Branch : CSE & IT Batch : Hinglish

Compiler Design

Lexical Analysis & Syntax Analysis

DPP

[NAT]

1. Consider the following C program:

```
int main ( )
{
/*finding maximum element out of a & b*/
int a, b max;
a = 10; b = 20
if (a < b)
        max = b;
else
        max = a;
return (max);
}
```

Calculate the total number of tokens present in the program?

[MSQ]

2. Consider the following C-program:

```
1. int main )(
2. {
4. x = a + b* c;
5. y = x + a ;
6. char f = 'e' ;
7. in t g = 200;
8. ch/* comment ar = "gate";
9. }
```

Which of the following is correct regarding above program?

(a) The given program has 47 tokens.
(b) Given program produces compilation error
(c) Given program produces lexical error
(d) No error produced by program

[MCQ]

3. Compiler's first phase makes uses of following patterns for token (S_1,S_2,S_3) reorganization over the alphabet a,b,c.

S_1: b#(b|a)*c
S_2: c#(c|b)*a
S_3: a#(b|c)*b

Note: x# means 0 or 1 occurrence of the symbol x. The analyzer outputs the token that matched the longest possible prefix of the string. If abbbccccba is processed by first phase of compiler then which one of the following is the sequence of token of output.

(a) S_1, S_2, S_3 (b) S_1, S_2
(c) S_3, S_2 (d) S_3, S3

[NAT]

4. How many of the following strings are said to be tokens in C-language without looking at next input character?

(i) ; (ii) return
(iii) int (iv) (
(v) && (vi) >>

[NAT]

5. Consider the following C-program:

```
int main ( )
{
int x; /* comment */
x = = = y /*abcd***/*abcd*/;
int **p;
int b = 10, y;
x = *p ++ + ++ y;
}
```

How many tokens are present in the given program?

[MCQ]

6. Which of the following is equivalent unambiguous grammar for following rules?

Operator	Priority	Associativity
↑, #	3	Left to right
⊕, *	2	Right to left
–, =	5	Left to right
/, &	1	Right to left
+, $	4	Left to right

Note: 5 has the highest priority and 1 has the least priority.

(a) $A \rightarrow B \uparrow A \mid B\#A \mid B$
$B \rightarrow C \oplus B \mid C*B \mid C$
$C \rightarrow C - D \mid C = D \mid D$

D→ D+E| D $ E | E
E→ E–F| E=F | F
F→ id

(b) A→ A↑B |A # B| B
B→ C⊕B |B * C| C
C→ C–D |C = D| D
D→ D+E |D $ E| E
E→ F–E |F = E| F
F→ id

(c) A→ A/B |A & B| B
B→ B⊕C |B * C| C
C→ C↑D |C # D| D
D→ D+E |D $ E| E
E→ F–E |F = E| F
F→ id

(d) A→ B/A |B & A| B
B→ C⊕B |C * B| C
C→ C↑D |C # D| D
D→ D+E |D $ E| E
E→ E–F |E = F| F
F→ id

[MCQ]

7. What will be the equivalent grammar after removing left factoring from the given grammar?
S → a|ab| abc|abcd|e|f

(a) S→aS'
S→ b|c|d|e|f

(b) S→ e|f|S'
S'→ a|ab|abc|abcd

(c) S→ e|f|aS'|∈
S'→ bA'|∈
A'→ cB'|∈
B'→ d

(d) S→ e|f|aS'
S'→ bA'|∈
A'→ cB'|∈
B'→ d|∈

[MSQ]

8. Which of the following is correct regarding FIRST & FOLLOW of the given grammar.
E→TE'
E'→+TE' |∈
T→FT'
T'→*FT' |∈
F→id |(∈)

(a) FIRST (T)= {id, (}
FOLLOW(T') = {+, $,)}

(b) FIRST (E) = {id, (}
FOLLOW(E') = {$,)}

(c) FIRST (E) = {id, (}
FOLLOW(F) = {+, $,), *}

(d) FIRST (T') = {*, ∈}
FOLLOW (E) = {$, * ,)}

[NAT]

9. Consider the following grammar:
S→ BB
B→ aB|b
How many items are there in Closure (S' →.S, $)?

[NAT]

10. Consider the given grammar:
S→ (A) | a
A→ SA'
A' → ,SA' | ∈
Assume that initially stack has 2 symbols $S on stack. Then, what will be the maximum size of stack during LL(1) parsing for the input string (a, a)?

