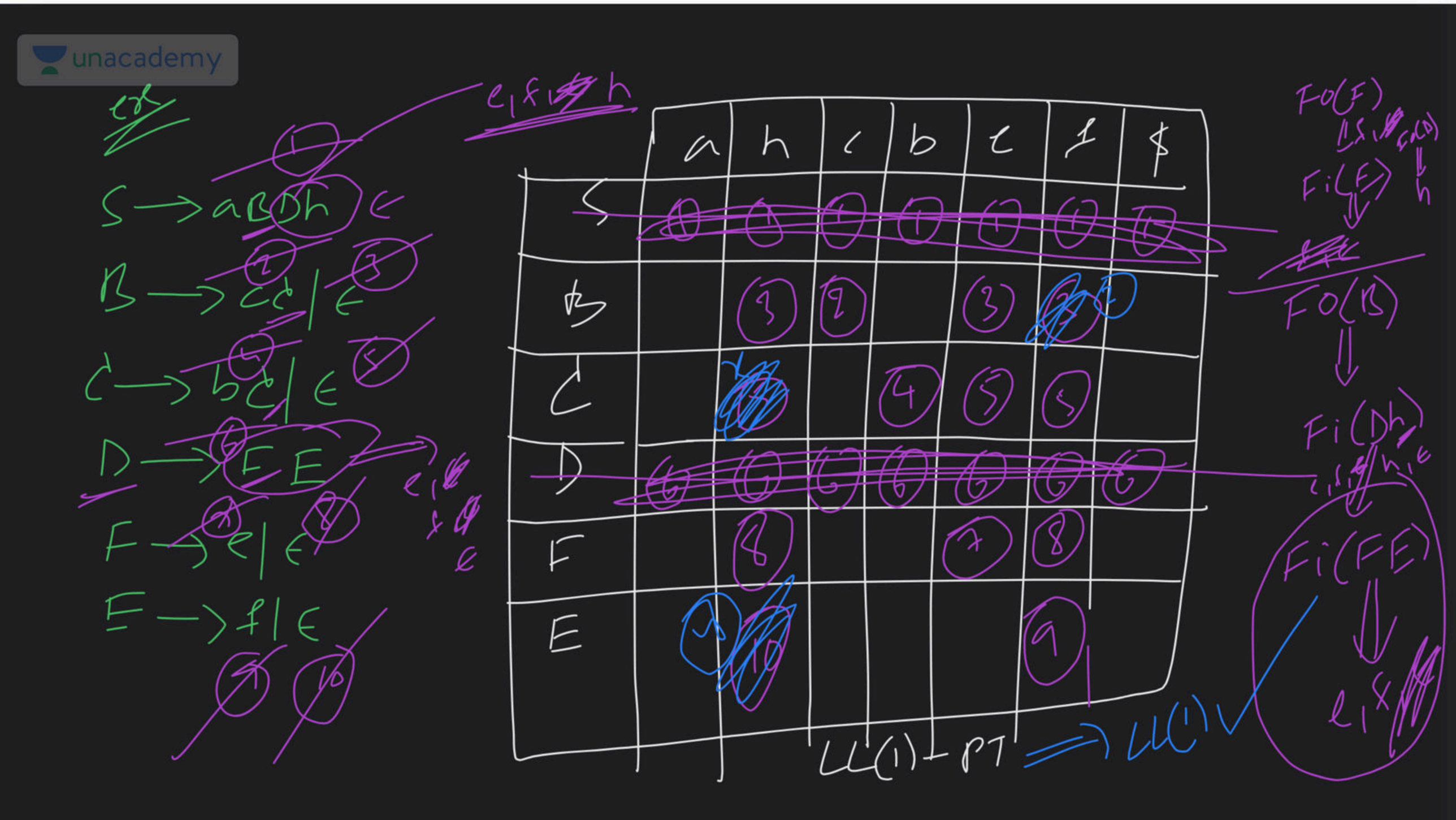
