Niket Keshari
2017 4013
Compiler Construction Assign - 2

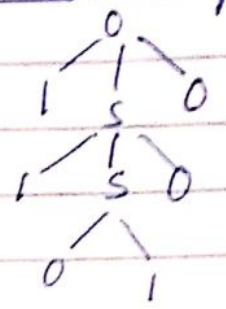
Q.1 (a) $S \to 1S0 | 01$

$\to 1S0$

$\to 11S00$

$\to 110100$

Parse tree (left deriv)



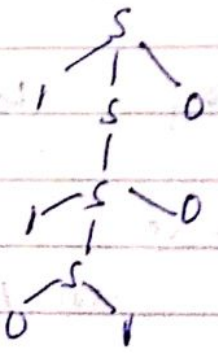
rightmost derivation

$S \to 1S0$

$\to 11S00$

$\to 110100$

rightmost deriv:-



(b) $S \to S+S | SS | (S) | S* | a$ to generate $\to (a+a)*a$

Left deriv:-

$S \to SS$

$\to S*S$

$\to (S)*S$

$\to (S+S)*S$

$\to (a+S)*S$

$\to (a+a)*S$

$\to (a+a)*a$

Right deriv:-

$S \to SS$

$\to Sa$

$\to S*a$

$\to (S)*a$

$\to (S+S)*a$

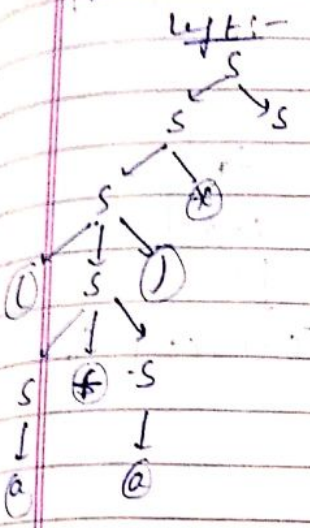
$\to (S+a)*a$

$\to (a+a)*a$

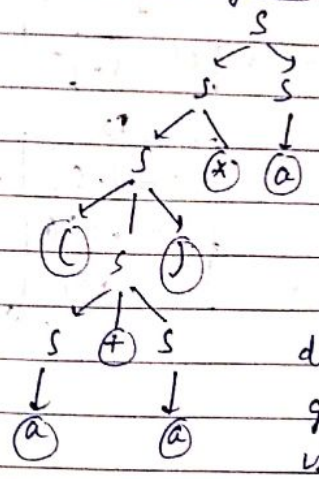
Scanned with CamScanner

Parse trees:-

Left:-



Right:-



since, left & right deriv. are same, hence given grammer is unambiguous.

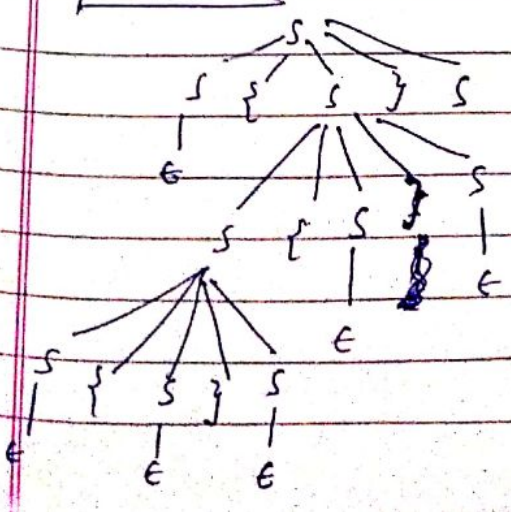
(c) left deriv:

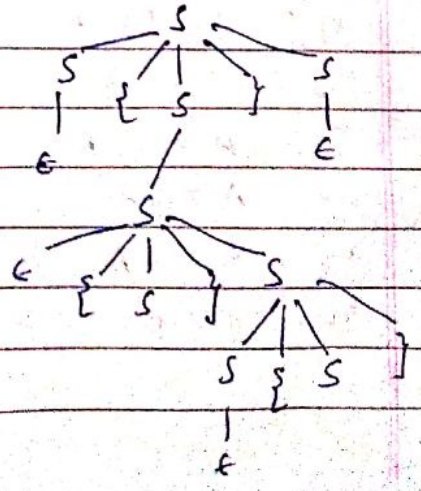
$S \to S\{S\}S$
$\to \{S\}S$
$\to \{S\{S\}S\}S$
$\to \{\{S\}S\}S$
$\to \{\{\}S\}S$
$\to \{\{\}S\{S\}S\}S$
$\to \{\{\}\{S\}S\}S$
$\to \{\{\}\}S\}S$
$\to \{\{\}\}S\}S$
$\to \{\{\}\{\}\}S$
$\to \{\{\}\{\}\}$

right deriv:

