CS & IT ENGINEERING

Compiler Design

Lexical Analysis & Syntax Analysis

Lecture No. 5

By- DEVA Sir

TOPICS TO BE COVERED

- Elimination of Left Recursion [Conversion from Left Rec to Right Rec]
- Elimination of common prefixes [Left Factoring Algorithm]
- FIRST and FOLLOW Sets
- Types of parsers

(10) $\boxed{S} \to \boxed{S}a\boxed{S} \mid b$

Left Rec & Right Rec with Same S — Ambiguous

(11) $\boxed{S} \to \boxed{S}a \mid b\boxed{S} \mid c$ — Ambiguous

(12) $\boxed{E} \to \boxed{E}+\boxed{E} \mid a$ — Ambiguous

(13) $S \to AB \mid ab$

$A \to aA \mid \varepsilon$

$B \to bB \mid \varepsilon$

ab:

Ambiguous

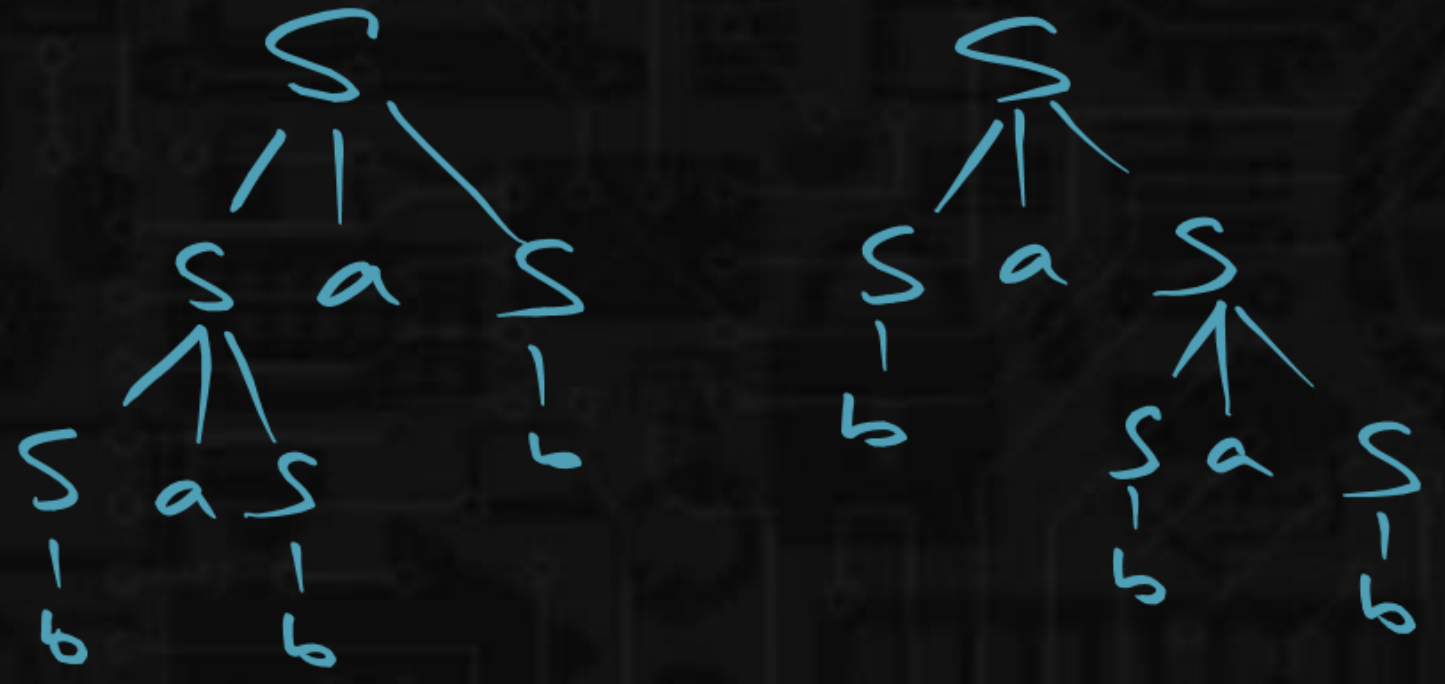
H.W.