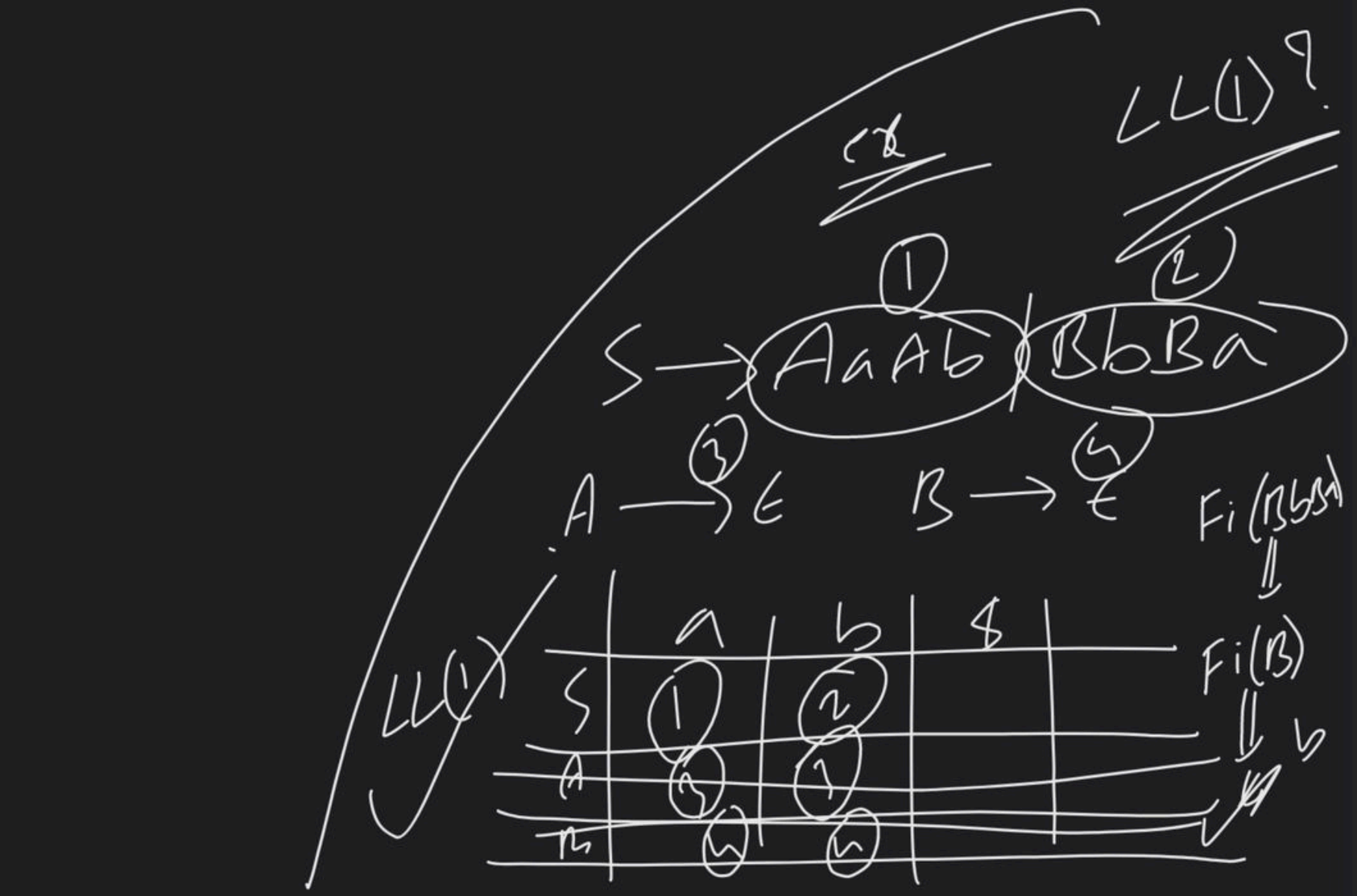
unacademy