$S \to S\{S\}S$
$\to S\{S\}$
$\to S\{S\{S\}S\}$
$\to S\{S\{S\}S\{S\}S\}$
$\to S\{S\{S\}S\{S\}\}$
$\to S\{S\{\}S\{\}\}$
$\to S\{S\}S\{\}\}$
$\to S\{S\{\}\{\}\}$
$\to S\{\{\}\{\}\}$
$\to \{\{\}\{\}\}$

parse trees:-



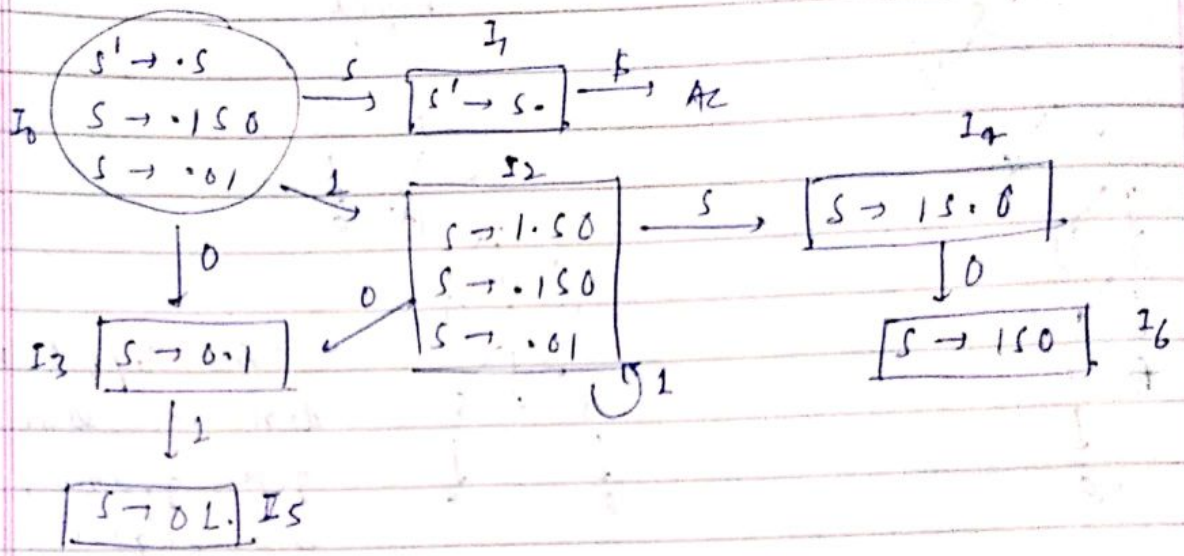


Scanned with CamScanner

Page :
Date : / /

since left & right trees are different hence ambiguous grammar

Q.2(a)