(10) $S \rightarrow SaS \mid b$

$b \Rightarrow 1$ PT

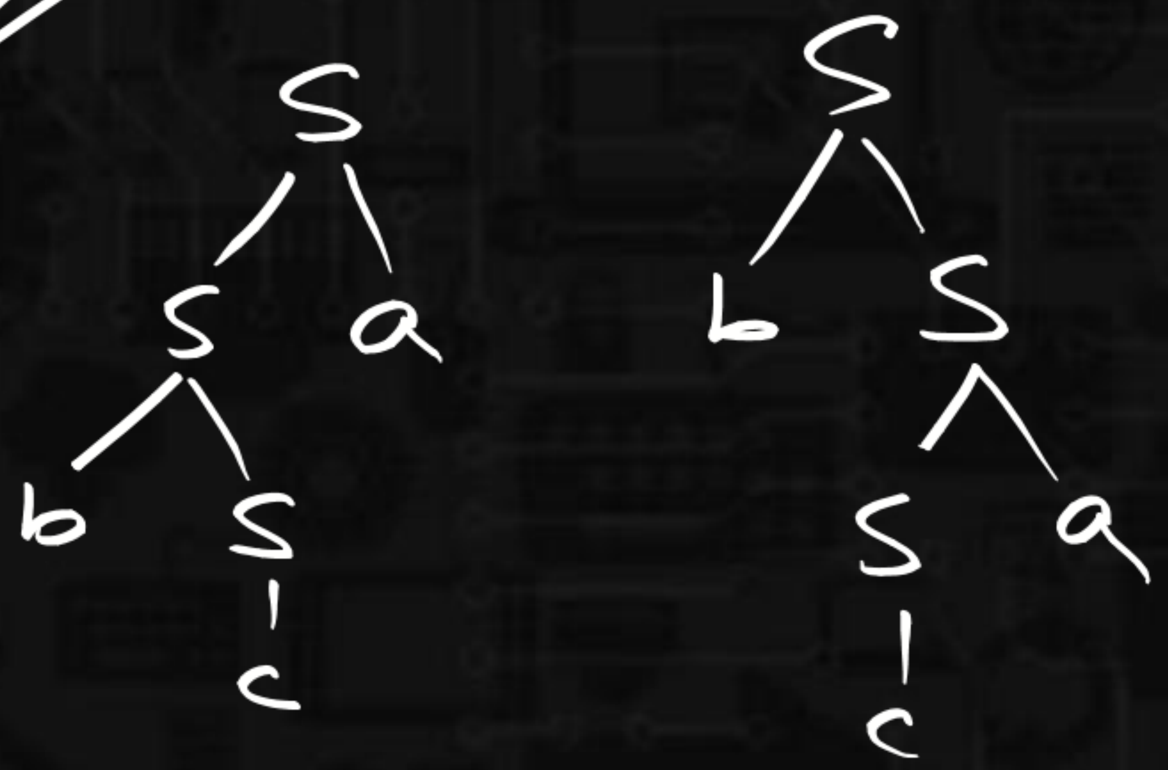
$bab \Rightarrow 1$ PT

$babab \Rightarrow 2$ PT



(11) $S \rightarrow Sa \mid bS \mid c$

Ambiguous



$c \Rightarrow 1$ PT

$ca \Rightarrow 1$ PT

$bc \Rightarrow 1$ PT

$bca \Rightarrow 2$ PT

Elimination of Left Recursion:

(Conversion from Left Rec to Right Rec)

