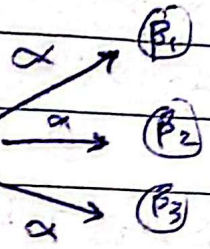
Date _____ Day _____

• Non-Deterministic CFGs

$A \to \alpha\beta_1 \mid \alpha\beta_2 \mid \alpha\beta_3$ → here α & β can be terminals / non-terminals or combination of both

or

$A \to \alpha\beta_1$
$A \to \alpha\beta_2$ ⟹
$A \to \alpha\beta_3$



↳ ek symbol par teen transition

→ Due to non-determinism the compiler needs to do a lot of backtracking

↳ Top-Down parsers tend to suffer from NCFGs

↳ Using Left Factoring procedure we need to eliminate the non-determinism

jitni ek production krdenge

• $A \to \underbrace{\alpha}_{\alpha}\underbrace{AB}_{\beta_1} \mid \underbrace{\alpha}_{\alpha}\underbrace{Bc}_{\beta_2} \mid \underbrace{\alpha}_{\alpha}\underbrace{Ac}_{\beta_3}$

$A \to \alpha A'$

$A' \to AB \mid Bc \mid Ac$

↳ it is still non-deter

$A \to \alpha\beta_1 \mid \alpha\beta_2 \mid \alpha\beta_3$

↳ $A \to \alpha A'$

$A \to \beta_1 \mid \beta_2 \mid \beta_3$

↳ $A' \to AA'' \mid Bc$

$A'' \to B \mid \epsilon$

Determinism cannot remove ambiguity

• $S \to iEtS \mid iEtSeS \mid a$

$E \to b$

$S \to iS' \mid a$

$S' \to EtS \mid EtSeS$

$S \to iEtSS' \mid a$

$S' \to \epsilon \mid eS$

 E → b

Date _____ Day _____

- $S \to aSSbS \mid aSaSb \mid abb \mid b$

  $S \to aS' \mid b$

  $S' \to SSbS \mid SaSb \mid bb$

  $S' \to SS'' \mid bb$

  $S'' \to SbS \mid aSb$

- $S \to bSSaaS \mid bSSaSb \mid bSb \mid a$

  $S \to bSS' \mid a$

  $S' \to SaaS \mid SaSb \mid b$

  $S' \to SaS'' \mid b$

  $S'' \to aS \mid Sb$

- $S \to a \mid ab \mid abc \mid abcd$

  $S \to aS'$

  $S' \to \epsilon \mid b \mid bc \mid bcd$

  $S' \to bS'' \mid \epsilon$

  $S'' \to \epsilon \mid c \mid cd$

  $S'' \to cS''' \mid \epsilon$

  $S''' \to \epsilon \mid d$

Date _____ Day _____

PARSERS (Syntax Analyzers)

"A parser is a program that generates a parse tree for the given string, if the string is generated from the underlying grammar

→ Generation of Parse Tree

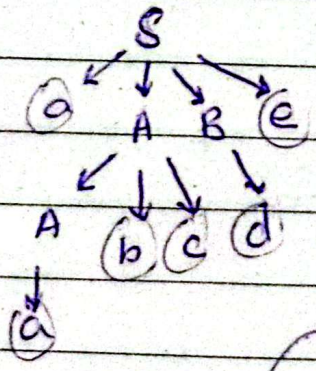
↳ There are two approaches for tree generation

① Top-Down Approach → Top Down Parser

② Bottom-Up " → Bottom Up "

S → aABe , A → Abc | a , B → d abcde

Top Down Approach



S ⇒ aABe

⇒ aAbcBe

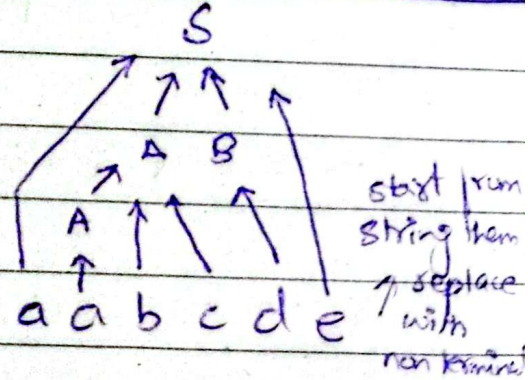
⇒ aabcBe

⇒ aabcde

(left most derivation)