	1	0	S	$
0	2	3	1	
1				Ac
2	2	3	4	
3	5			
4		6		
5		r_3		
6		r_2		

Follow Table:-

	first	follow
S	10	$, 0

(b) $S \rightarrow S+S \mid SS \mid S* \mid (S) \mid a$

$S \rightarrow (S) A \mid aA$

$A \rightarrow +SA \mid SA \mid *A \mid \epsilon$

r_1 : $S \rightarrow (S) A$

r_2 : $S \rightarrow aA$

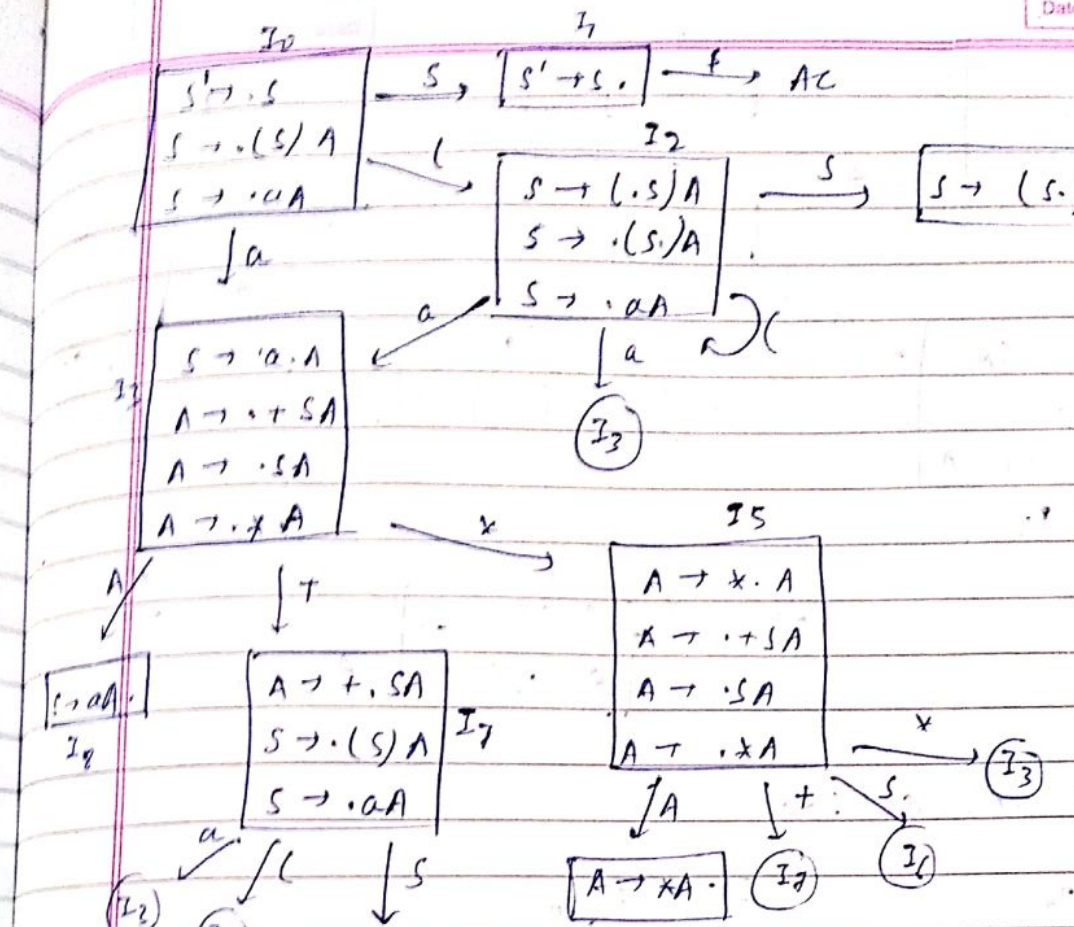
r_3 : $A \rightarrow +SA$

r_4 : $A \rightarrow SA$

r_5 : $A \rightarrow *A$

