CS & IT Engineering

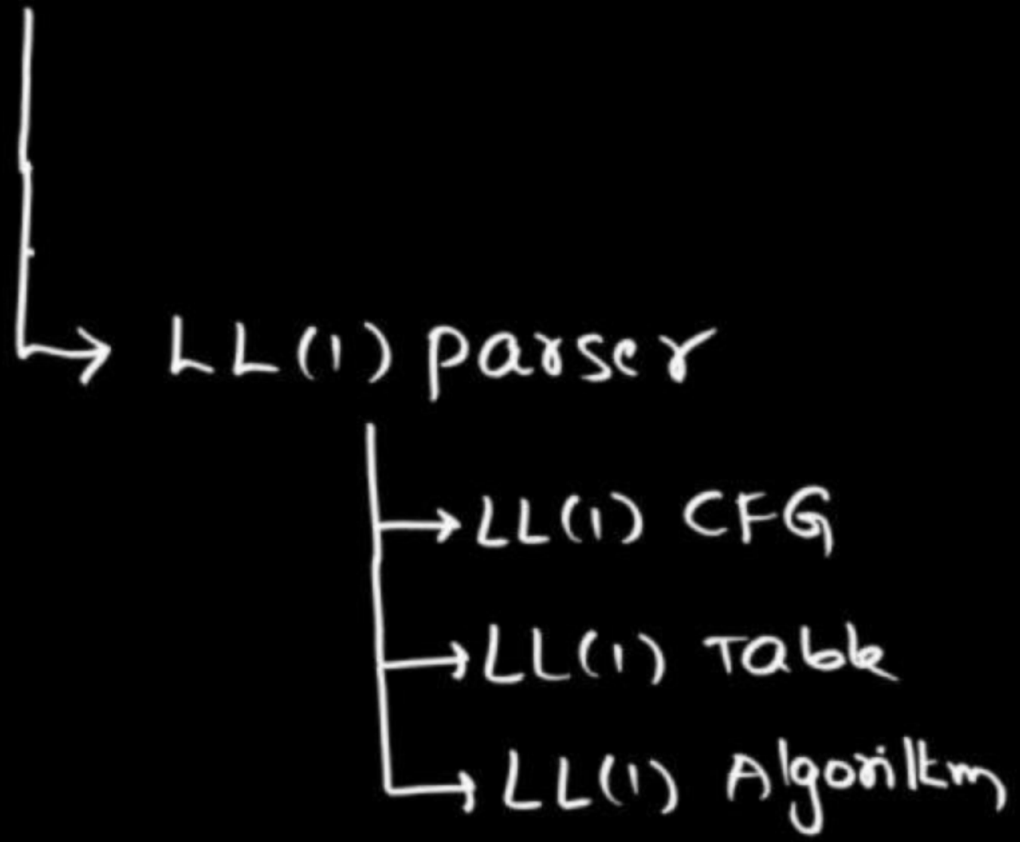
Compiler Design

Lexical Analysis and syntax Analysis

Lecture: 7

Deva sir

Topics to be covered:



(4)

$S \rightarrow AB$

$A \rightarrow Aa \mid \varepsilon$

$B \rightarrow Bb \mid \varepsilon$

$Fi(S) = \{a, b, \varepsilon\}$

$Fi(A) = \{a, \varepsilon\}$

$Fi(B) = \{b, \varepsilon\}$

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

$Fo(A) = \{b, \$, a\}$

$Fo(B) = \{b, \$\}$

	a	b	$
S	AB	AB	AB
A	Aa ε	ε	ε
B		Bb ε	ε

(5)

$S \rightarrow aAb$

$A \rightarrow Bb \mid \varepsilon$

$B \rightarrow ab$

$Fi(S) = \{a\}$

$Fi(A) = \{a, \varepsilon\}$

$Fi(B) = \{a\}$

$Fo(A) = \{b\}$

	a	b	$
S	S→aAb		
A	A→Bb	A→ε	
B	B→ab		

How to Identify LL(1) CFG ?