[NAT]

11. Consider the following
A→ B = C/C
C→ B
B→ *C | id
How many number of states are required for above grammar using CLR parser?
Note: A, B & C are non-terminal and *, =, id are terminal.

[MCQ]

12. How many conflicting entries are there in SLR(1) parse table of the following grammar?

S→EeEf

S→FfEe

E→x

f→x

(a) 1 (b) 2

(c) 3 (d) 4

[MSQ]

13. Consider following grammar G.

S→ Aa| Bb

A→ Ac| ∈

B→ Bc| ∈

Which of the following statement is true?

(a) It has shift-reduce conflict in the first state of the LR(0) machine.

(b) It has reduce-reduce conflict in the first state of the SLR(1) machine.

(c) It has reduce-reduce conflict on the first state on the LR(0) machine.

(d) It has shift-reduce conflict in the first state of the SLR(1) machine.

❑❑❑

[MSQ]

14. Consider the following grammar:

P→ Qx | yQz | tz | ytx

Q→ t

Which of the following is correct for above grammar?

(a) It is LR(1) (b) It is LL (1)

(c) It is LALR(1) (d) It is CLR (1)

[NAT]

15. The maximum number of reduce moves that can be taken during bottom-up evaluation of 21 token string by using a bottom-up parser. Assuming the grammar has no epsilon and unit production.

Answer Key

1. **(41)**
2. **(b,c,)**
3. **(c)**
4. **(2)**
5. **(39)**
6. **(d)**
7. **(d)**
8. **(a,b,c)**
9. **(4)**
10. **(4)**
11. **(13)**
12. **(b)**
13. **(b,c)**
14. **(a,c,d)**
15. **(20)**

❑❑❑

Hints & Solutions

1. (41)

The total tokens in the program are:

int(1) main(2) (3))(4)
{(5)
/* finding maximum element out of a &b */
int(6) a(7) ,(8) b(9) ;(10) max(11) ;(12)
a(13) =(14) 10(15) ;(16) b(17) 20(19) ;(20)
if(21) (22) a(23) <(24) b(25))(26)
max(27) =(28) *b*(29) ;(30)
else(31)
max(32) =(33) a(34) ;(35)
return(36) (37) max(38))(39) ;(40)
}(41)

Total token in above program are: 41

2. (b, c)

The given program generates compilation and lexical error.

Therefore, option b, c are correct.

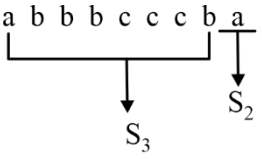
3. (c)

Minimum string of S_1: c

Minimum string of S_2: a

Minimum string of S_3: b

It is given that prefer longest matching. So,



Therefore, option (c) is correct answer.

4. (2)

Among all of the above ; , (are the only token for whom we need not to check next input character.

return can have next input character as returna which could be a variable name, similarly int..

&& can be **&&=** ·Similarly, >> could be **>>=**

5. (39)

The given program is:

int(1) main(2) (3))(4)
{(5)
int(6) x(7) ;(8) /*comment*/
x(9) =(10) =(11) =(12) y(13) /*abcd ***/ *(14) abcd(15) *(16) /(17) ;(18)
int(19) b(20) =(21) 10(22) ,(23) y(24) ;(25)
x(26) =(27) *(28) *p*(29) ++(30) +(31) ++(32) y(33) ;(34)
}(35)

There are total 39 tokens in program.

6. (d)

Given that

/, & has least priority and right to left associativity. So, it will be evaluated at last.

–, = has highest priority. It must be evaluated first.

The equivalent unambiguous grammar would be.

A→ B | A | B & A| B

B→ C⊕ B | C * B | C

C→ C ↑ D | C # D | D

D→ D + E | D $ E | E

E→ E – F | E = F | F

F→ id.