r_6 : $A \rightarrow \epsilon$

Scanned with CamScanner



Follow Table

	first	follow
S	(, a	$,), *, +, (, a
A	*, +, (, a	$,), *, +, (, a

grammar is unambiguous

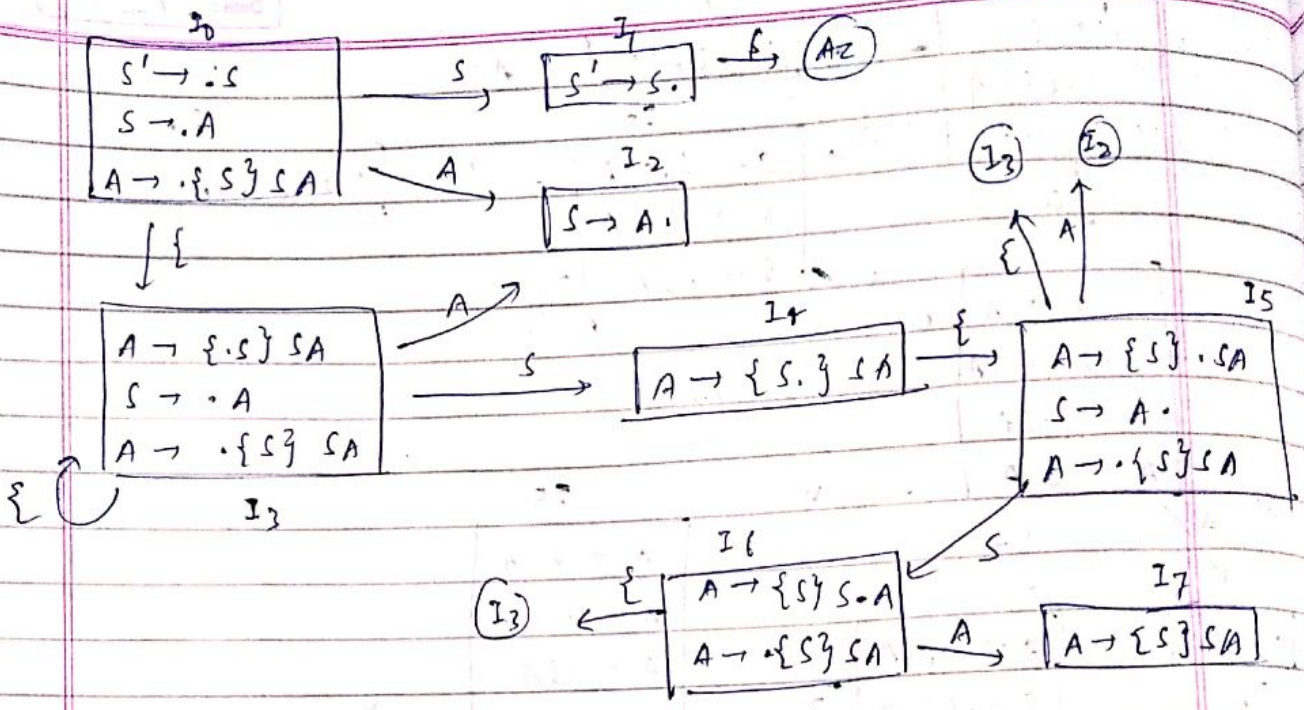
Q2 (c) $S \to S\{S\}S \mid \epsilon$

$r_1 : S \to A$

$r_2 : A \to \{S\}SA \mid \epsilon$

	{	}	$	S	A
0	s3			1	2
1			Acc		
2	r_1	r_1	r_1		
3				4	2
4		s5			
5	s3			6	2
6	s3				7
7	r_2	r_2	r_2		