Sometimes Better — I) Theory

Best — II) Shortcut

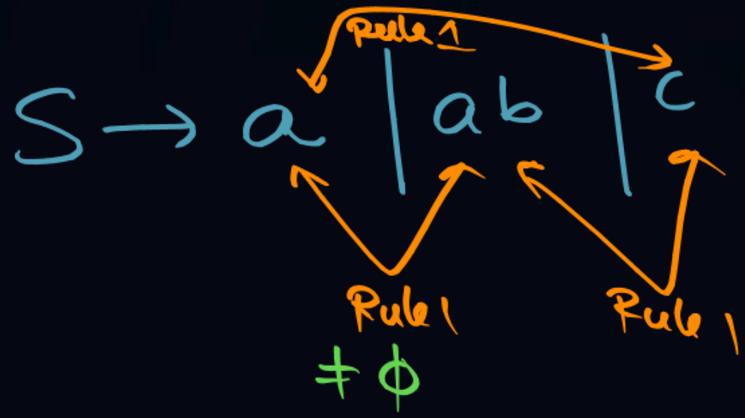
Lengthy — III) Table

Shortcut:

Rule 1: $X \rightarrow \alpha_1 \mid \alpha_2$

If $First(\alpha_1) \cap First(\alpha_2) \neq \phi$

then given CFG is not LL(1)



It is not LL(1)

Rule 2: $X \to \alpha \mid \varepsilon$

If $First(\alpha) \cap Follow(X)_{LHS} \neq \phi$

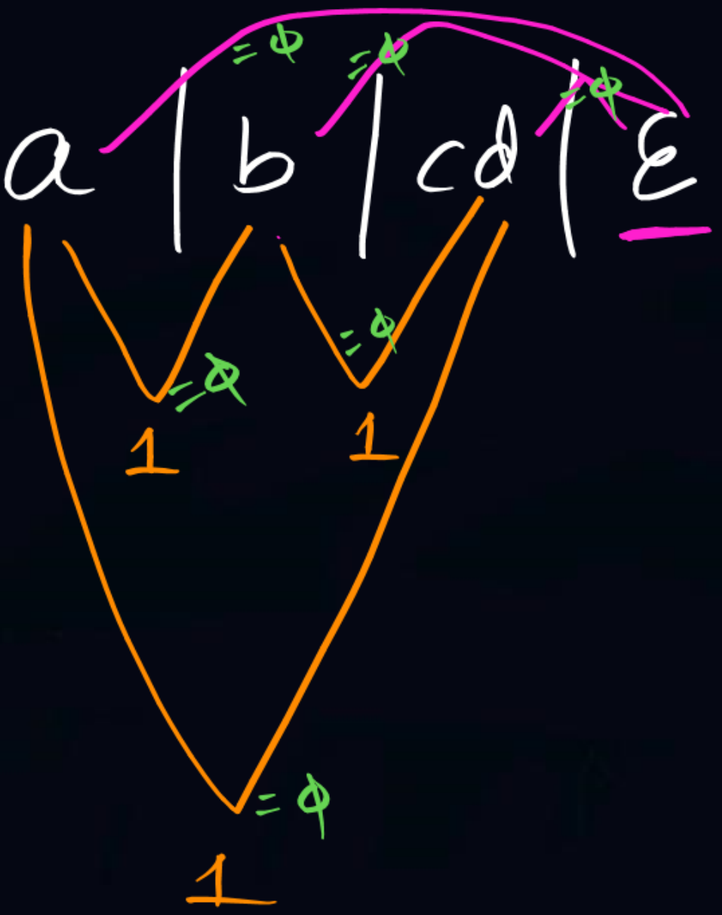
then given CFG is not LL(1)

$S \to aSb \mid \varepsilon$ It is LL(1)

$First(aSb) \cap Follow(S) = \phi$

$\{a\} \cap \{\$, b\} = \phi$

① $S \rightarrow a \mid b \mid cd \mid \underline{\varepsilon}$



It is LL(1)

② $S \rightarrow Sa \mid bS \mid c \mid d \mid \varepsilon$

↳ Not LL(1)

③ $S \rightarrow a \mid bc \mid aef$

↳ Not LL(1)

④ $S \to AB \mid ab$

$A \to a \mid \varepsilon$

$B \to b \mid \varepsilon$

not LL(1)

⑤ $S \to aAbB \mid bAaB \mid \varepsilon$

$A \to S$

$B \to S$

$F_i(aAbB) \cap F_i(bAaB)$

$\{a\} \cap \{b\} = \phi$

$\neq \phi$

$\neq \phi$

$Fo(S) = \{\$\} \cup Fo(A) \cup Fo(B)$

$= \{\$\} \cup \{a,b\} \cup Fo(S)$

$= \{\$, a, b\}$

not LL(1)

If CFG is "not LL(1)" then which is possible?

