Branch : CSE & IT Batch : Hinglish

Compiler Design

Syntax Directed Translation

DPP

[MCQ]

1. Synthesized attributes can be easily simulated using
(a) LL grammar
(b) LR grammar
(c) ambiguous grammar
(d) None of these

[MCQ]

2. Consider the following translation rules for the grammar G:

S → a{print "A"} A

A → b {print "C"}B

A → ∈ {print "C"}

B → e {print "B"}A

B → ∈ {print "C"}

C → c {print "A"}

What will be the output for the input string abesebe you top-down parser?
(a) ACBCCBAC
(b) ACCBCCBC
(c) ACBCCBCC
(d) ACBCBCBC

[MCQ]

3. Consider the following attribute grammar:

A → BA'	A'· b = a · a
	A · a = A' · b
A_1' → +BA_2'	A_2' = A_2'b + B · a
	A_1'a = A_2'a

Which of the following is true?
(a) Both a and b are inherited attributed.
(b) Both a and b are synthesized attributed.
(c) a is inherited, b is synthesized
(d) b is inherited, a is synthesized

[MCQ]

4. Consider the following grammar:

E → E + T | T

T → T/F | F

F → F ∗ A | A

A → id

Which one of the following is true?
(a) / have higher precedence than ∗
(b) ∗have higher precedence than +
(c) + have lower precedence than/
(d) ∗, +, / all have some precedence.

[MCQ]

5. A shift reduce parser perform action specified within process immediately after reduction to the corresponding rule of grammar.

S → abv {print '11'}

S → cc {print '2'}

V → Sd {print '33'}

What is the translation of ababccdd using the SDT scheme described by above rules/
(a) 2233113311 (b) 1133113322
(c) 2211331122 (d) 1122113322

[MCQ]

6. Consider the following transition rules:

A → BC

C → +BC | A + | ∈

B → DB {print '+'} | ∈

D → (A) id {print number value}

If input is given "2 + 34" then his translation scheme will generate output.
(a) 2 + 3 + 4 + (b) + 2 + 3 + 4
(c) + + 2 + 34 (d) 2 + 34 + +

[MCQ]

7. ____is performed by attaching rules or algorithms to production in a grammar.

(a) Lexical analysis
(b) Execution
(c) syntax directed translation
(d) None of these.

[MCQ]

8. Consider a translation scheme is given as:

$S \rightarrow S_1 + S_2$ {S · val = S_1· val + S_2 · val}

$S \rightarrow S_1 * S_2$ {S · val = S_1· val * S_2 · val}

S → id {S · val = id}

What will be the output for $5 * 6 + 7$?

(a) 18
(b) 37
(c) 65
(d) Cannot be identified because it is ambiguous grammar.

[MCQ]

9. Consider the given translation rules.

If the expression 8 # 12 & 4 # 16 & 12 # 4 & 2 is evaluated to 512, then which of the following is correctly representing x?

E → E # T	{E·val = E1· val * T · val}
IT	{E · val = T · val}
T → T & F	x
IF	{T · val = F · val}
F → id	{F · val = id}

(a) T · val = T_1 · val * f · val
(b) T · val = T_1 · val + f · val
(c) T · val = T_1 · val – f· val
(d) T · val = T · val ÷ f · val

[NAT]

10. Consider the following SDT :

S → E	{S · val = E · val}
E → E + T	{E · val} = E_1 · val + T · val}
E → T	{E · val = T · val}
T → T F	{T ·v al = T_1 · val * f· val}
T → F	{T · val = f · val}
F → (E)	{F · val = E · val}
F → a	{f · val = a}

What will be the output of the expression "20 + 8 × 6"

❑❑❑

Answer Key

1. **(b)**
2. **(d)**
3. **(d)**
4. **(b, c)**
5. **(a)**
6. **(d)**
7. **(c)**
8. **(d)**
9. **(c)**
10. **(68)**

❑❑❑

Hints & Solutions

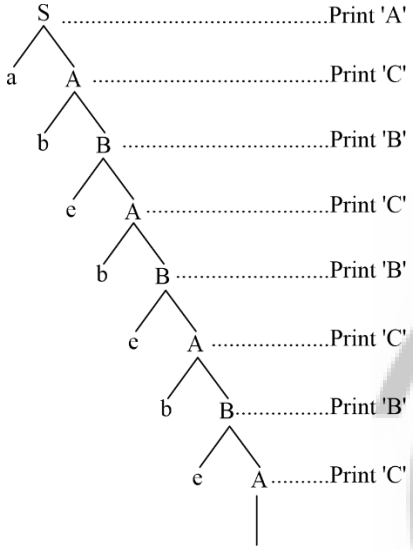
1. (b)

Synthesized attributes can be easily simulated by LR grammar.

As Synthesized attribute are evaluated in bottom-up manner and LR grammar is also evaluated in bottom-up manner.

2. (d)

Evaluating given grammar using top-down parser.



So, the output will be ACBCBCBC.

Therefore, option (d) correct.

3. (d)

For the first production a, b are attributes, such that a ∈A, and b ∈ A'.

Now if you look into the translation closely,

A' · b = A · a (b attribute is taking value from its parent attribute)

A'· a = A' · b (a attribute is taking value from one of its child attributes)

Similarly for second production, b is taking values from its parent as will as sibling which left to it (it is ∠ -attributed as well)

So, (b) follows inherited attribute definition while a follows synthesized attribute definition.