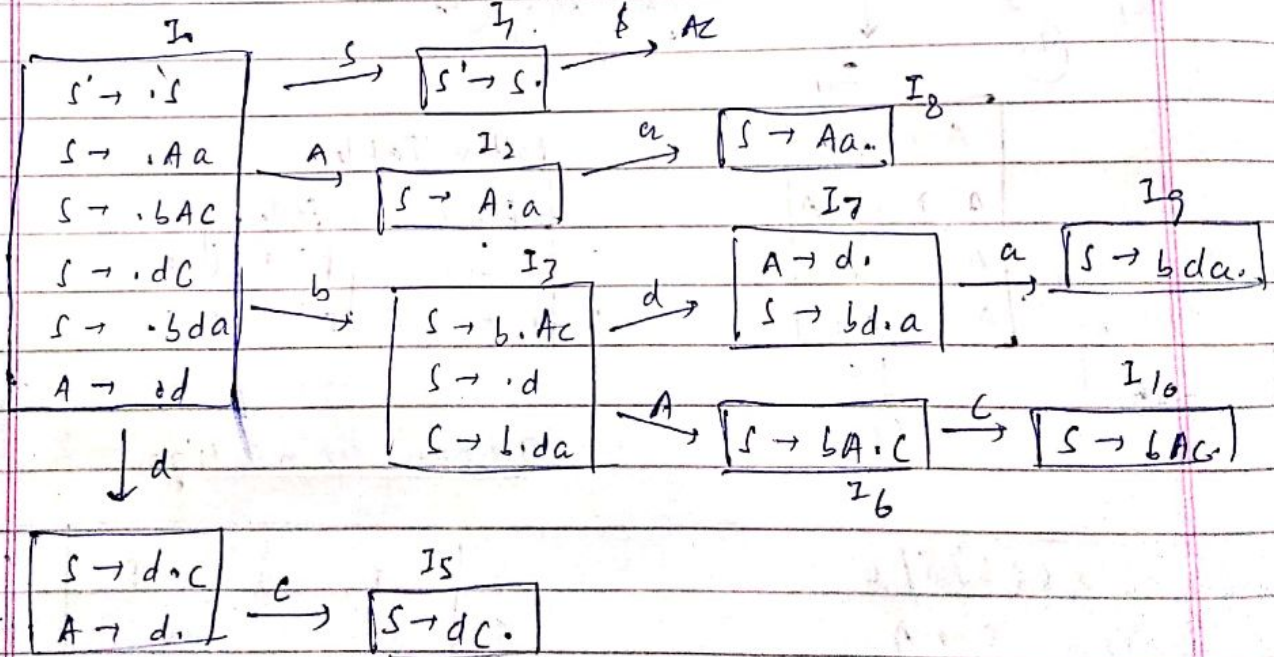
Scanned with CamScanner



Q.(1) $S \to Aa \mid bAc \mid dc \mid bda$

$A \to d$

SLR:



Rules:

r_1: $S \to Aa$

r_2: $S \to bAc$

r_3: $S \to dC$

r_4: $S \to bda$

r_5: $A \to d$

Scanned with CamScanner

	a	b	c	d	$	S	A
0		s_3		s_4/r_5		1	2
1					Ac		
2	s_8						
3				s_7/r_5		6	
4			s_5				
5			r_3				
6			s_{10}				
7	s_9			r_5			
8	r_1		r_1				
9	r_9						
10			r_2				

S/R conflict occurs grammer is not SLR

LALR:

I_0
- $S' \to .S, \$$
- $S \to .Aa, \$$
- $S \to .bAc, \$$
- $S \to .dc, \$$
- $S \to .bda, \$$
- $A \to .d, a$

I_1: $S' \to S., \$$ (on $\$$ → Ac)

I_2: $S \to A.a, \$$ (on a → I_5: $S \to Aa., \$$)

I_3
- $S \to b.Ac, \$$
- $S \to b.da, \$$
- $A \to .d, c$

I_4: $S \to d.c, \$$; $A \to d., a$ (on c → I_8: $S \to dc., \$$)

I_6
- $S \to bd.a, \$$
- $A \to d., c$

(I_6 on a → I_{10}: $S \to bda., \$$)

I_7: $S \to bA.c, \$$ (on c → I_9: $S \to bAc., \$$)

The given grammer is LALR(1) as it can be obtained by merging items with common first components.

There is no S/r conflict.

Scanned with CamScanner

Page :
Date : / /

q ④ given: $E \to E+T \mid T$

$T \to num.num \mid num$

production	semantic rules
$E \to E+T$	if (E.type == int & T.type = int) then E.type = int else E.type = float
$E \to T$	E.type = T.type
$T \to num.num$	T.type = float
$T \to num$	T.type = int

q ⑤ Given: r_1: crownsike → kaa_crownsike

r_2: crownsike → kaa

{ replaced " " space with "_" for simplicity

I_0:
- S' → .crownsike, $
- crownsike → .kaa_crownsike, $
- crownsike → .kaa, $

crownsike → I_1:
- S → crownsike., $ → (Ac)

kaa → I_2:
- crownsike → kaa._crownsike, $
- crownsike → kaa., $

_ → I_3:
- crownsike → kaa_.crownsike, $
- crownsike → .kaa_crownsike, $
- crownsike → .kaa, $

crownsike → I_4:
- crownsike → kaa_crownsike., $

kaa → I_2

	_	kaa	$	crownsike
0		S_2		1
1			Ac	
2	S_3		r_2	
3		S_2		4
4			r_1	

Scanned with CamScanner

Page :
Date : / /

parsing :-

	Input
0	kaa-kaa-kaa $
0 kaa 2	-kaa-kaa $
0 kaa 2 -3	kaa-kaa $
0 kaa 2 -3 kaa 2	-kaa $
0 kaa 2 -3 kaa 2 -3	kaa $
0 kaa 2 -3 kaa 2 -3 kaa 2	$
0 kaa 2 -3 kaa 2 -3 crownrise 4	$
0 kaa 2 -3 crownrise 4	$
0 crownrise	$

Since $, 1 → AC, hence the string is a part of grammer

Nikeet Keshari
2017400 13

Scanned with CamScanner