Parsing-VI

Complete Course on Compiler Design

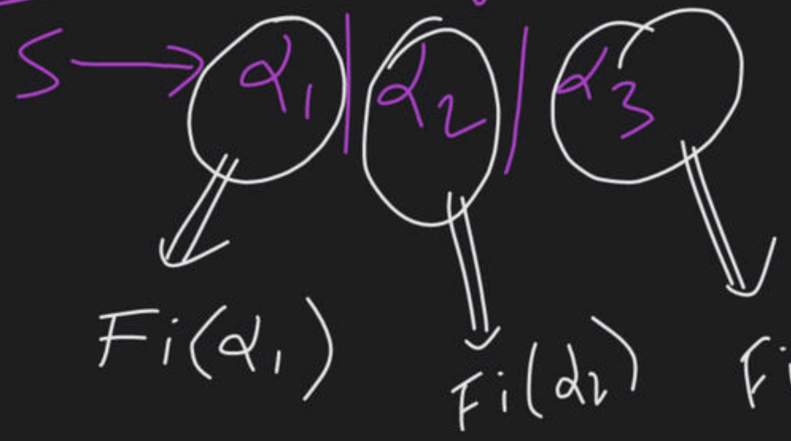
Subbarao Lingamgunta • Lesson 9 • Aug 25, 2023


unacademy
S → aBDh
B → cC | ε
C → bC | ε
D → EE
F → e | ε
E → f | ε
a b c b c d $
S
B
C
D
F
E
LL(1) - PT
LL(1)
Fi(E)
Fo(E)
Fo(B)
Fi(Dh)
Fi(FE)



How do check LL(1) [shortcut]

Note ① If grammar is not containing null productions



$$Fi(\alpha_1) \cap Fi(\alpha_2) = \phi$$
$$Fi(\alpha_2) \cap Fi(\alpha_3) = \phi$$
$$Fi(\alpha_1) \cap Fi(\alpha_3) = \phi$$

Pairwise Disjoint

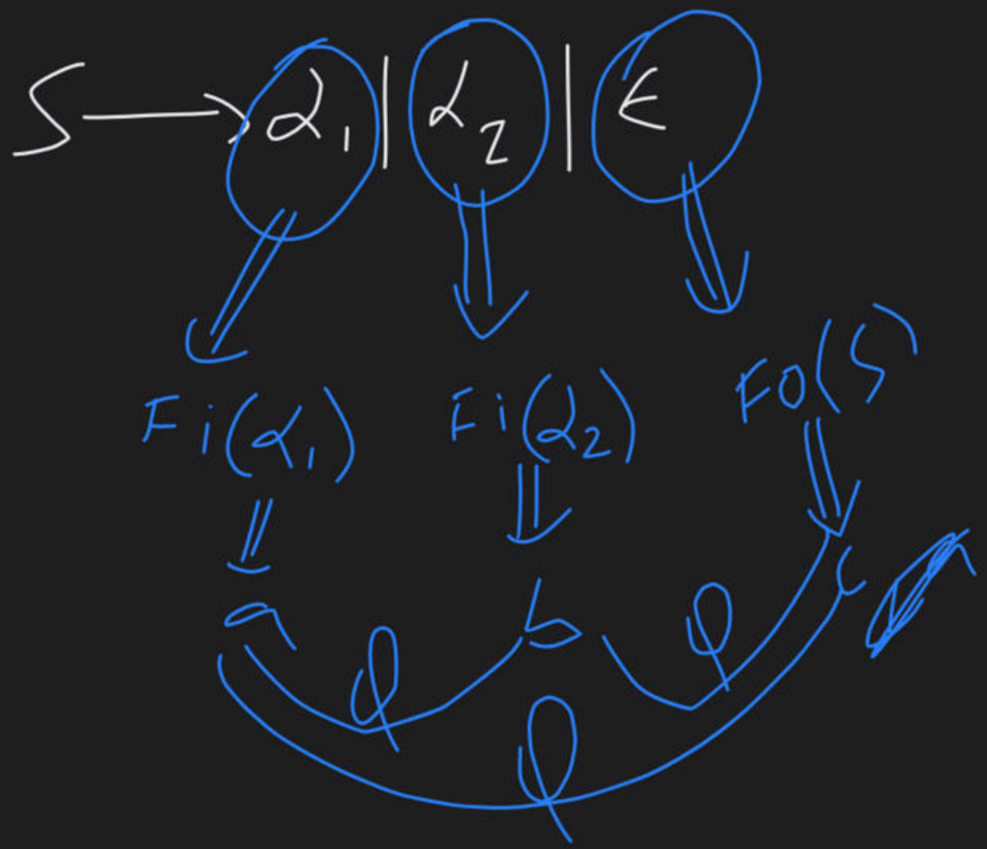
Then it is LL(1)