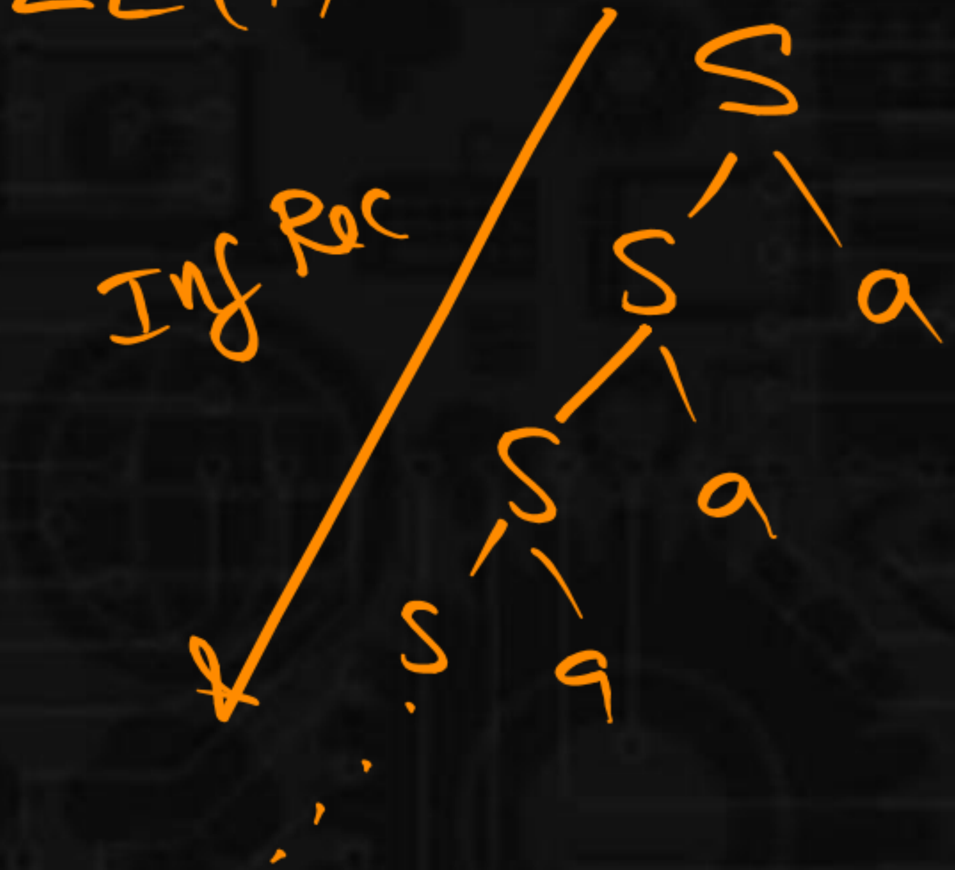
Top-down parser follows LMD.

$$S \rightarrow \overset{b\ldots}{Sa} \mid \overset{b}{b}$$

First Symbol

b

LL(1)



(1) $S \to Sa \mid b$

$\underbrace{Sa}_{\text{Left Rec}} \quad \underbrace{b}_{\text{Non Left Rec}}$

$A \to A\alpha \mid \beta$

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

$= \{ ba^0, ba^1, ba^2, ba^3, \ldots \}$

$= b\boxed{a^*}$

$$\boxed{\begin{array}{l} S \to bS' \\ S' \to aS' \mid \varepsilon \end{array}}$$

$$\boxed{\begin{array}{l} S \to bX \\ X \to aX \mid \varepsilon \end{array}}$$

$A \to \beta A'$

$A' \to \alpha A' \mid \varepsilon$

Formula:

(Direct Left Rec Elimination)

$$A \rightarrow \underbrace{A\alpha_1 \mid A\alpha_2 \mid \dots \mid A\alpha_n}_{\text{Left Rec}} \mid \underbrace{\beta_1 \mid \beta_2 \mid \dots \mid \beta_k}_{\text{Non Left Rec}}$$

$\alpha \in (V \cup T)^*$

$\beta \in (V \cup T)^*$ ↳ should not start with A

$$\Downarrow$$

$$A \rightarrow \beta_1 A' \mid \beta_2 A' \mid \dots \mid \beta_k A'$$

$$\underbrace{\boxed{A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \dots \mid \alpha_n A'}}_{\text{Right Rec}} \mid \varepsilon$$