A) Ambiguous

B) Left Recursive

C) Non Left factored

D) Some Table entry may contain more than 1 production

E) A & B

F) A & C

G) A & D

H) B & C

I) B & D

J) C & D

K) A, B & C

L) A, B & D

M) B, C & D

N) A, B, C, & D

If CFG is "not LL(1)" then which is TRUE?

A) Ambiguous
B) Left Recursive
C) Non Left factored

not necessary

$S \to aSa \mid \varepsilon$

	a
S	$S \to aSa$, $S \to \varepsilon$

$\{a\} \cap \{\$, a\} = \{a\} \neq \phi$

guarantee ~~B)~~ some Table entry may contain more than 1 production

E) A & B
F) A & C
G) A & D
H) B & C
I) B & D
J) C & D
K) A, B & C
L) A, B & D
M) B, C & D
N) A, B, C, & D

Possible

⇕

Sometimes TRUE

TRUE

⇕

Always TRUE

If CFG is LL(1) then which is TRUE?

A) Unambiguous

B) Non Left Rec

C) Left factored

D) A, B, & C

E) Every entry of table has atmost 1 production

F) A, B, C, & E

G) B & C

H) C & E

I) B & E

J) A, B, E

CFG is LL(1) iff CFG is ____

A) Unambiguous

B) Non Left Rec

C) Left factored

D) A, B, & C

E) Every entry of table has atmost 1 production

F) A, B, C, & E

G) B & C

H) C & E

I) B & E

J) A, B, E

① $S \to aSa \mid bS \mid c$ is ____

$= \phi$

$\{a\} \cap \{b\} = \phi$ $\{b\} \cap \{c\} = \phi$

A) ~~LL(1)~~

B) Not LL(1)

(2)

$S \to a\overset{I}{A}bB \mid b\overset{II}{A}aB \mid \overset{III}{\varepsilon}$

$A \to \overset{IV}{S}$

$B \to \overset{V}{S}$

$|E_1| = 2$

$|E_2| = 2$

	a	b	$
S	I, III E_1	II, III E_2	III
A	.	.	.
B	.	.	.

What is E_1 and E_2?

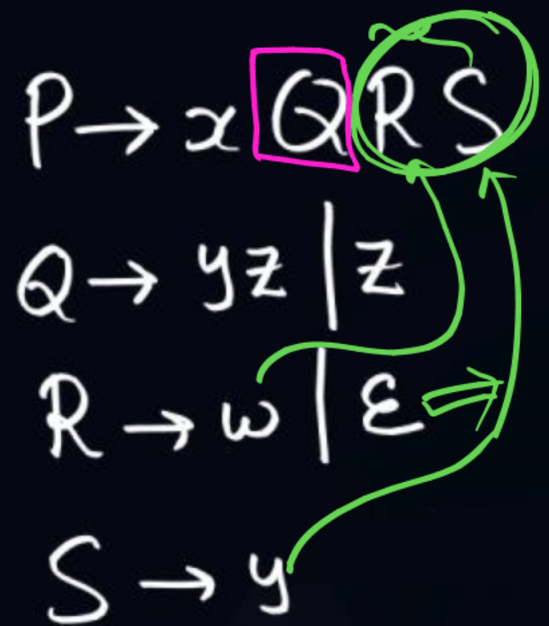
First(S) = {a, b, ε}

Follow(S) = {$, a, b}

E_1 = I, III = {S → aAbB, S → ε}

E_2 = II, III = {S → bAaB, S → ε}

(3)



Follow(Q) = ?

