CS & IT ENGINEERING

Compiler Design

Lexical & Syntax Analysis

Lecture No. 5

By- DEVA Sir

01 Elimination of Left Recursion

02 Left Factoring

03 FIRST & FOLLOW Set

04

05

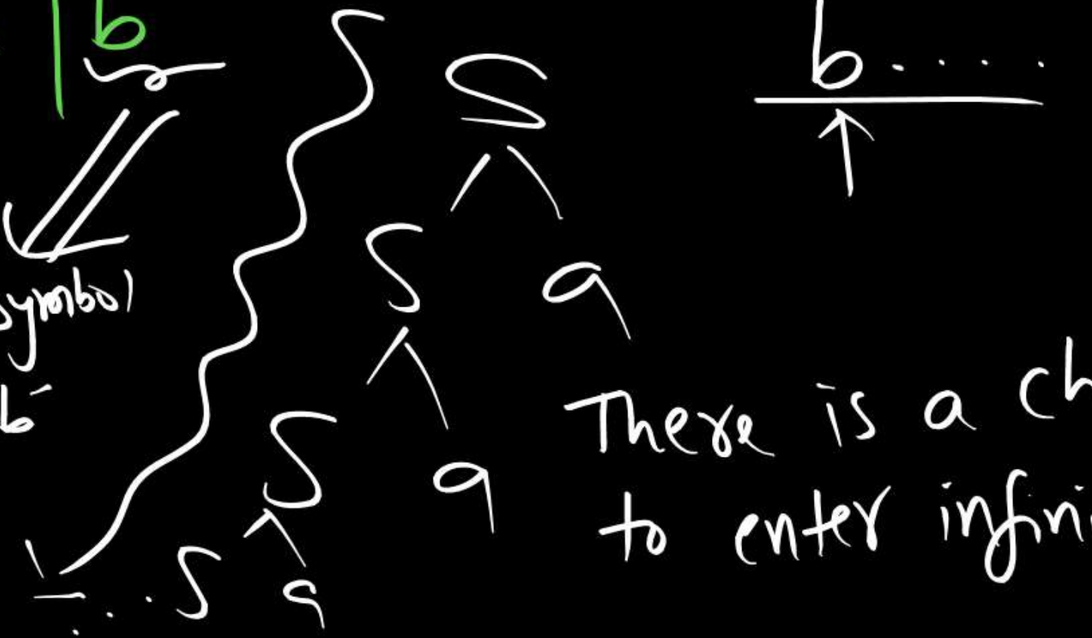
Elimination of Left Recursion :

- Top-down parser not possible if CFG has left Rec.
- Conversion from Left Rec to Right Rec

$S \rightarrow Sa \mid b$

First Symbol is 'b'

Uses LMD :



There is a chance to enter infinite Rec

① $S \rightarrow \underbrace{Sa}_{\text{Left Rec}} \mid \underbrace{b}_{\text{Non Left Rec}}$

$L = \{b, ba, baa, baaa, \ldots\}$

$= b\boxed{a^*}$

$S \rightarrow bX$

$X \rightarrow aX \mid \epsilon$

① $S \rightarrow Sa \mid b$

Sa: Left Rec

b: Non Left Rec

⇓

$S \rightarrow bX$

$X \rightarrow aX \mid \varepsilon$

OR

$S \rightarrow bX \mid b$

$X \rightarrow aX \mid a$

② $S \rightarrow Sa \mid Sb \mid c \mid d \mid eS$

Left Rec

Non Left Rec

⇩

$S \rightarrow cX \mid dX \mid esX$

$X \rightarrow aX \mid bX \mid \varepsilon$

Note:

$$A \to A\alpha_1 \mid A\alpha_2 \mid \ldots \mid A\alpha_k \mid \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n$$

k left Rec terms — n non-left rec terms

⇓ Direct Left Rec Elimination

$$A \to \beta_1 X \mid \beta_2 X \mid \ldots \mid \beta_n X$$

$$X \to \alpha_1 X \mid \alpha_2 X \mid \ldots \mid \alpha_k X \mid \varepsilon$$

③ $S \rightarrow Sab \mid cdS \mid Se \mid f \mid gh$

$\Downarrow$

$S \rightarrow \underbrace{Sab \mid Se} \mid \underbrace{cdS \mid f \mid gh}$

$\Downarrow$

$S \rightarrow cdsX \mid fX \mid ghX$

$X \rightarrow abX \mid eX \mid \varepsilon$

④

$S \rightarrow Sa \mid Abc \mid efg$

Sa: Direct Left Rec

Abc: Indirect Left rec

$A \rightarrow Sbs \mid f$

Sbs: Indirect Left Rec

Answer:

① ②

Step1 : choose S

$S \rightarrow Sa \mid Abc \mid efg$

Sa: Left Rec

Abc | efg: Assume non left rec

$S \rightarrow AbcX \mid efgX$

$X \rightarrow aX \mid \varepsilon$ → ①

Step2 : Choose A

$A \rightarrow Sbs \mid f$

⇓ substitute S in 1st place

$A \rightarrow Abcxbs \mid efgxbs \mid f$

⇓ Apply Direct left rec elimination

$A \rightarrow efgxbsY \mid fY$

$Y \rightarrow bcxbsY \mid \varepsilon$ → ②

S →

A →

B →

C → ;

Algo 1

S

A

B

C

Algo 2

S

A

C

B

Algo 3

A

B

S

C

. . . .