Note

unacademy

Note (2)

If a non-terminal contains null production



$Fi(\alpha_1) \cap Fi(\alpha_2) = \phi$

$Fi(\alpha_1) \cap FO(S) = \phi$

$Fi(\alpha_2) \cap FO(S) = \phi$

} pairwise Disjoint

Then it is LL(1)

Q

$S \rightarrow E \mid b$

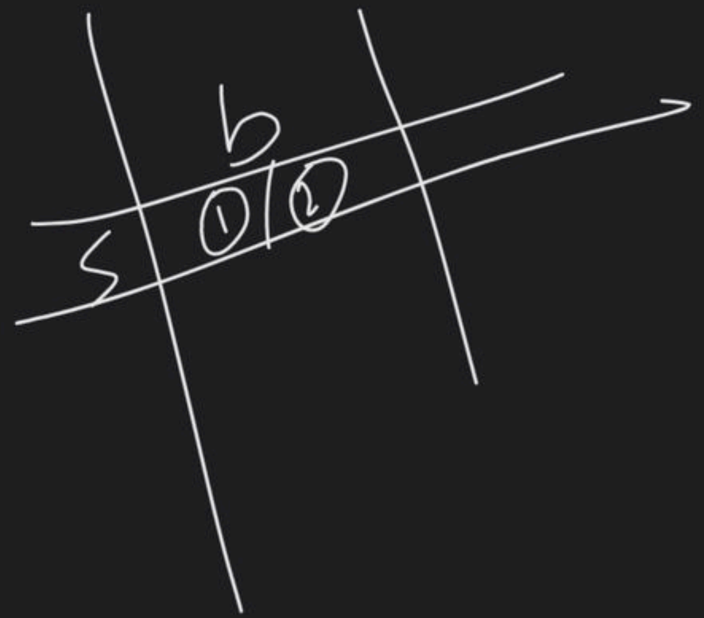
$E \rightarrow b$

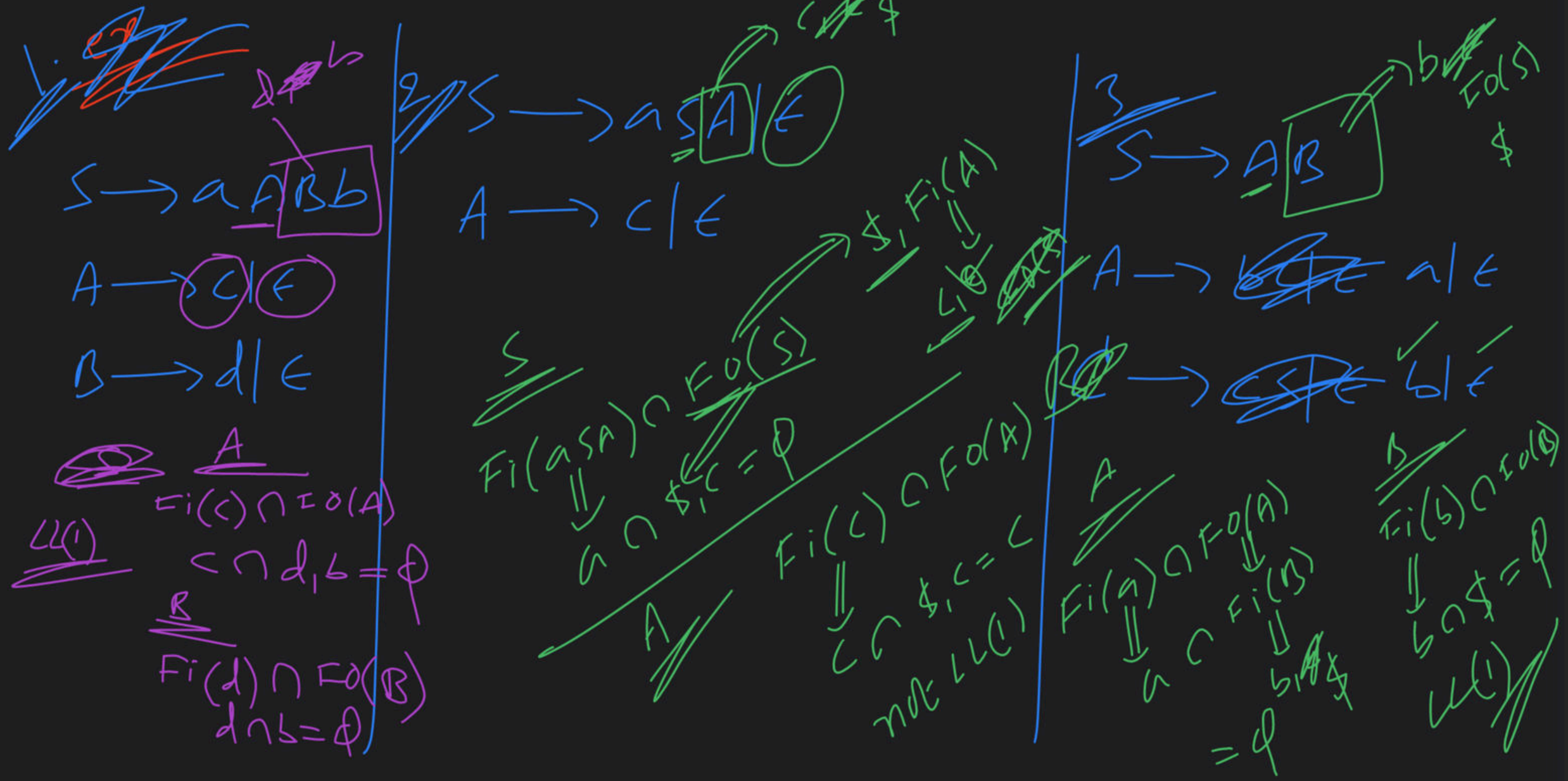
S

$Fi(E) \cap Fi(b)$

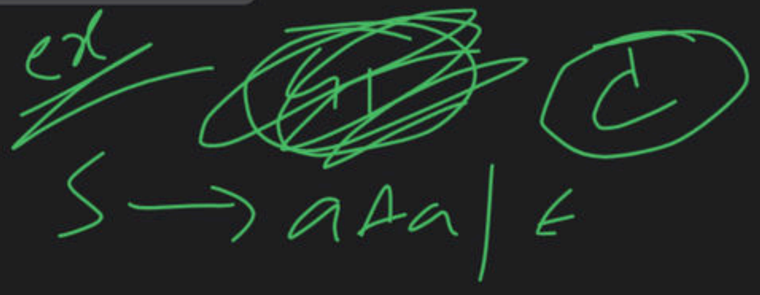
$b \cap b = b$

it is not LL(1)