A. $\{\varepsilon\}$
B. $\{w\}$
C. $\{w, y\}$
D. $\{w, \$\}$

Follow(Q) = $\{w, y\}$

(4)

$E \to E-T \mid T$

$T \to T+F \mid F$

$F \to (E) \mid id$

Equivalent non-left recursive grammar is —

A. $E \to E-T \mid T$
$T \to T+F \mid F$
$F \to (E) \mid id$

B. $E \to TE'$
$E' \to -TE \mid \varepsilon$
$T \to T+F \mid F$
$F \to (E) \mid id$

C. $E \to TX$
$X \to -TX \mid \varepsilon$
$T \to FY$
$Y \to +FY \mid \varepsilon$
$F \to (E) \mid id$

D. $E \to Tx \mid (Tx)$
$X \to -Tx \mid +Tx \mid \varepsilon$
$T \to id$

(5)

$S \to daT \mid Rf$

$T \to aS \mid baT \mid \varepsilon$

$R \to caTR \mid \varepsilon$

	a	b	c	d	f	$
S			① RF		② RF	
T			③ T→ε			④ T→ε
R						

$First(S) = \{d, c, f\}$ ✓

$First(T) = \{a, b, \varepsilon\}$

$Follow(T) = \{\$, c, f\}$

$Fo(S) = Fo(T) \cup \{\$\}$

$Fo(R) = \{f\} \cup Fo(R)$

$S \to daT$: $Fo(T) = Fo(S) = Fo(T) \cup \{\$\}$

$T \to baT$: ✗

$R \to caTR$: $\{c\} \cup Fo(R) \cup \{f\}$

(5)

