LL(1) Parsing - Solved Problems (set 1 & 2)

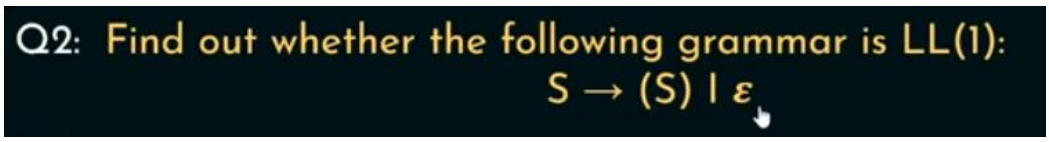
2 de junho de 2024 17:22

Five solved problems on whether the grammar is LL(1).

-	FIRST	FOLLOW
S -> aSbS | bSaS | ε	a, b, ε	b, a, $

-	a	b	$
S -> aSbS | bSaS | ε	S -> aSbS S -> ε	S -> bSaS S -> ε	S -> ε

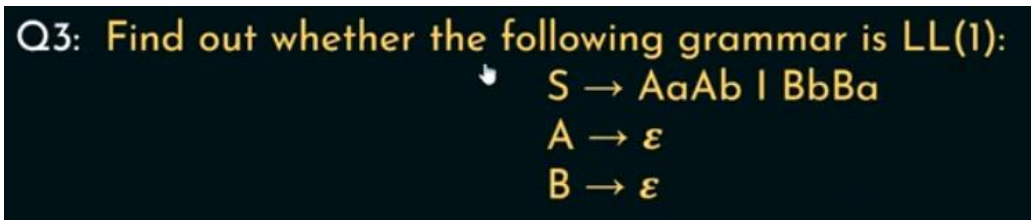
TWO DIFFERENT PRODUCTION RULES TO CHOOSE FROM
⇒ NOT LL(1) GRAMMAR



-	FIRST	FOLLOW
S -> (S) | ε	(, ε	), $

-	(	)	$
S -> (S) | ε	S -> (S)	S -> ε	S -> ε

⇒ LL(1) GRAMMAR



-	FIRST	FOLLOW
S -> AaAb | BbBa	a, b	$
A -> ε	ε	a, b
B -> ε	ε	b, a

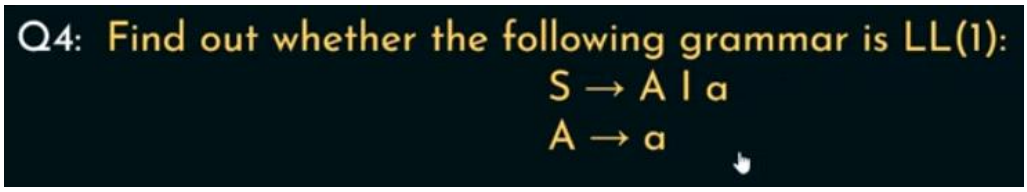
(none of the rows have multi-valued attributes)

-	a	b	$
S -> AaAb | BbBa	S -> AaAb	S -> BbBa	

-	a	b	$
S -> AaAb | BbBa	S -> AaAb	S -> BbBa	
A -> ε	A -> ε	A -> ε	
B -> ε	B -> ε	B -> ε	

multi-valued attributes)

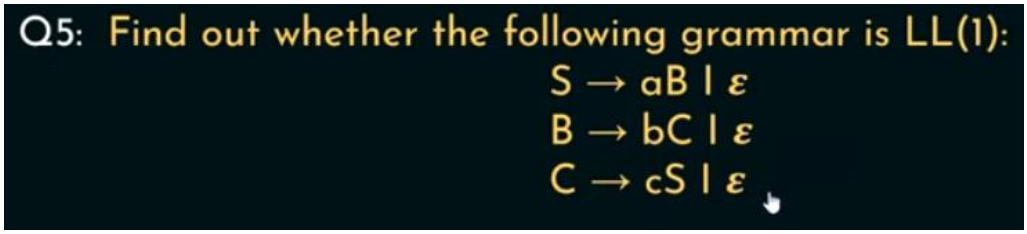
⇒ LL1 GRAMMAR



-	FIRST	FOLLOW
S -> A | a	a	$
A -> a	a	$

-	a	$
S -> A | a	S -> a S -> A	
A -> a	S -> a	

⇒ NOT LL(1) GRAMMAR



-	FIRST	FOLLOW
S -> aB | ε	a, ε	$
B -> bC | ε	b, ε	$
C -> cS | ε	c, ε	$

-	a	b	c	$
S -> aB | ε	S -> aB			S -> ε
B -> bC | ε		B -> bC		B -> ε
C -> cS | ε			C -> cS	C -> ε

⇒ LL(1) GRAMMAR

Set 2

Four solved problems on whether the grammar is LL(1).