Decision: which production to chose

Bottom Up Approach



S ⇒ aABe

S ⇒ aAde

S ⇒ aAbcde

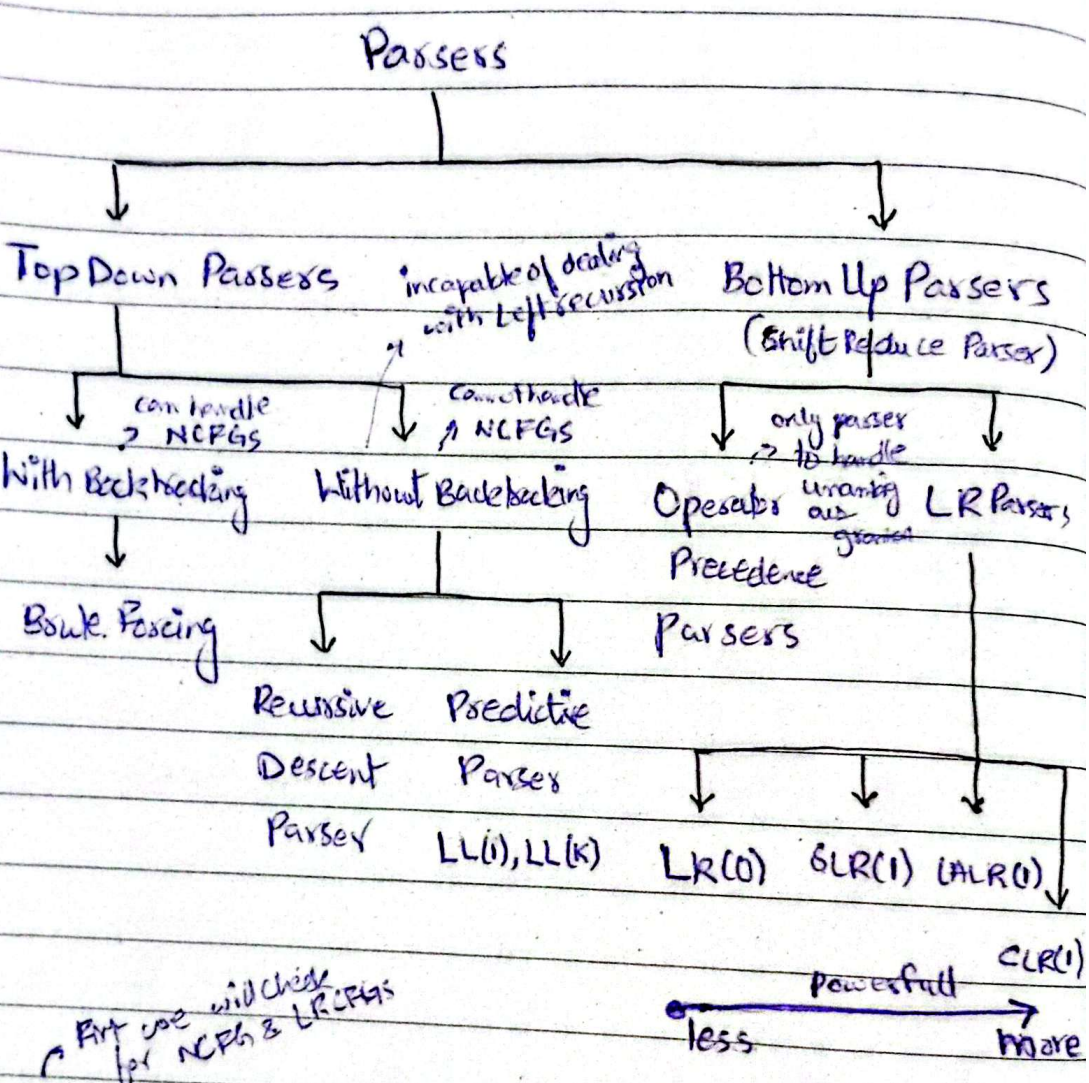
S ⇒ aabcde

(Right most Deriv) - in reverse

Decision: When to reduce

Date _____ Day _____

▷ Classification of Parser



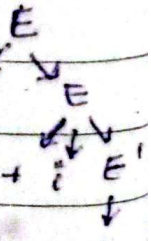
▸ Recursive Descent Parser :

→ In this parser we convert each of the Non-terminal into an equivalent Function. Then all those function becomes exactly like the gramar.

