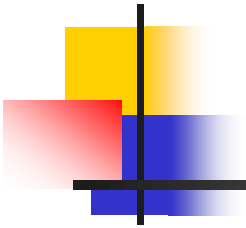
CSC 3201 Compiler Construction

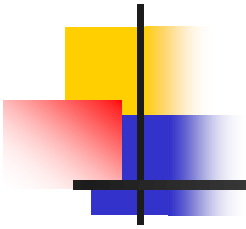
Department of Computer Science
SZABIST (Islamabad Campus)

Week 6 (Lecture 2)



LL(1) Parsing

- The 1st L represents that the scanning of the Input will be done from Left to Right manner.
- The 2nd L shows that Left most Derivation will be used.
- The number 1 represents the number of look ahead.
  - Look Ahead means how many symbols will be seen to make a decision.



LL(1) Parsing Table

- Two functions:
  - First(): The beginning Terminal Symbol if from a variable all strings are derived.
  - Follow(): The Terminal Symbol which follows a variable in the process of derivation.
- After computing the First and Follow set for each Non-Terminal symbol we have to construct the Parsing table.
  - Rows contain the NTs.
  - Column contain the Ts.

LL(1) Parsing Table

- Example 1:

E → TE'

E' → +TE' | ε

T → FT'

T' → *FT' | ε

F → id | (E)

LL(1) Parsing Table

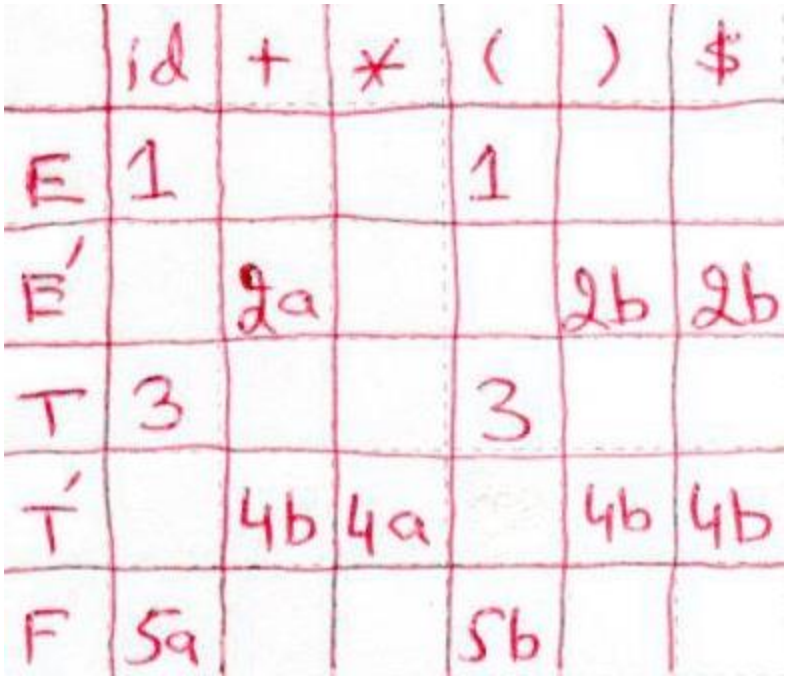
Production	First	Follow
E –> TE’	{ id, (}	{ $,) }
E’ –> +TE’ | ε	{ +, ε }	{ $,) }
T –> FT’	{ id, (}	{ +, $,) }
T’ –> *FT’ | ε	{ *, ε }	{ +, $,) }
F –> id | (E)	{ id, (}	{ *, +, $,) }

LL(1) Parsing Table

	First	Follow	Production No
$E \rightarrow TE'$	{id, (}	{$,)}	1
$E' \rightarrow +TE' \mid \varepsilon$	{+, ε}	{$,)}	2
$T \rightarrow FT'$	{id, (}	{+,), $}	3
$T' \rightarrow *FT' \mid \varepsilon$	{*, ε}	{+,), $}	4
$F \rightarrow id \mid (E)$	{id, (}	{*, +,), $}	5

LL(1) Parsing Table

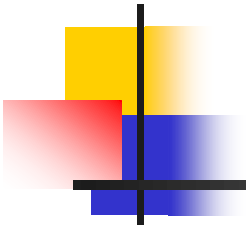
- Check First(LHS) in the production
- If it is ε then check Follow(LHS)



	id	+	*	(	)	$
E	1			1		
E′		2a			2b	2b
T	3			3		
T′		4b	4a		4b	4b
F	5a			5b		

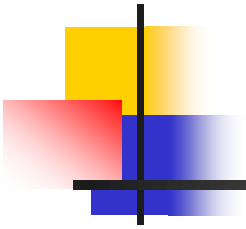
LL(1) Parsing Table

NON-TERMINAL	INPUT SYMBOL					
	id	+	*	(	)	$
E	$E \to TE'$			$E \to TE'$		
E'		$E' \to +TE'$			$E' \to \epsilon$	$E' \to \epsilon$
T	$T \to FT'$			$T \to FT'$		
T'		$T' \to \epsilon$	$T' \to *FT'$		$T' \to \epsilon$	$T' \to \epsilon$
F	$F \to \mathbf{id}$			$F \to (E)$		