Q1: Find out whether the following grammar is LL(1):

$S \rightarrow AB$
$A \rightarrow a \mid \varepsilon$
$B \rightarrow b \mid \varepsilon$

-	FIRST	FOLLOW
S -> AB	a, b, ε	$
A -> a | ε	a, ε	b, $
B -> b | ε	b, ε	$

-	a	b	$
S -> AB	S -> AB	S -> AB	S -> AB
A -> a | ε	A -> a	A -> ε	A -> ε
B -> b | ε		B -> b	B -> ε

⇒ LL(1)

Q2: Find out whether the following grammar is LL(1):

$S \rightarrow aSA \mid \varepsilon$
$A \rightarrow c \mid \varepsilon$

-	FIRST	FOLLOW
S -> aSA | ε	a, ε	$, c
A -> c | ε	c, ε	$, c

	a	c	$
S -> aSA | ε	S -> aSA	S -> aSA	S -> ε
A -> c | ε		A -> c A -> ε	A -> ε

FOLLOW(A) = FOLLOW(S)

a [a S A] A ... c

in this case, c follows A

NOT LL(1)

Q3: Find out whether the following grammar is LL(1):

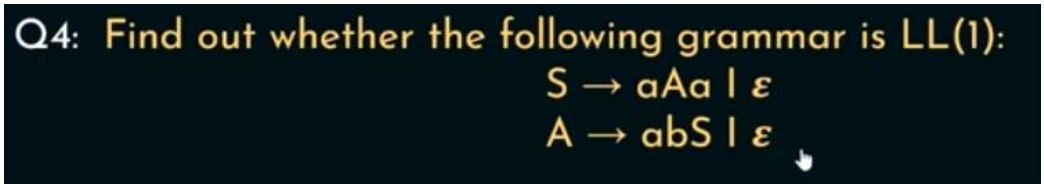
$S \rightarrow A$
$A \rightarrow Bb \mid Cd$
$B \rightarrow aB \mid \varepsilon$
$C \rightarrow cC \mid \varepsilon$

-	FIRST	FOLLOW
S -> A	a, b, c, d	$
A -> Bb | Cd	a, b, c, d	$

B -> aB | ε	a, ε	b
C -> cC | ε	c, ε	d

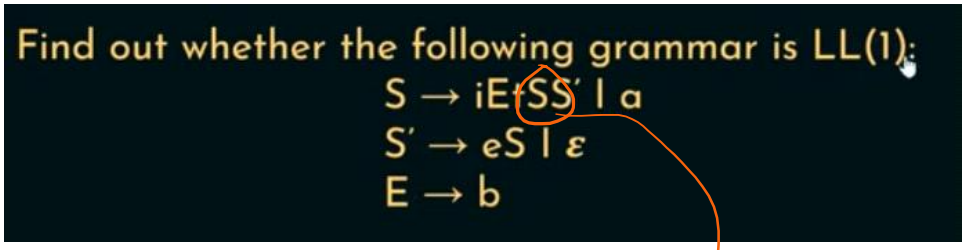
-	a	b	c	d	$
S -> A	S -> A	S -> A	S -> A	S -> A	
A -> Bb | Cd	A -> Bb	A -> Bb	A -> Cd	A -> Cd	
B -> aB | ε	B -> aB	B -> ε			
C -> cC | ε			C -> cC	C -> ε	

⇒ LL(1)



-	FIRST	FOLLOW
S -> aAa | ε	a, ε	$, a
A -> abS | ε	a, ε	a

→ Repeated
⇒ NOT LL(1)



-	FIRST	FOLLOW
S -> iE+SS' | a	i, a	$
S' -> eS | ε	e, ε	$
E -> b	b	+

Apresenta recursividade à esquerda
↳ ciclo infinito

a	i	a	e	b	$
S	S -> iE+SS'	S -> a			
S'			S' -> eS		S' -> ε
E				E -> b	

FOLLOW(S) = FOLLOW(S')
FOLLOW(S') = FOLLOW(S)

⇒ NOT LL(1) GRAMMAR