$E \rightarrow i\,E'$

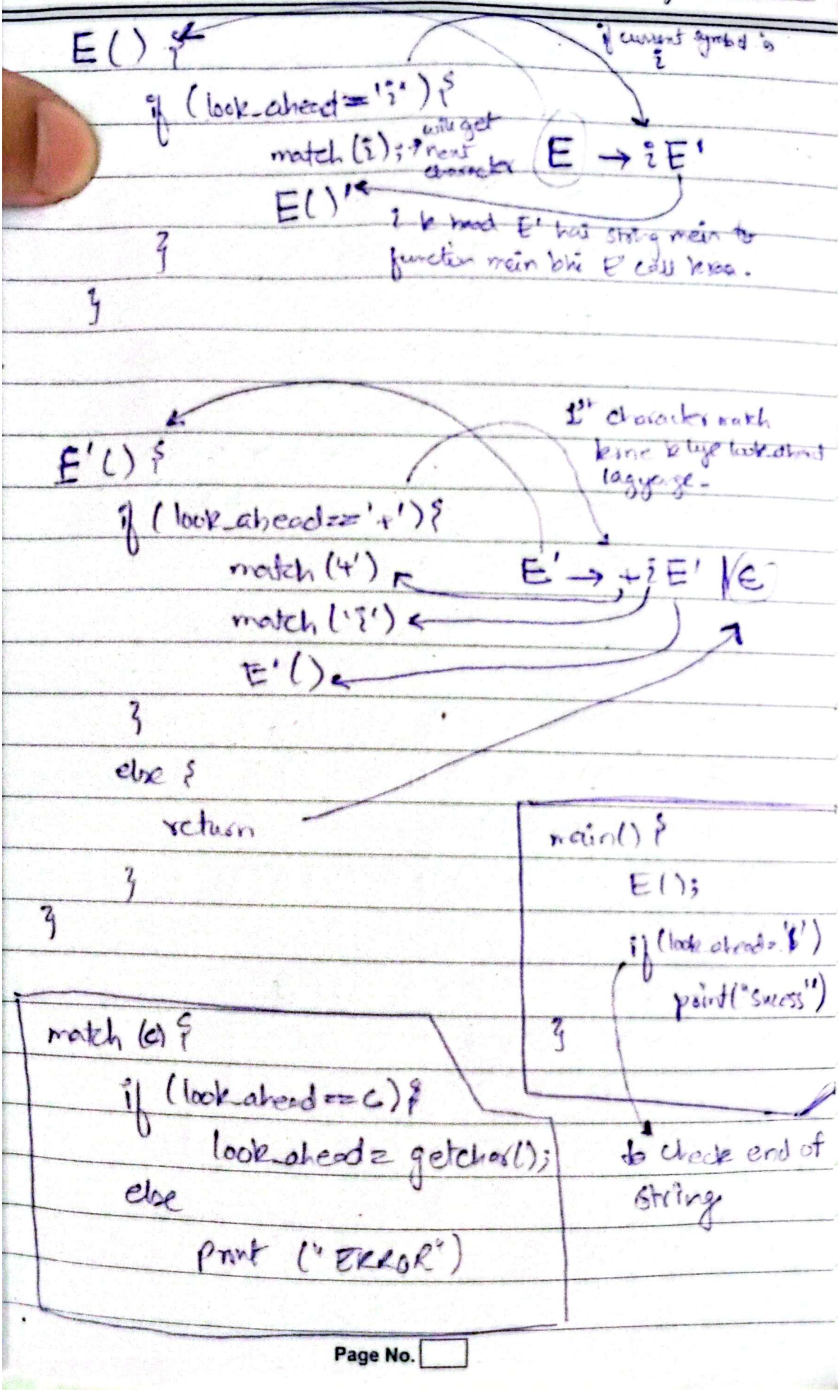
$E' \rightarrow +\,i\,E' \mid \epsilon$

Input: i+i$ →



" We declaration functions to all procedure which will missed the grammar "

Date _____ Day _____



Date _____ Day _____

▸ FIRST() & FOLLOW()

• FIRST()

→ First(A) ⇒ gives set of terminals that begins in all strings derive from A.

1. if A → aα (α: any terminal / non-terminal)
   First(A) = {a}
   ↳ A → bSCB
   First(A) = {b}

2. if A → ∈
   First(A) = {∈}

3. if A → BC
   i- First(A) = First(B) → if First(B) doesn't contain ∈
   ii- First(A) = First(B) ∪ First(C) if First(B) contain ∈
   iii- First(A) = First(B) ∪ First(C) ∪ {∈} if both B & C have ∈

   A → BC → First(A) = {b}
   B → b , C → d
   First(B) = {b} ↳ First(C) = {d}

   A → BC → First(A) = {b, d}
   B → b | ∈ , C = d
   ↳ First(B) = {b, ∈} ↳ First(C) = {d}
   ↳ ismain ∈ hai to C bhi dekhenge.