4. (b, c)

Here, ∗ have highest precedence, then/have precedence and + have least precedence

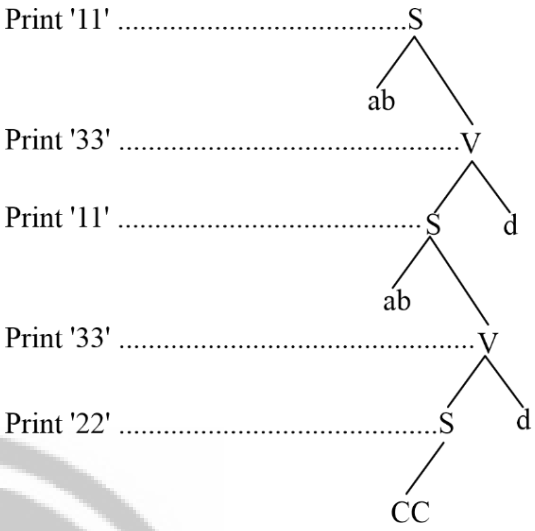
> / > +

As, ∗ will be evaluated first, so it has highest precedence.

Therefore, option (b) and (c) are correct.

5. (a)

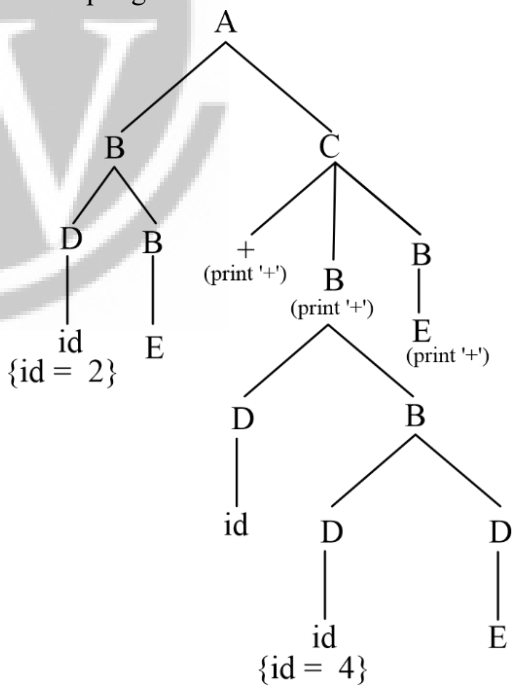
Given input string ababccdd



So, output will be 2233113311. Therefore option (a) is correct.

6. (d)

The input given is "2 + 34"



On scanning them left to right.

Output = 2 + 34 ++

So, option (d) is correct.

7. (c)

SDT is done by attaching augmented rules to the analysis. SDT rules used the following:

1. lexical value of node
2. Constants
3. Attributes associated to the non-terminal in their definition.

8. (d)

The given grammar is ambiguous grammar

S → S1 + S2 | S1 * S2 | id an ambiguous grammar.

Because for expression 5 * 6 + 7 two output are possible.

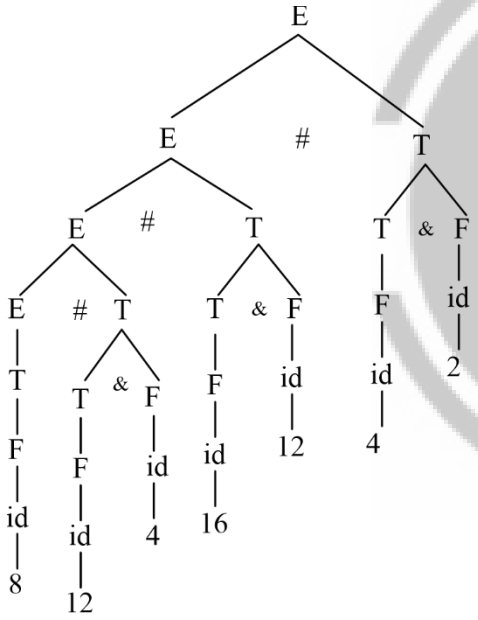
Output 1 : 37

Output 2 : 65

So option (d) is correct answer.

9. (c)

8 # 12 & 4 # 16 & 12 # 4 & 2 = 512



8 # 12 & 4 # 16 & 12 # 4 & 2

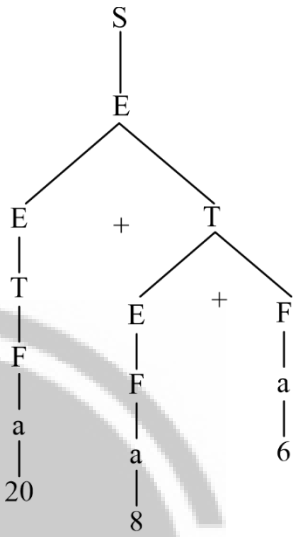
⇒ (8 # (12 & 4)) * (16 & 12) * (4 & 2)

If & = –

Then, ((8 ∗(12 – 4) ∗ (16 – 12) ∗ (4 – 2) = 8 ∗ 8 ∗ 4 ∗ 2 = 512

So, option (c) is correct answer.

10. (68)



20 + 8 × 6 = 68

❑❑❑