$S \to daT \mid Rf$

$T \to aS \mid baT \mid \varepsilon$

$R \to caTR \mid \varepsilon$

	a	b	c	d	f	$
S			① RF		② RF	
T			③ T→ε			④ T→ε
R						

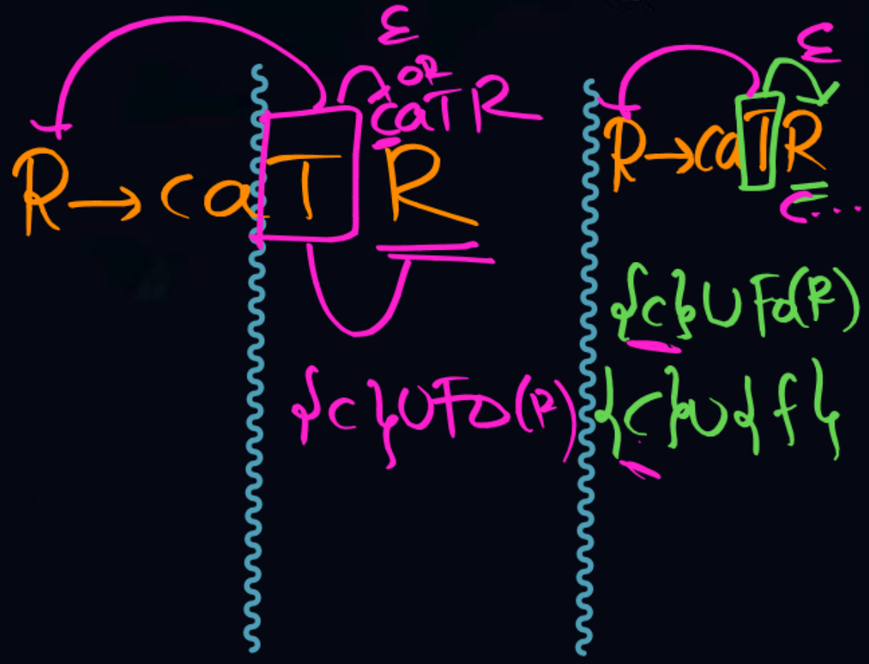
$First(S) = \{d, c, f\}$ ✓

$First(T) = \{a, b, \varepsilon\}$

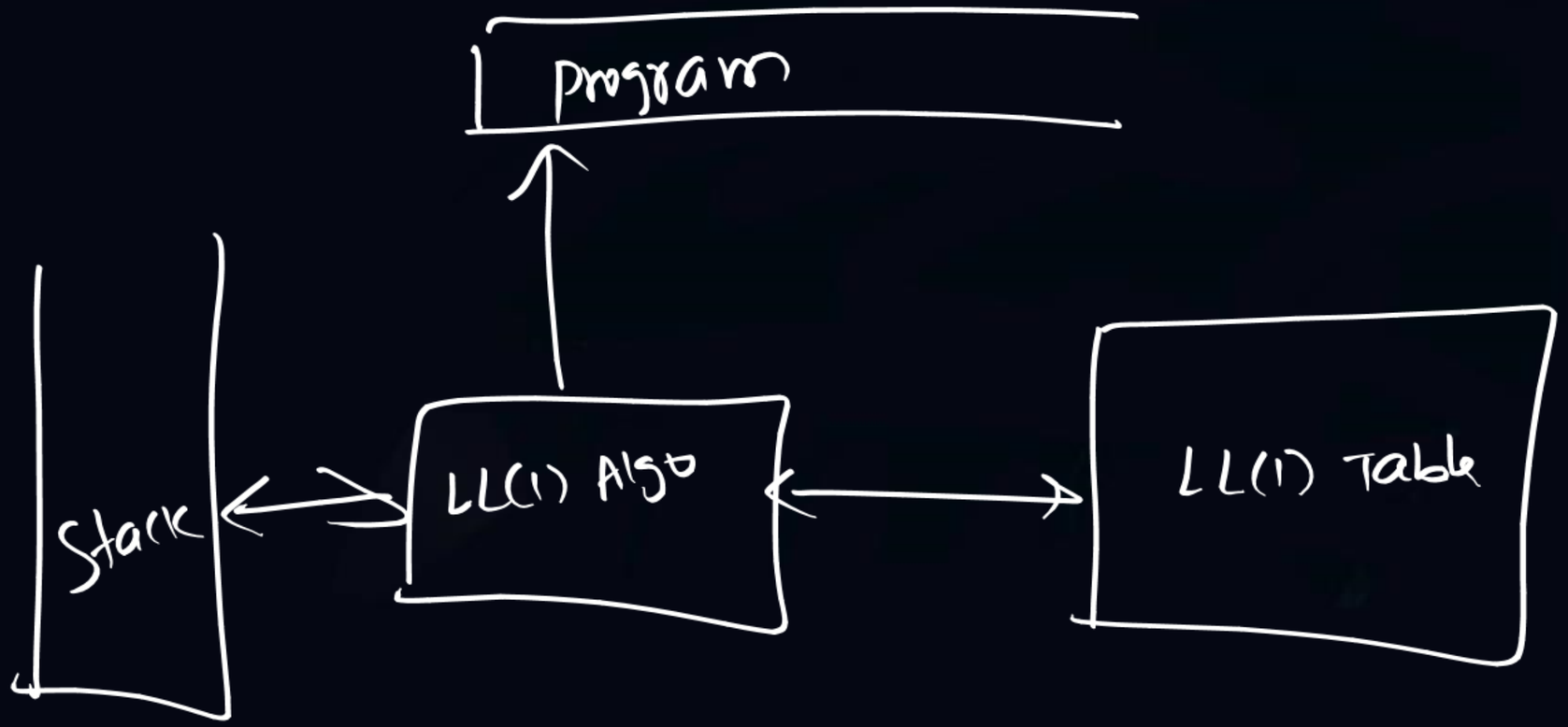
$Follow(T) = \{\$, c, f\}$

$Fo(S) = Fo(T) \cup \{\$\}$

$Fo(R) = \{f\} \cup Fo(R)$



LL(1) Parsing Algorithm:



Input: a b $

Given Table

M	a	b	$
S	$S \to aSb$	$S \to \varepsilon$	$S \to \varepsilon$

I) PREDICTION

If tos = Nonterminal

then M[tos, input] : LHS → RHS

i) pop tos (LHS)