Date _____ Day _____

$A \to BC \to First(A) = \{b, d, \epsilon\}$ → ye tab hi dekhta hai jab dono B & C mein ε ho.

$B \to b \mid \epsilon \to First(B) = \{b, \epsilon\}$

$C \to d \mid \epsilon \to First(C) = \{d, \epsilon\}$

→ Follow() k set mein "ε" nhi aata kabhi

- Follow()

1. If S is start symbol then
   Follow(S) = { $ }

2. If $A \to \alpha B \beta$, then (β can be terminal or non-terminal)
   Follow(B) = First(β)

3. If $A \to \alpha\beta$, then
   Follow(B) = Follow(A)

E' on right hand side having follow terminal so in 1st production $E \to TE'$ ⇒ Follow(E') = Follow(E)
in 2nd production Follow(E') = Follow(E)

- Solved Examples of ~~First() & Follow()~~

	First	Follow
$E \to TE'$	$\{id, (\}$	$\{\$,)\}$
$E' \to +TE' \mid \epsilon$	$\{+, \epsilon\}$	$\{\$,)\}$
$T \to FT'$	$\{id, (\}$	$\{+, \$,)\}$
$T' \to *FT' \mid \epsilon$	$\{*, \epsilon\}$	$\{+, \$,)\}$
$F \to id \mid (E)$	$\{id, (\}$	$\{*, +, \$,)\}$

Always start with bottom terminal in first
" " Top " " bottom

Follow(C) = Follow(B)

Date

	First()	Follow()
• S → aBDh	{a}	{$}
B → cC	{c}	{$, g, f, h}
C → bC | ∈	{b, ∈}	{g, f, h}
D → EF	{g, f, ∈}	{h}
E → g | ∈	{g, ∈}	{f, h}
F → f | ∈	{f, ∈}	{h}

Follow(B):

S → aBDh

First(D) = {g, f, ∈}

(g) S → aB(g)h — Follow(B) = {g}

(f) S → aB(f)h — Follow(B) = {g, f}

(∈) S → aB(∈)h — Follow(B) = {g, f, h}

↳ epsilon but kreage to "h" hojayeg follow of B.

Follow(E):

D → E(F)

First(F) = {f, ∈}

(f) D → E(f) — Follow(E) = {f}

(∈) D → E — Follow(E) = {f, h}

follow of D → {h}

Date _____ First() Day _____

- $E \to TE'$
  $E' \to \epsilon \mid +TE'$
  $T \to FT'$
  $T' \to \epsilon \mid * FT'$
  $F \to id \mid (E)$

	First()	Follow()
• $S \to Bb \mid Cd$	$\{a, b, c, d\}$	$\{\$\}$
$B \to aB \mid \epsilon$	$\{a, \epsilon\}$	$\{b\}$
$C \to cC \mid \epsilon$	$\{c, \epsilon\}$	$\{d\}$
• $S \to AaAb \mid BbBa$	$\{\epsilon, ab\}$	$\{\$\}$
$A \to \epsilon$	$\{\epsilon\}$	$\{a, b\}$
$B \to \epsilon$	$\{\epsilon\}$	$\{b, a\}$
• $P \to xQRS$	$\{x\}$	$\{\$\}$
$Q \to yz \mid z$	$\{y, z\}$	$\{w, y\}$
$R \to w \mid \epsilon$	$\{w, \epsilon\}$	$\{y\}$
$S \to y$	$\{y\}$	$\{\$\}$
• $S \to ABCDE$	$\{a, b, c\}$	$\{\$\}$
$A \to a \mid \epsilon$	$\{a, \epsilon\}$	$\{b, c\}$
$B \to b \mid \epsilon$	$\{b, \epsilon\}$	$\{c\}$
$C \to c$	$\{c\}$	$\{e, \$, d\}$
$D \to d \mid \epsilon$	$\{d, \epsilon\}$	$\{e, \$\}$
$E \to e \mid \epsilon$	$\{e, \epsilon\}$	$\{\$\}$

Page No.