So, option (d) is the correct answer.

7. (d)

On eliminating left factoring from

S→ a |ab|abc|abcd|e|f

The equivalent grammar will be.

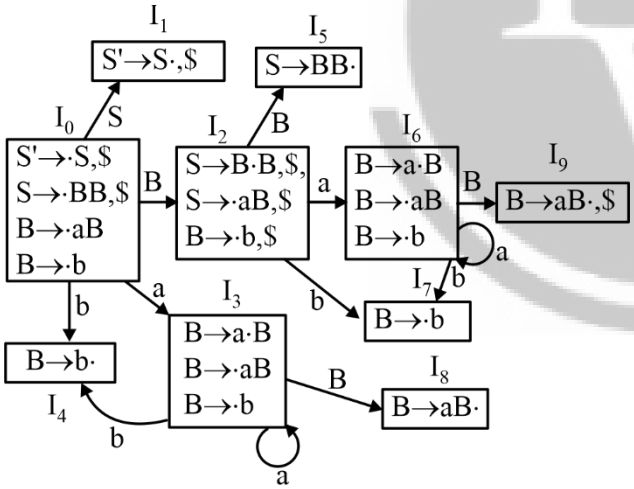
S→ e | f | aS’
S’→ bA’ |∈
A’→ cB’ |∈
B’→ d |∈
So, option (d) correct answer.

8. (a, b, c)
Given that

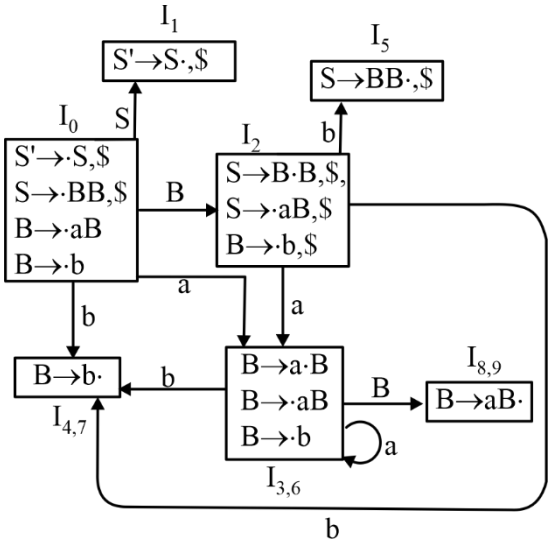
	Frist	**Follow**
E –	{(, id}	{$,)}
E’ –	{+, id}	{$,)}
T –	{id, (}	{+, $,) }
T’ –	{*, ∈}	{+ , $,)}
F –	{id, (}	{+, $,) , * }

So, option a, b, c are correct.

9. (4)
Augmented grammar:
S→ ·S, $
S→ ·BB, $
B→ ·ab
B→ ·b·



The closure of (S’→.S, $) have 4 items.



LALR (1) parser has 7 states.

10. (4)
The given non-left recursive grammar is:
S→ (A) | a
A→ SA’
A’ → ,SA’ | ∈
Now, First (S) = { (, a) }
Follow (S) = { $, t }
First A = { , a }
Follow A = {) }
First A’ = {Follow (A’) = Follow (A’) ={) }
So, parsing table of LL(1) will be as follows.

	(	)	A	,	$
S	S→(A)		S→a		
A	S→SA’		A→SA’		
A’		A’→∈	A’→SA’		

For string (a,a)

Stack	Input	Output
$S	(a,a)$	S→(A)
$)A(	(a,a)$	
$)A	a,a)$	S→A’
$)A’S	a,a)$	S→a
$)A’a	a,a)$	
$)A’	, a)$	
$)A’S	, a)$	A’→SA’
$)A’S,	a)$	
$)A’a	a)$	S→a

$)A'	)$	
$)	)$	A'→∈
$	$	

So, maximum stack size is 4.

