22-02-24
Thursday

Day-2 Analytical Problems

(1) Construct a recursive descent parser for the following grammar.

$E \rightarrow TE'$

$E' \rightarrow +TE' / \epsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' / \epsilon$

$F \rightarrow [E] / id$

(13) (a) Construct leftmost & rightmost derivation for the sentence abab $S \rightarrow aSbS / bSaS / \epsilon$

(b) Check whether following grammar is ambiguous or not $S \rightarrow [S]S$ $S \rightarrow E$

@ LMD & RMD

$S \rightarrow aSbS / bSaS / \epsilon$

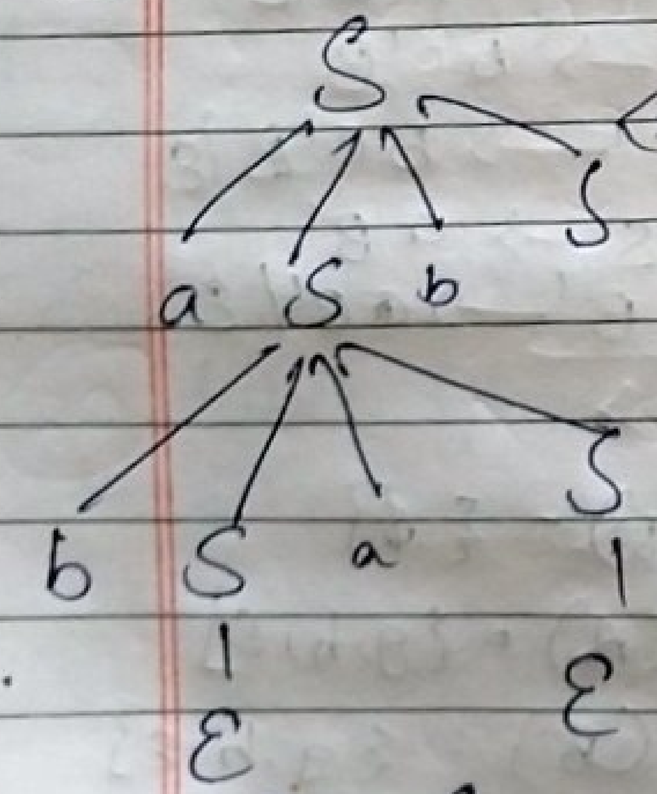
$S \rightarrow aSbS$
$S \rightarrow bSaS$
$S \rightarrow \epsilon$

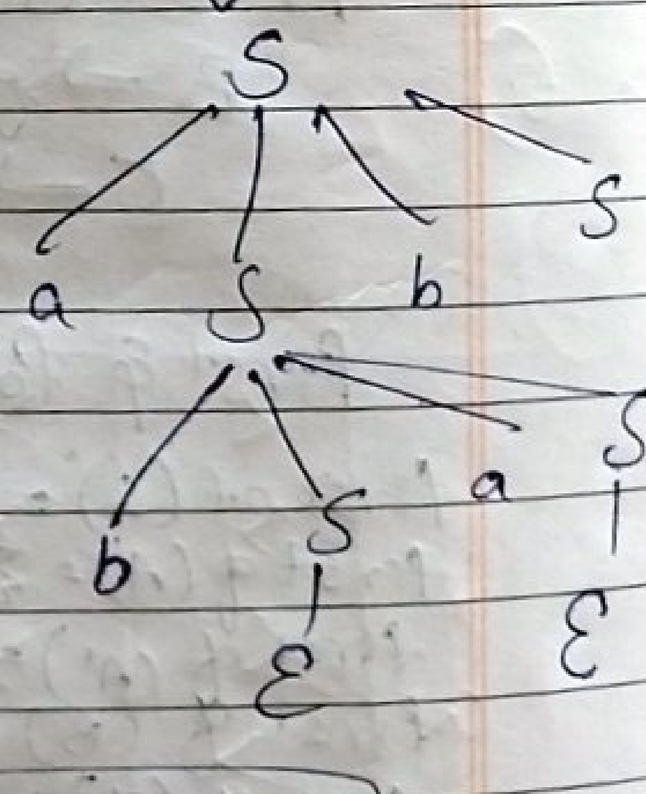
LMD: sentence = abab

$S \rightarrow aSbS$
$abSaSbS$
$abaSbS$
$abaSb$
$ababS$
$abab$

RMD:

$S \rightarrow aSbS$
$aSbS$
$aSb\epsilon$
$aSb\epsilon$
$abSa\epsilon b$
$ab\epsilon ab$
$abab$





(b) $S \rightarrow (S)S$

$S \rightarrow \epsilon$

String: abab

The given grammer is not an ambiguos grammar

(12) Construct a CFG for the language L = anb2n where n >= 1

$$L = \{ a^n b^{2n} \mid n \geq 1 \}$$

$$L = \{ aabbbb, aaabbbbbb \ldots \}$$

$$S \rightarrow aSbb$$

$$S \rightarrow abb$$

(11) Construct a CFG for a language $L = \{ wcw^R \mid w \in (a,b)^* \}$.