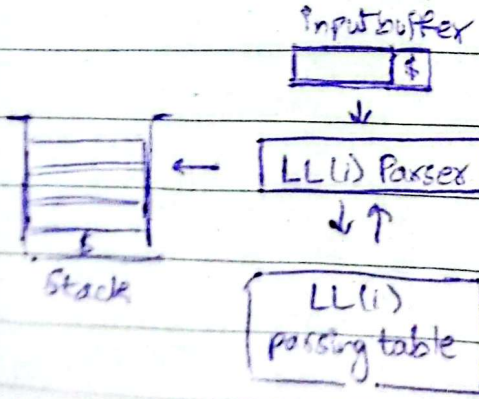
Date _____ Day _____

• $S \to Bb \mid Cd$	{a, b, c, d}	{$}
$B \to aB \mid \epsilon$	{a, ε}	{b}
$C \to cC \mid \epsilon$	{c, ε}	{d}

• $S \to ACB \mid CbB \mid Ba$	{d, g, h, ε, b, a}	{$}
$A \to da \mid BC$	{d, g, h, ε}	{h, g, $}
$B \to g \mid \epsilon$	{g, ε}	{$, a, h, g}
$C \to h \mid \epsilon$	{h, ε}	{$, g, h, g, $}

▷ LL(1) Parser :

- Non-recursive Descent Parser
- They are predictive parser
- "1" look-ahead symbol
- Use "L"eft most derivation
- Scan from "L"eft to right



Date _____ Day

	First()	Follow()
$E \to TE'$	{id, (}	{$,)}
$E' \to \epsilon \mid +TE'$	{ε, +}	{$,)}
$T \to FT'$	{id, (}	{+, $,)}
$T' \to \epsilon \mid *FT'$	{ε, *}	{+, $,)}
$F \to id \mid (E)$	{id, (}	{*, +, $,)}

1. Add $A \to \alpha$ under $M[A, a]$ where $a \in first(\alpha)$ → Simply jis First(A) nikalenge or un terminals par value put karenge. Agr usmein ε hai to follow(A) k set mein $A \to \epsilon$ likhenge.

2. Add $A \to \alpha$ under $M[A, a]$ where $a \in follow(A)$ if first(α) contain ε

M	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	$F \to id$			$F \to (E)$		

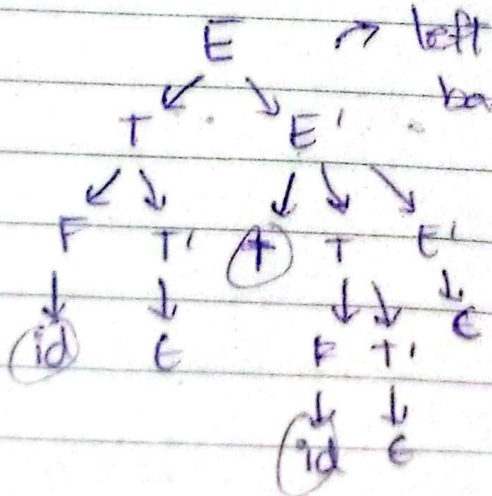
Agr ek cell mein do entries aajayein so that Grammar cannot be passed thru LL(1) parser.

Date _____ Day _____

severe main dealings production rule?

Stack	Input	Production
$E	id+id$	E → TE'
$EE'T	id+id$	T → FT'
$EE'T'F	id+id$	F → id
$EE'T'id	id+id$	pop()
$EE'T' pop	+id$	T' → ∈
$EE'	+id$	E' → +TE'
$EE'T+	+id$	pop()
$EE'T pop	id$	T → FT'
$EE'T'F	id$	F → id
$EE'T'id	id$	pop()
$EE'T' pop	$	T' → ∈
$EE'	$	E' → ∈
$E	$	~~Accept~~ pop(!)

E → left most derivation ~~banegi~~



→ If G is Ambiguous, not CFGs, LR CFGs then also it is not LL(1).

Date _____ IDENTIFYING LL(1) Grammar Day _____

- If G doesn't contain ∈

  $A \to \alpha_1 \mid \alpha_2 \mid \alpha_3$ → for production rule of same LHS

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

  $First(\alpha_1) \cap First(\alpha_3) = \phi$

  $First(\alpha_2) \cap First(\alpha_3) = \phi$

- If G contains ∈

  $A \to \alpha_1 \mid \alpha_2 \mid \in$

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

  $First(\alpha_1) \cap Follow(A) = \phi$

  $First(\alpha_2) \cap Follow(A) = \phi$

- $S \to E \mid a$ $\quad a \cap a = a$

  $E \to a$ ↗ LL(1) X

  $First(E) = a$

- $S \to aABb$ ✓ → ∵ only one production so rule not applied

  $A \to a \mid \in$ ✓ → $a \cap \{d, b\} = \phi$

  $B \to d \mid \in$ → $d \cap \{d\} = \phi$

- $S \to aSA \mid \in$ → $a \; A \; \{c, \$\} = \phi$ ✓

  $A \to c \mid \in$ → $c \cap \{c, \$\} = \phi$ X

  LL(1) X