ii) PUSH RHS in reverse order

II) Match

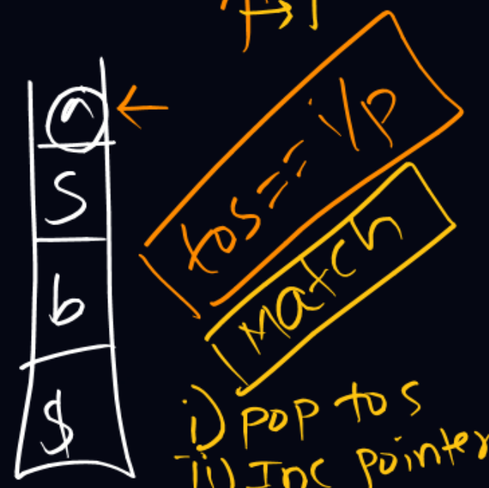
TOS == i/p

i) Pop tos

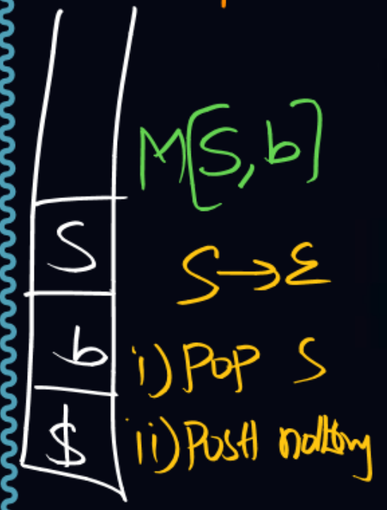
ii) Inc pointer

Step1: a b $

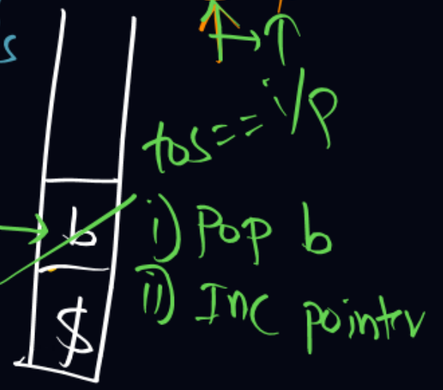
Step2: a b $



Step3: b $



Step4: b $



Step5: $

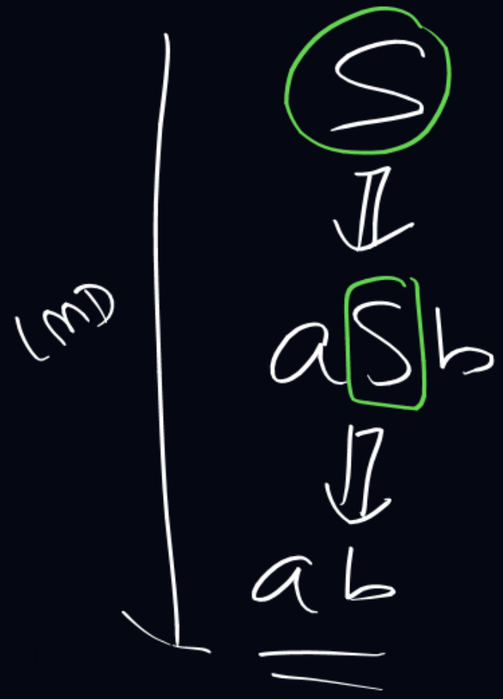
tos == i/p == $

ACCEPT

(No syntax error)

$S \rightarrow aSb \mid \varepsilon$

ab



What is max size of stack by assuming initially $ and S are already on stack)

= 4

Conflicts during LL(1) parsing: (errors)

Syntax Error

I) N- conflict:
If tos is Nonterminal and Table has blank entry, then N- conflict

II) T- conflict
If tos is terminal and tos ≠ i/p
then T- conflict

M	t
X	$X \to \alpha$

If tos == X

then what will happen in LL(1) parser ?

PREDICTION (Substitution in LMD)

I) Pop X

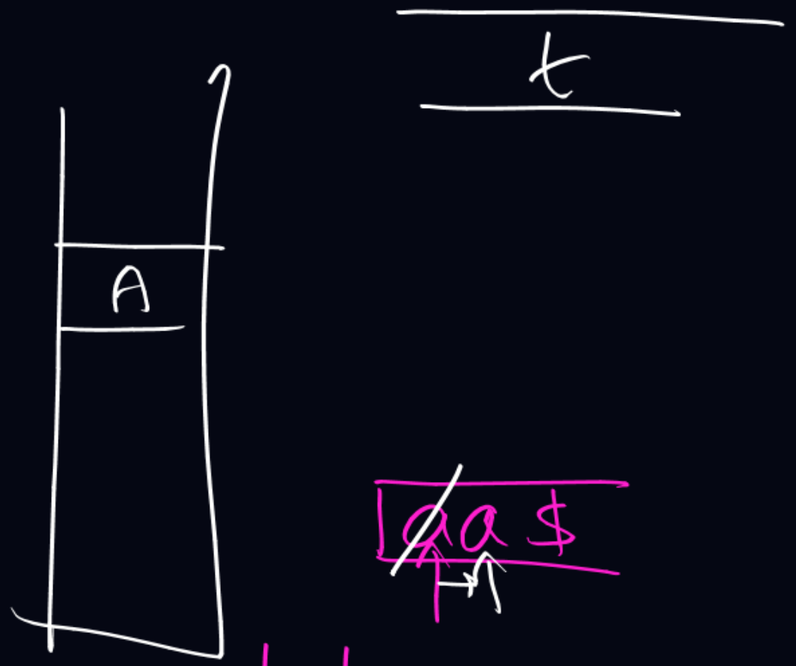
II) Push α in reverse order

Why to compute Follow ?