We follow this one

4! Algorithms
24 Answers

Algo 1

S →
A →
B →
C →

S Step 1: Solve S using Direct Left Rec elimination

A Step 2: Substitute S in A rules then apply Direct Left Rec elim.

B Step 3: Substitute both S & A in B rules then apply Direct elimination recursively

C Step 4: Substitute S, A, and B in C productions recursively then Apply Direct left rec elimination

Direct Left Rec

⇓

Visible

Indirect Left Rec

⇓

Substitute using Alg (only 1st place in RHS)

⇓

Direct Left Rec

(5) $E \to E+T \mid a$

$T \to T*F \mid b$

$F \to c$

$\Rightarrow$

$E \to aX$

$X \to +TX \mid \varepsilon$

$T \to bY$

$Y \to *FY \mid \varepsilon$

$F \to c$

⑥ $S \rightarrow Ab$
$A \rightarrow Sa \mid Ad \mid f$
$B \rightarrow Sg \mid Ah \mid e$

Step 1:

$S \rightarrow Ab$ → ①

Step 2:

$A \rightarrow Sa \mid Ad \mid f$

↓ substitute S from ①

$A \rightarrow Aba \mid Ad \mid f$

↓ Apply elimination

$A \rightarrow fX$
$X \rightarrow baX \mid dX \mid \varepsilon$ → ②

Step 3:

$B \rightarrow Sg \mid Ah \mid e$

↓ Substitute S from ① A from ②

$B \rightarrow Abg \mid fXh \mid e$

↓ Substitute A from ②

$B \rightarrow fXbg \mid fXh \mid e$ → ③

Answer:

① ② ③

Left Factoring

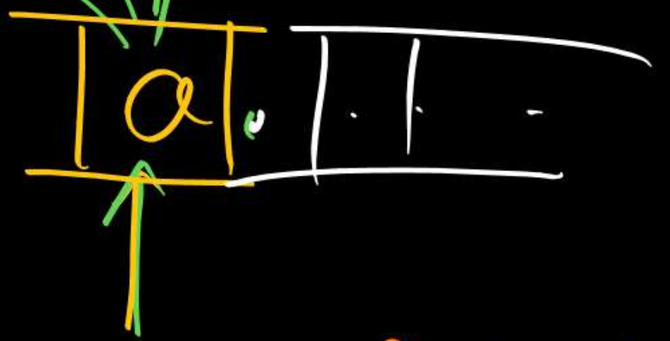
Unambiguous CFG

Left factoring

Left Factored CFG [free from common prefixes]

$S \rightarrow ab \mid ac \mid abd \mid acf \mid dg$

Not Left factored



Backtracking issue

LL(1)

Knowledge of 1 token

make this CFG into Left factored by eliminating common prefix.

① $S \rightarrow ab \mid ac$ Not Left factored

⇓ Left Factoring

$S \rightarrow aX$

$X \rightarrow b \mid c$

Left factored CFG

② $S \to a \mid ab \mid abc \mid d$

$S \to aX \mid d$

$X \to \varepsilon \mid b \mid bc \implies X \to \varepsilon \mid bY$

$Y \to \varepsilon \mid c$

$S \to aX \mid d$

$X \to \varepsilon \mid bY$

$Y \to \varepsilon \mid c$

(3) $S \rightarrow Ab \mid acA \mid d$

$A \rightarrow ab \mid f$ → (2)

Indirect Common prefix

---

non-terminal substitute

$S \rightarrow \boxed{A}b \mid acA \mid d$

⇓ Substitute A

$S \rightarrow abb \mid fb \mid acA \mid d$

⇓

$S \rightarrow aX \mid fb \mid d$

$X \rightarrow bb \mid cA$ → (1)

$S \rightarrow aX \mid fb \mid d$

$X \rightarrow bb \mid cA$

$A \rightarrow ab \mid f$

④ $S \rightarrow Sab \mid bA$

$A \rightarrow Bc \mid deB$

$B \rightarrow f \mid d$

⇓ After elimination of Left Rec

$S \rightarrow bAX$

$X \rightarrow abX \mid \varepsilon$

$A \rightarrow Bc \mid deB$

$B \rightarrow f \mid d$

i) $S \rightarrow bAX$ ✓

ii) $X \rightarrow abX \mid \varepsilon$ ✓

iii) $A \rightarrow Bc \mid deB$

⇓ Substitute B

$A \rightarrow fc \mid dc \mid deB$

$A \rightarrow fc \mid dY$

$Y \rightarrow c \mid eB$ ✓

iv) $B \rightarrow f \mid d$ ✓

Left Factoring Algorithm

Unambiguous CFG

step 1 : Eliminate Left Rec

Non Left Recursive CFG

step 2 : Substitute nonterminals if they appear in 1st place

CFG which is free from "nonterminal that appears in 1st place"

step 3 : eliminate Direct common prefixes

Left factored CFG

FIRST Set and FOLLOW Set Computation:

FIRST(X) = {a, f, e, ε}

$$X \longrightarrow \overbrace{a\gamma_b}^{1^{st}\text{ terminal}} \mid \overbrace{f\gamma_g}^{1^{st}\text{ terminal}} \mid \overbrace{e}^{1^{st}\text{ terminal}} \mid \varepsilon$$

First (X) = Set of terminals (including ε) where every terminal is derived as 1st symbol from X

$$= \{ t \mid X \Rightarrow t\alpha, \text{ where } t \text{ is terminal} \}$$

or epsilon if t is not there

FIRST Set and FOLLOW Set Computation:

Follow(Y) = {b, g}

Any LHS ⟶ a Y b | f Y g | e | ε

Follow (Y) = Set of terminals where every terminal is derived as 1st symbol after Y

= {t | t is terminal, S ⇒ α Y t β }

FIRST(A)

Look at all A productions

$A \rightarrow$ ____

FOLLOW(A)

Look at whole CFG where RHS has A

$S \rightarrow AB \mid aAb \mid \epsilon$

$A \rightarrow Ba \mid Aa \mid bAc \mid d$

$B \rightarrow SAb \mid Ab \mid a$

FIRST(A) = { }

⇓

First (Ba)
∪
First (Aa)
∪
First(bAc)
∪
First(d)

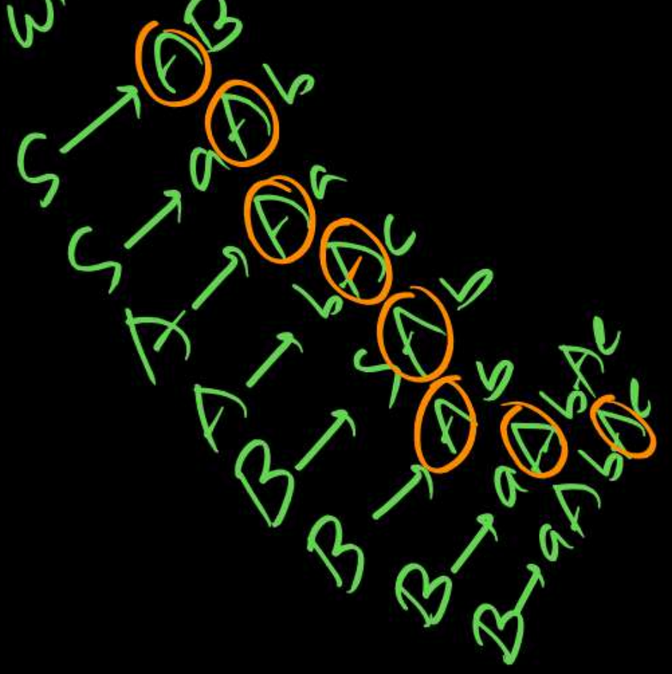
$S \rightarrow AB \mid aAb \mid \epsilon$

$A \rightarrow Ba \mid Aa \mid bAc \mid d$

$B \rightarrow SAb \mid Ab \mid aAbAc$ (two time)

Follow(A) = ?

Answer will depend on:



Note : FIRST(X) may contain ε

FOLLOW(X) never contain ε

Computation of FIRST Sets :

① $S \rightarrow a \mid \varepsilon \mid bcd$

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

② $S \rightarrow Aab \mid c$

$A \rightarrow d$

First(S) = {c, d}

First(A) = {d}

---

③ $S \rightarrow Aab \mid c$

$A \rightarrow \varepsilon$

What is 1st terminal?

First(Aab)

First(ab) = {a}

First(S) = {a, c}

First(A) = {ε}

④ $S \rightarrow A$

$A \rightarrow \varepsilon$

$First(S) = \{\varepsilon\}$

$First(A) = \{\varepsilon\}$

⑤ $S \rightarrow AB$

$A \rightarrow \varepsilon$

$B \rightarrow a$

$First(S) = First(AB) = First(a)$ (ε, a)

$= \{a\}$

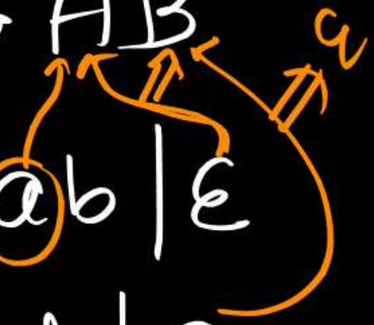
$First(A) = \{\varepsilon\}$

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

⑥ $S \rightarrow AB$

$A \rightarrow ab \mid \varepsilon$

$B \rightarrow cd \mid \varepsilon$



First(S) = {a, c, ε}

First(A) = {a, ε}

First(B) = {c, ε}

Size of First(S) = 3

i) 1

ii) 2

iii) 3 ✓

Summary

- Elimination of Left Rec
- Left factoring
- FIRST Set
- Next : Follow Set


THANK YOU
GW
SOLDIERS !