S → aABb
A → c | ϵ
B → d | ϵ
A
Fi(c) ∩ Fo(A)
c ∩ d, b = ϕ
LL(1)
B
Fi(d) ∩ Fo(B)
d ∩ b = ϕ
S → aSA | ϵ
A → c | ϵ
S
Fi(aSA) ∩ Fo(S)
a ∩ $, c = ϕ
A
Fi(c) ∩ Fo(A)
c ∩ $, c = c
not LL(1)
S → A B
A → b | ϵ
B → c | ϵ
A
Fi(a) ∩ Fo(A)
a ∩ Fi(B)
b, $
= ϕ
B
Fi(b) ∩ Fo(B)
b ∩ $ = ϕ
LL(1)

ex



$S \rightarrow aAa \mid \epsilon$

$A \rightarrow ab \mid \epsilon$

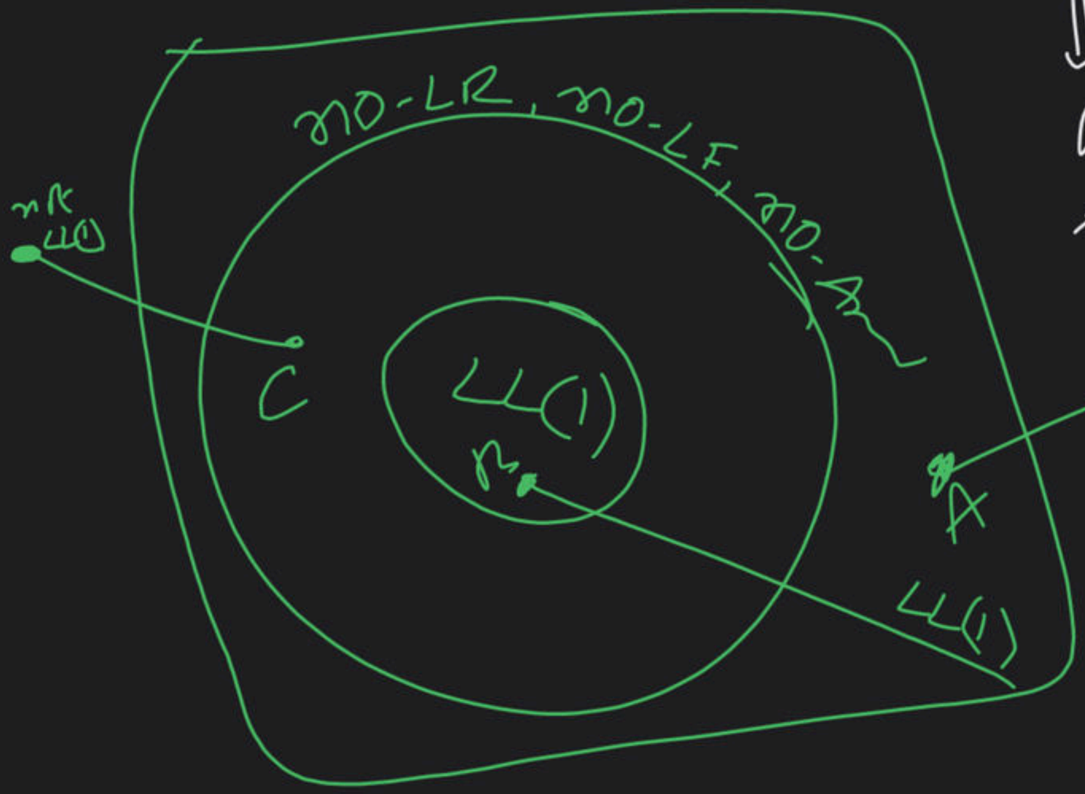
S

$Fi(aAa) \cap Fo(S)$

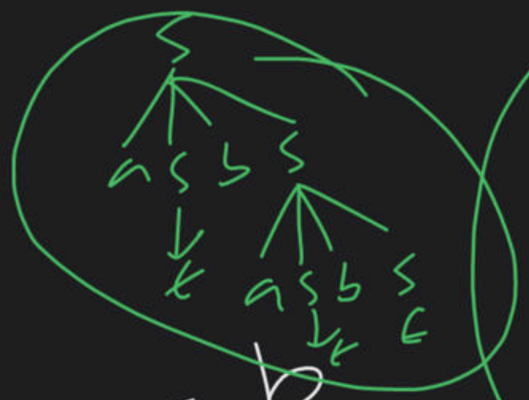
$a \cap \$, \; Fo(A) = a$

a

non LL(1)