LL(1) Parsing Table

- NOTE:
  - LL(1) Table cannot be constructed for left recursive and non-deterministic grammars.
    - Remove left recursion.
    - Ensure left factoring.
  - Sometimes valid grammars may not have LL(1) table. Example:
    - A → Multiple-Options (More than one productions go to the same cell).



LL(1) Parsing Table

- Example 2:

S --> A | a

A --> a

LL(1) Parsing Table

- Example 2:

S → A | a

A → a

Production	First	Follow
S → A | a	{a}	{$}
A → a	{a}	{$}

LL(1) Parsing Table

Production	First	Follow
S → A | a	{a}	{$}
A → a	{a}	{$}

	A	$
S	S → A S → a	
A	A → a	

- Language is not LL(1)

LL(1) Parsing Table

Production	First	Follow
S → A | a	{a}	{$}
A → a	{a}	{$}

LL(1) Parsing Table

- Example 3:

E → ABC

A → a | Cb | ε

B → c | xA | ε

C → y | z

LL(1) Parsing Table

First(E) = {a,y,z,ε}

Frist(A) = {a,y,z,ε}

First(B) = {c,x,ε}

First(C) = {y,z}

Follow(E)={$}

Follow(A)={c,x,y,z}

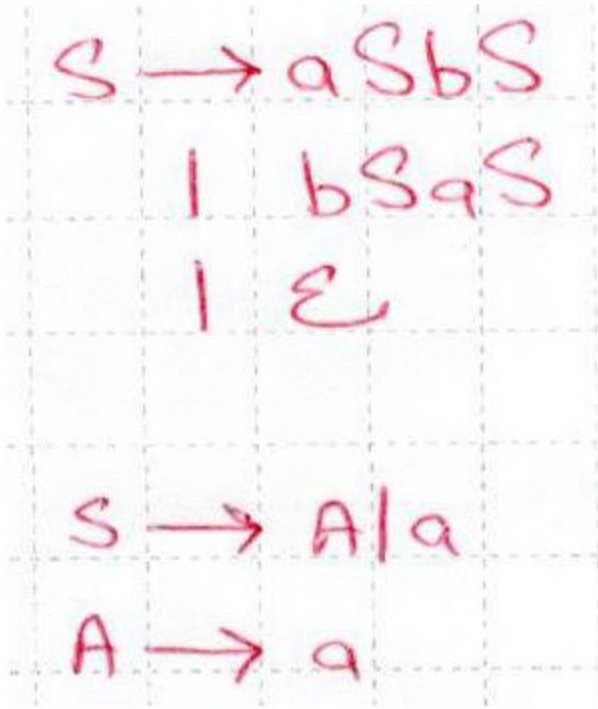
Follow(B)={y,z}

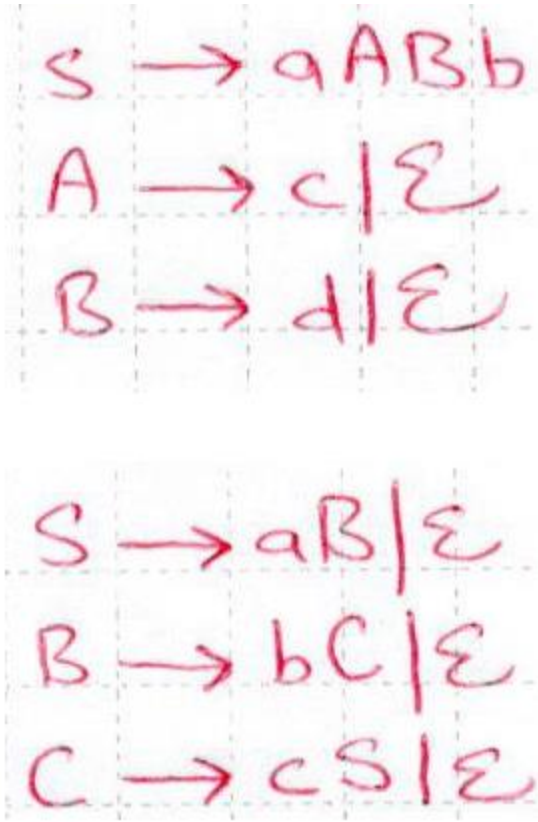
Follow(C)={b,$}

LL(1) Parsing Table

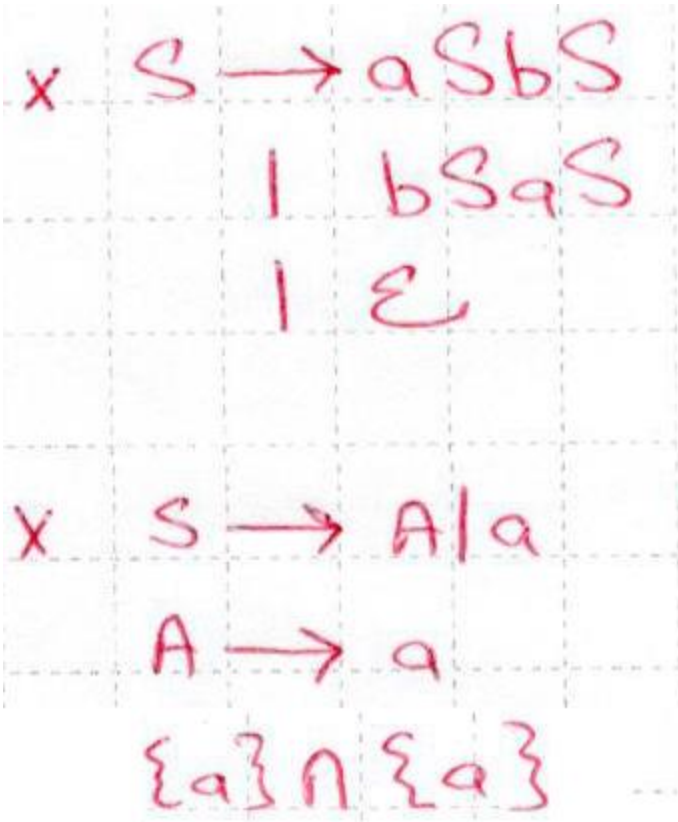
	x	a	b	c	y	z	$
E		ABC			ABC	ABC	ε
A	ε	a		ε	Cb	Cb	
B	xA			c	ε	ε	
C					y	z	

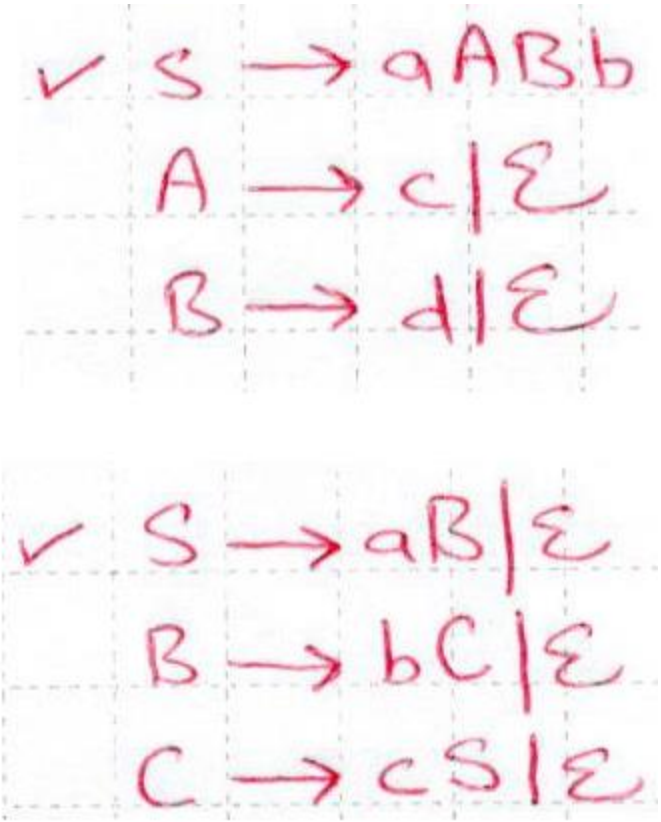
LL(1)?





LL(1)?





LL(1)?

