(S)
⇒ {a,c,$,b,) }

Q.15) Max Marks: 2

Consider the following context free grammar
S→ bAB | bb | C
A→ BC | aCB | ε | a
B→ bB | C | ε
C→ aaC | bbC | D
D→ a | b
The Equivalent simplified (After Elimination of Unit, Null, Useless Symbols) from the grammar is

A
S→ bAB | bb | aaC | bbC | a| b
A→ BC | aCB | a
B→ bB | aaC | bbC | a | b
C→ aaC | bbC | a | b
D→ a | b

B
S→ bAB | bb | aaC | bbC | a| b
A→ BC | aCB | a
B→ bB | aaC | bbC | a | b
C→ aaC | bbC | a | b

C
S→ bAB | bb | aaC | bbC
A→ BC | aCB | a
B→ bB | aaC | bbC | a | b
C→ aaC | bbC | a | b
D→ a|b

D None of these Correct Option

Solution: (D)

Explanation:

S→ bAB | bb | C
A→ BC | aCB | ε | a
B→ bB | C | ε
C→ aaC | bbC | D
D→ a | b
Eliminate Null productions
S→ bAB | bb | C | bB | bA | b
A→ BC | aCB | a | C | aC
B→ bB | C | b
C→ aaC | bbC | D
D→ a | b
Eliminate the Unit productions
S→ bAB | bb | aaC | bbC | a| b | bB | bA | b
A→ BC | aCB | a| aaC | bbC | a| b | aC
B→ bB | aaC | bbC | a| b | b
C→ aaC | bbC | a| b
D→ a | b
Eliminate Useless Symbols
S→ bAB | bb | aaC | bbC | a| b | bB | bA | b
A→ BC | aCB | a| aaC | bbC | a| b | aC

B→ bB | aaC | bbC | a| b | b
C→ aaC | bbC | a| b

close