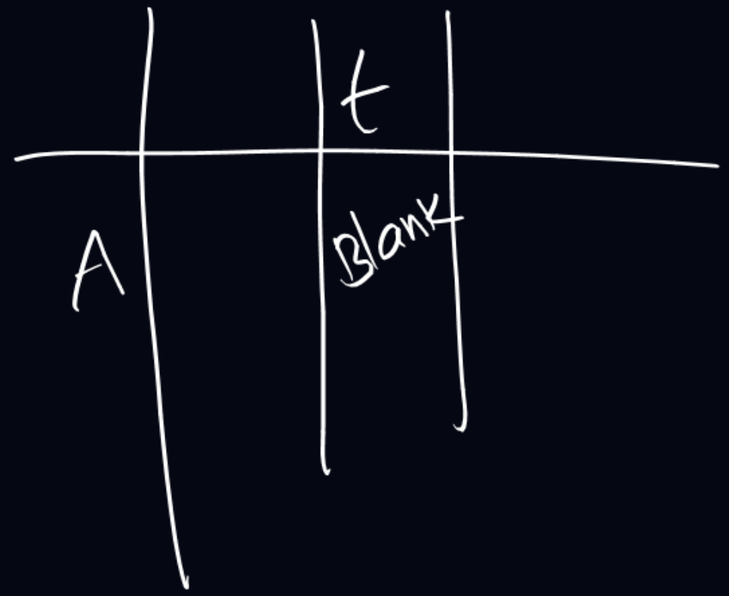
	t
A	A→ε

A → ε

N-conflict

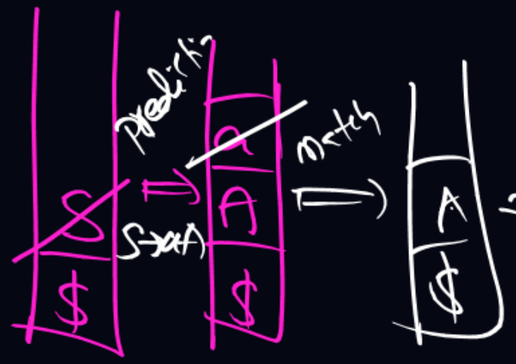


	a	b
S	S→aA	
A	Blank	A→b



S→aA

A→b



M[A,a] = Blank

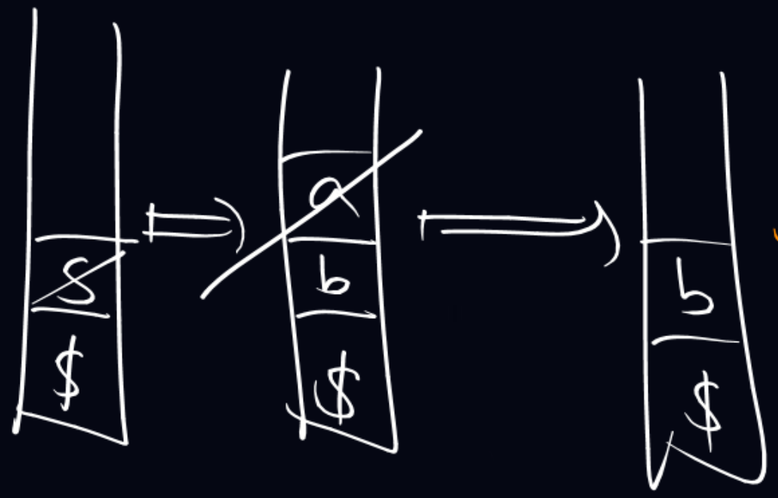
Error

N-conflict

T-conflict

$S \rightarrow ab$

	a	b
S	S→ab	



$b \neq a$

T-conflict

Summary

- Syntax Error ✓
- CFG Basics ✓
- *** Amb & Unamb CFG ✓
- Elimination of Left Rec ✓
- Left factoring ✓
- *** First & Follow ✓
- LL(1) Table ✓
- LL(1) CFG ✓
- LL(1) Algo ✓


THANK YOU GW
SOLDIERS !