$S \rightarrow AB$

$A \rightarrow a \mid \varepsilon$

$B \rightarrow b \mid \varepsilon$

$S \rightarrow iEtSS' \mid a$

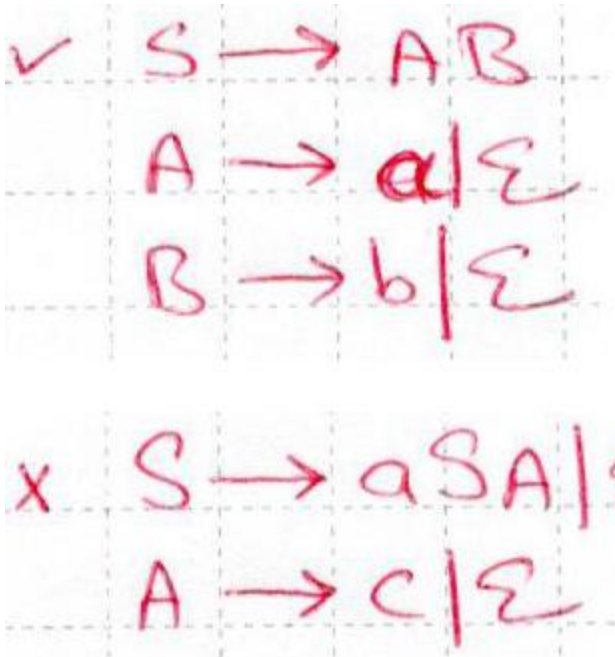
$S' \rightarrow eS \mid \varepsilon$

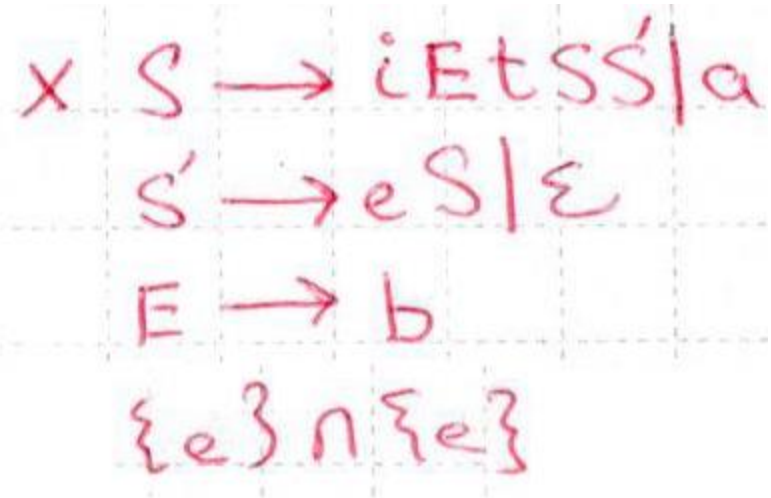
$E \rightarrow b$

$S \rightarrow aSA \mid \varepsilon$

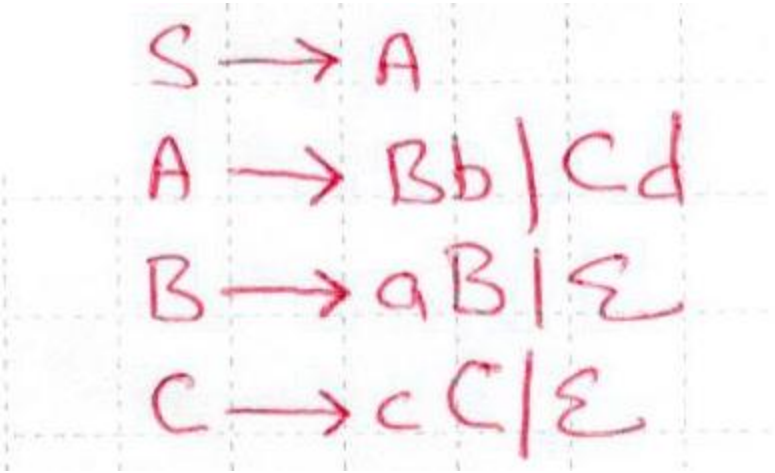
$A \rightarrow c \mid \varepsilon$

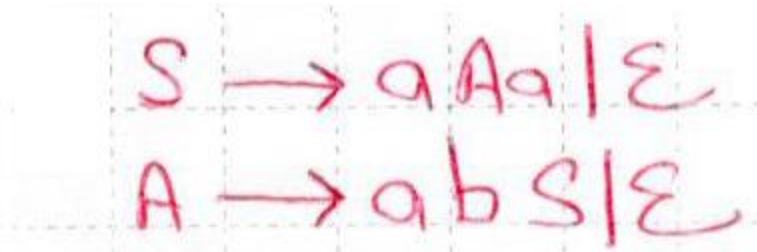
LL(1)?





LL(1)?





LL(1)?

✓ $S \rightarrow A$

$A \rightarrow Bb \mid Cd$

$B \rightarrow aB \mid \varepsilon$

$C \rightarrow cC \mid \varepsilon$

✗ $S \rightarrow aAa \mid \varepsilon$

$A \rightarrow abS \mid \varepsilon$