$S \rightarrow Sa/b$



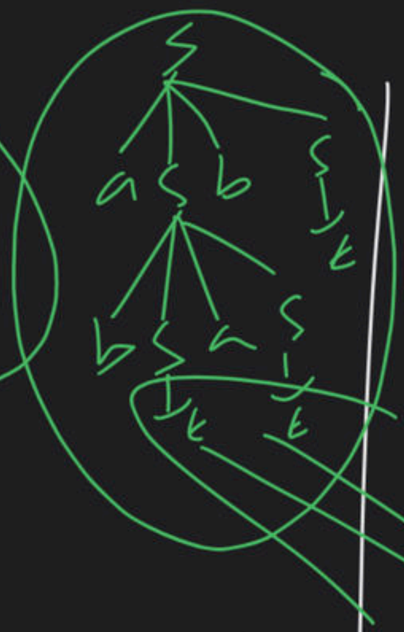
S

$Fi(Sa) \cap Fi(b) = b$

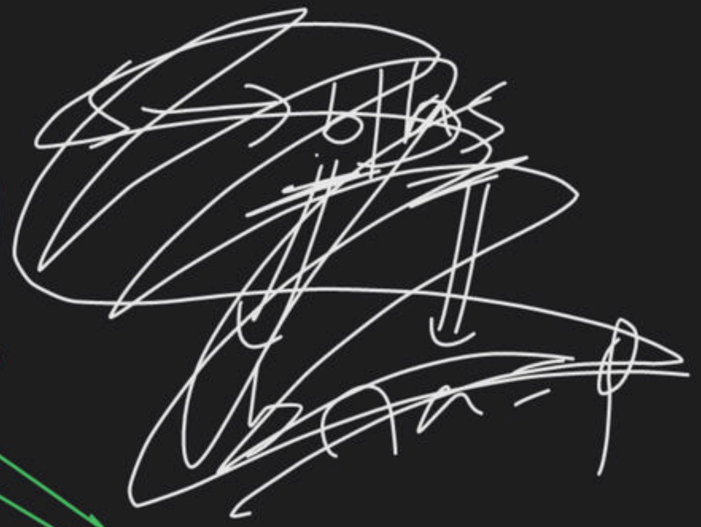
$Fi(S) \quad b$

$Fi(Sa), Fi(b)$

$Fi(S), b$

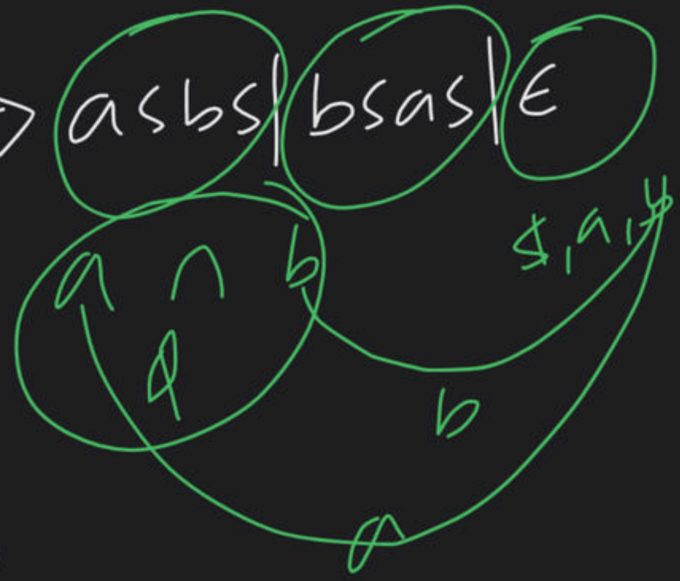


not LL(1)



aba

$S \rightarrow asbs/bsas/\epsilon$



not LL(1)

unacademy

Note ①

$S \rightarrow ab \mid ac$

S

$Fi(ab) \cap Fi(ac)$

$a \cap a = a$

Not LL(1)