11. (13)

The augmented grammar for the given grammar is:

A' → ·A,$

A → ·B = C , $

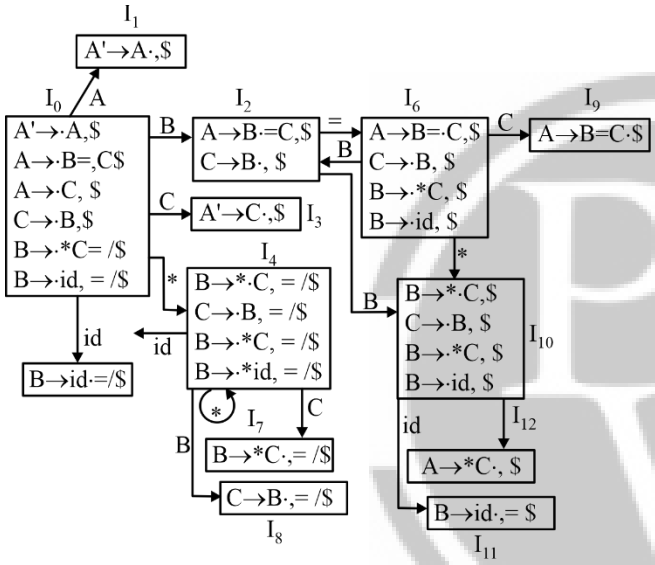
A → ·C , $

C → ·B , $

B → ·* C, = /$

B → ·id , = /$

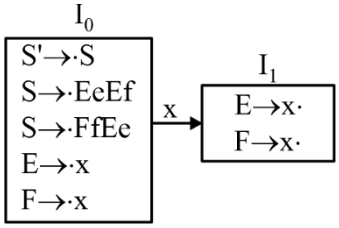
State diagram is as follows:



So, there are total 13 states needed for given grammar.

12. (2)

The LR(0) items of the grammar is:



R- R conflict (Reduce Reduce conflict)

In SLR(1) parse table on item I_1, E→ x & F→x are to be reduced in Follow (E) and Follow (F) respectively, which is both terminals e and f.

So, the number of conflicting entries in SLR(1) parse table is 2.

13. (b,c)

Here is the first state of the LR(0) machine.

S' →·S

S →·Aa

S →·Bb

A →·Ac

A →·∈

B →·Bc

B →·∈

Follow(A) = {a,c} & Follow (B) = {b,c}. Here it has reduce- reduce conflict between production (A→∈) and production (B→∈).

14. (a,c,d)

Augmented grammar:

P' →.P, $

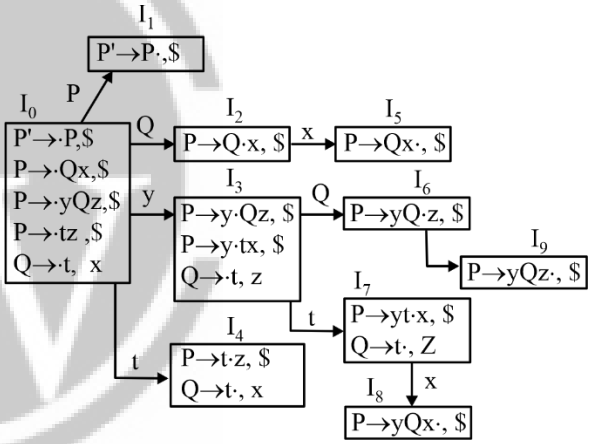
P →·Qx, $

P →·yQz, $

P →·tz, $

P →·ytx, $

Qr →·t, x



There are no conflict present so it will LALR(1) and no state is different in just look ahead symbol so it will also be LALR(1).

First (P)= First (Qx) ∩ First (yQz) ∩ First (tz) ∩ First (ytx)

= t ∩ y ∩ t ∩y

First (P) ≠ ϕ

So, it is not LL(1).

15. (20)

Maximum number of reduce moves for n tokens = n – 1.

So, for 21 token, 20 reduce moves are required.

❑❑❑

Any issue with DPP, please report by clicking here:- https://forms.gle/t2SzQVvQcs638c4r5

For more questions, kindly visit the library section: Link for web: https://smart.link/sdfez8ejd80if

PW Mobile APP: https://smart.link/7wwosivoicgd4