② $S \rightarrow \underbrace{Sab}_{\alpha} \mid \underbrace{c}_{\beta_1} \mid \underbrace{d}_{\beta_2}$

$S \rightarrow S\alpha \mid \beta_1 \mid \beta_2$

$$\boxed{\begin{array}{l} S \rightarrow cS' \mid dS' \\ S' \rightarrow abS' \mid \varepsilon \end{array}} \quad \text{OR} \quad \boxed{\begin{array}{l} S \rightarrow cS' \mid dS' \mid c \mid d \\ S' \rightarrow abS' \mid ab \end{array}}$$

(3) $S \to Sa \mid Sb \mid cd \mid e \mid fS \mid Sg$

$$\Downarrow$$

$$S \to \underbrace{S\underset{\alpha_1}{a} \mid S\underset{\alpha_2}{b} \mid S\underset{\alpha_3}{g}}_{\text{Left Rec}} \mid \underbrace{\underset{\beta_1}{cd} \mid \underset{\beta_2}{e} \mid \underset{\beta_3}{fS}}_{\text{Non Left Rec}}$$

$\Downarrow$ Direct Left Recursion Elimination

$$\boxed{\begin{aligned} S &\to cdX \mid eX \mid fSX \\ X &\to aX \mid bX \mid gX \mid \epsilon \end{aligned}}$$

Left Recursion

$$\boxed{S} \to \boxed{S}a \mid b$$

Direct Left Rec

$$S \to \underline{A}a \mid d$$

$$A \to Sb \mid c$$

Indirect Left Recursion

$\boxed{\text{Indirect}}$ Left Recursion

$\Downarrow$

How to convert into $\boxed{\text{Direct}}$ Left Rec?

$\boxed{\text{Using Substitution}}$

Step1 : S [Direct Left Rec in S]

Step2 : A i) Substitute S in A then ii) Eliminate Direct left rec in A

Step3 : B i) Substitute S and A productions recursively in B then (ii) Apply direct left rec elimi. in B

Step4 : C i) Sub S, A, B in C then (ii) Eliminate Direct left rec in C

①

$$S \rightarrow Aa \mid Sb \mid c$$

$$A \rightarrow Sb \mid Ae \mid f$$

$$\begin{tikzcd} S \arrow[d, bend right] \\ A \arrow[u, bend right] \end{tikzcd}$$

Algo 1 :	Algo2 :
Step1 :	Step1 :
S	A
Step2 :	Step2 :
A	S

①

$$S \to Aa \mid Sb \mid c$$

$$A \to Sb \mid Ae \mid f$$

Step 1: choose S productions

$$S \to \underbrace{Sb}_{\text{Left rec}} \mid \underbrace{Aa \mid c}_{\text{Non left Rec}}$$

$\Downarrow$ Direct Left Rec elimination

$$\boxed{\begin{aligned} S &\to AaX \mid cX \\ X &\to bX \mid \varepsilon \end{aligned}} \to ①$$

Step 2: Choose A productions

$$A \to \boxed{S}b \mid Ae \mid f$$

$\Downarrow$ Substitute S rules if appears in 1^{st} position

$$S \to AaX \mid cX$$

$$A \to A\underbrace{aXb}_{\alpha_1} \mid \underbrace{cXb}_{\beta_1} \mid A\underbrace{e}_{\alpha_2} \mid \underbrace{f}_{\beta_2}$$

$\Downarrow$ Direct left rec elimi..

$$\boxed{\begin{aligned} A &\to cXbY \mid fY \\ Y &\to aXbY \mid eY \mid \varepsilon \end{aligned}} \to ②$$

② $E \to E+T \mid F$

$T \to F*T \mid F$

$F \to id$

Step1: $E \to \underline{E}\underbrace{+T}_{\alpha} \mid \underbrace{F}_{\beta}$

$\Downarrow$

$$\boxed{\begin{aligned} &E \to FX \\ &X \to +TX \mid \varepsilon \end{aligned}} \to ①$$