$$L = \{ wcw^R \mid w \in (a,b)^* \}$$

w^R denotes the reverse of string

S, A, B {start symbols}

$S \rightarrow aSa$

$S \rightarrow bSb$

$S \rightarrow A$

$A \rightarrow aA$

$A \rightarrow bA$

$A \rightarrow B$

$B \rightarrow aB$

$B \rightarrow bB$

$B \rightarrow \varepsilon$

(9) $S \rightarrow ACB \mid CbB \mid Ba$

$A \rightarrow da \mid BC$

$B \rightarrow g \mid \varepsilon$

$C \rightarrow h \mid \varepsilon$

$S \rightarrow CA'B$

$A' \rightarrow CA'B \mid \varepsilon$

$S \rightarrow bC'B$

$C' \rightarrow bC'B \mid \varepsilon$

$S \rightarrow aB'$

$B' \rightarrow aB' \mid \varepsilon$

first & follow functions:-

first of (S) = {cba}

first of (A) = {dc}

first of (A') = {c ε}

first of (B) = {g, ε}

first of (B') = {e, ε}

first of (C) = {h, ε}

first of (C') = {b, ε}

follow (S) = {$}

follow (A) = {g, h, ε}

follow (B) = {g, h, ε}

follow (C) = {g, h, ε}

$S \to CA'B \mid bC'B \mid aB'$

$D \to da \mid cB$

$A' \to CA'B \mid \varepsilon$

$B \to g \mid \varepsilon$

$B' \to cB' \mid \varepsilon$

$C \to h \mid \varepsilon$

$C' \to bC'B \mid \varepsilon$

(10) $S \to ABD$

$A \to a \mid Db \mid \varepsilon$

$B \to g\ D \mid dA \mid \varepsilon$

$D \to e \mid f$

$first(S) = \{A\}$

$first(A) = \{a, e, f, \varepsilon\}$

$first(B) = \{g, d, a, e, f, \varepsilon\}$

$first(D) = \{e, f\}$

$follow(S) = \{\$\}$

$follow(A) = \{g, d, a, e, f\}$

$follow(B) = \{g, d, a, e, f\}$

$follow(D) = \{g, d, a, e, f\}$

	a	d	e	f	g	$
S						
A	A		a	a		ε
B	B	B	B	B	B	ε
D			B	B		

→ no multiple entries in any cell of the parsing table LL(1)

→ so, the given grammar is LL(1) parser.

(8) $S \to aBdh$

$B \to cC$

$C \to bC \mid \varepsilon$

$D \to Ef$

$E \to g \mid \varepsilon$

$f \to f \mid \varepsilon$

First(S) = a
First(B) = c
First(c) = {b/ε}
First(D) = {g, ε}
First(E) = {g, ε}
First(F) = {f, ε}

Follow(S) = {$}
Follow(B) = {h, h}
Follow(c) = {h}
Follow(E) = {g, h}
Follow(F) = {h}

(7) S → a | ab | abc | abcd

common prefix is a we can take out

S → a(ε | b | bc | bcd)

S → aS'

S' → ε | b | bc | bcd

(6) A → Ba | Aa | c

B → Bb | Ab | d

A → aB' | c A' → aA' | ε

B → bB'

B' → bB' | ε

B → bA'

A → aB' | c

B' → aB' | ε

A → aA'

A' → bA' | ε

B → d

(2) S → (L) | a BODMAS

L → L, S | S

	c	g	,	a	$
c	<	>	>	<	>
g	>	>	>	<	>
,	<	>	>	<	>
a	>	>	>	-	>

(3) $E \to E + E$
$E \to E * E$
$E \to id$

id + id * id

Stack	Ip buffer	Action
$	id + id * id $	Shift + id
$ id	+ id * id $	Shift +
$ id +	id * id $	Shift id
$ id + id	* id $	$E \to id$
$ E + id	* id $	$E \to id$
$ E + E	* id $	Shift *
$ E + E *	id $	Shift id
$ E + E * id	$	$E \to id$
$ E + E * E	$	$E \to E * E$
$ E + E	$	$E \to E + E$
$ E	$	Ip is accepted

(4) $S \to AS \mid b$
$A \to SA \mid a$

0 $S' \to .S$
1 $S \to .AS$
2 $S \to .b$
3 $A \to .SA$
4 $A \to .a$

I_1: go to (I_0, S):
$S' \to S.$
$S \to A.S$
$S \to .AS$
$S \to .b$

I_2: go to (I_0, A):
$A \to S.A$
$A \to .SA$
$A \to .a$
$A \to a.$

I_3 : go to $(I_1, \cdot)$

$S \rightarrow AS.$

I_4 : goto $(I_2, \cdot)$

$A \rightarrow SA.$

Follow of $S, A)$

$S = \{\$, A\}$,

$A = \{S, A, \$\}$

$I_1 \rightarrow S' \rightarrow S. \quad S \rightarrow b.$

$I_2 \rightarrow A \rightarrow a.$

$I_3 \rightarrow .S \rightarrow AS.$

$I_4 \rightarrow A \rightarrow SA.$

State	action			goto	
	$\$$	ab		AS	
0		I_3	I_4	I_1	I_2
1	acc.				
2		I_3	I_4	I_5	
3		I_3	I_4	I_6	
4	r_1	r_3	r_4		
5					
6					

(8) $S \rightarrow a \mid \uparrow \mid (L)$ BODMAS

$A \rightarrow SA \mid a$

	a	↑	(	)	$\$$
a	−	>	>	>	>
↑	<	−	<	>	>
(	<	>	>	>	>
)	<	>	>	>	>

(a, a) → input

Stack	Procedure	IP	action
$	<	(a,a)	shift c
$ (	<	a,a)	shift a
$ (a	>	,a)	shift ,
	↓		
	until the greater than symbol is eliminated		