$S \to aSa \mid bSb \mid a \mid b$

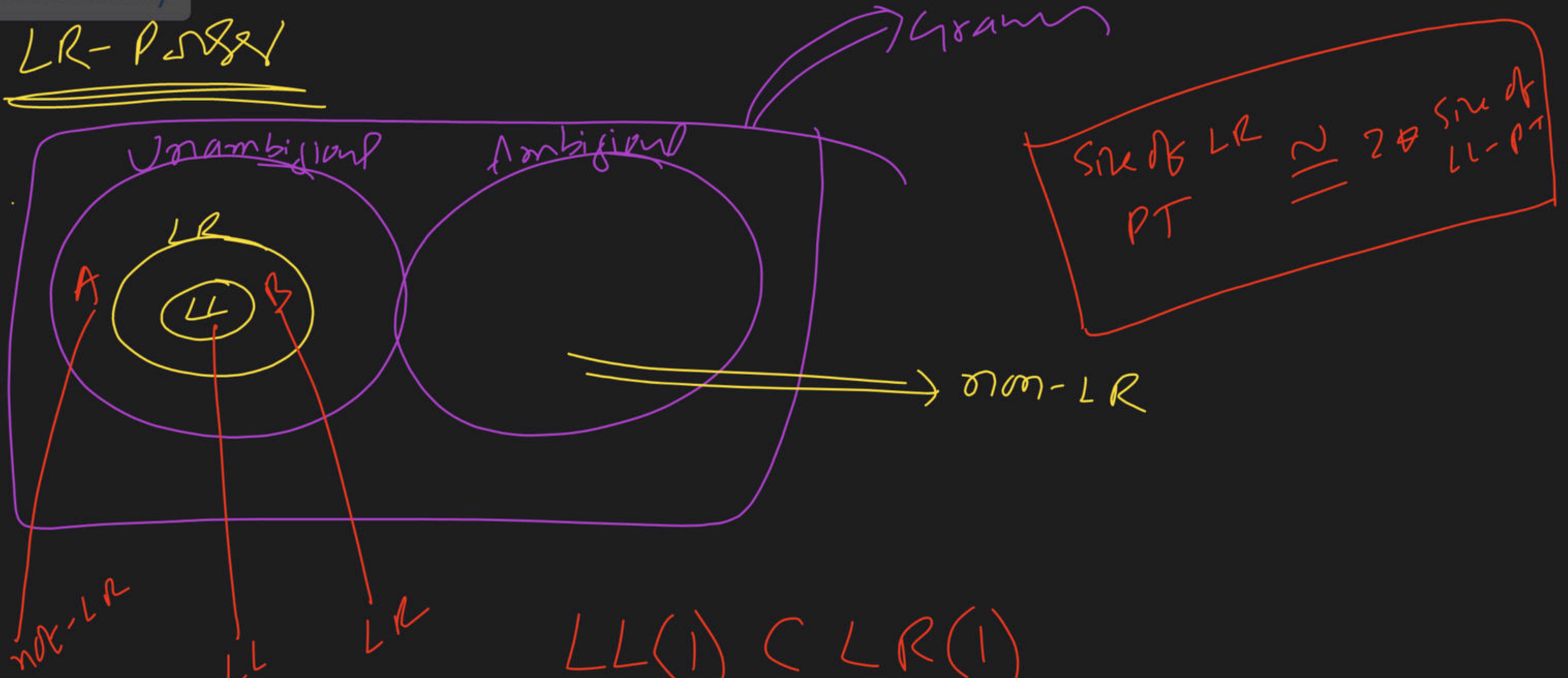
$S \to SaSa \mid b$

$S \to a \mid aS$

Bottom-Up-Parser (Without Backtracking ⇓ but Table is Huge.)

- LR-Parsers
  - LR(0)
  - SLR(1)
  - LALR(1)
  - CLR(1)
- Operator Precedence Parser.

LR-Parser



$LL(1) \subset LR(1)$

$LL(2) \subset LR(2)$

$\vdots$

$LL(k) \subset LR(k)$