Step2: $\boxed{T \to F*T \mid F} \to ②$

$\Downarrow$ no E in 1st position. So, no substitution

no Direct Left Rec

Step3: $\boxed{F \to id} \to ③$

Answer:

$$\boxed{\begin{aligned} &E \to FX \\ &X \to +TX \mid \varepsilon \\ &T \to F*T \mid F \\ &F \to id \end{aligned}}$$

③ $S \to Aa$

$A \to Sb \mid Bc$

$B \to Se \mid f$

Step1 : Choose S

$\boxed{S \to Aa}$ → ①

Step 2 : Choose A

$A \to \boxed{S}b \mid Bc$

⇓ Sub 'S' in 1st place $\boxed{S \to Aa}$

$A \to Aab \mid Bc$

⇓ Eliminate Direct Left Rec in A

$\boxed{\begin{array}{l} A \to BcX \\ X \to abX \mid \epsilon \end{array}}$ → ②

Step3 : Choose B

$B \to \boxed{S}e \mid f$

⇓ Sub S & A in B

$B \to \boxed{A}ae \mid f$

⇓ Sub A in B

$B \to BcXae \mid f$ $\xrightarrow[\text{Direct left rec}]{\text{Eliminate}}$ $\boxed{\begin{array}{l} B \to fY \\ Y \to cXaeY \mid \epsilon \end{array}}$ → ③

$$X \rightarrow X\alpha_1 \mid X\alpha_2 \mid \beta_1 \mid \beta_2 \mid \beta_2$$

$$\Downarrow$$

$$X \rightarrow \beta_1 X' \mid \beta_2 X' \mid \beta_3 X'$$

$$X' \rightarrow \alpha_1 X' \mid \alpha_2 X' \mid \varepsilon$$

Left Factoring:

CFG → Elimination of common prefixes (Left factoring) → Left Factored CFG (CFG without common prefixes)

$S \to aX$
$X \to b \mid cd$

Not Left factored

$S \to \boxed{a}b \mid \boxed{a}cd$

One length

common prefix

⇓

Issue is Backtracking

a . . . .

1. $S \to a$ Left factored CFG

2. $S \to a \mid \varepsilon \mid bc$ Left factored CFG

3. $\left.\begin{array}{l} S \to Aa \\ A \to b \end{array}\right\}$ Left factored

(4) $S \rightarrow \boxed{a}b \mid \boxed{a} \mid cd$ ✓

$\Downarrow$

$$\boxed{\begin{array}{l} S \rightarrow aX \mid cd \\ X \rightarrow b \mid \varepsilon \end{array}}$$

Left factored

5

$S \rightarrow \boxed{a} \mid \boxed{a}b \mid \boxed{a}bc \mid ef$

$$\Downarrow$$

$\boxed{S \rightarrow aX \mid ef} \rightarrow 1$

$X \rightarrow \varepsilon \mid \boxed{b} \mid \boxed{b}c$

$$\Downarrow$$

$\boxed{\begin{array}{l} X \rightarrow \varepsilon \mid bY \\ Y \rightarrow \varepsilon \mid c \end{array}} \rightarrow 2$

$\boxed{\begin{array}{l} S \rightarrow aX \mid ef \\ X \rightarrow \varepsilon \mid bY \\ Y \rightarrow \varepsilon \mid c \end{array}}$

(6)

sub A

$S \to \boxed{A}b \mid \underline{a}cA \mid d$

$$\boxed{A \to ae \mid g}$$

$$\Downarrow$$

$S \to \boxed{a}eb \mid gb \mid \boxed{a}cA \mid d$

$$\Downarrow$$

$$\boxed{\begin{aligned} S &\to aX \mid gb \mid d \\ X &\to eb \mid cA \end{aligned}}$$

Answer:

$S \to aX \mid gb \mid d$

$X \to eb \mid cA$

$A \to ae \mid g$

FIRST SET

$S \rightarrow \boxed{a}bc \mid \boxed{d}e \mid \boxed{\varepsilon}$

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

$= \{x \mid x$ is either ε or terminal where terminal is derived as 1^{st} symbol from $S\}$

FOLLOW SET

$X \rightarrow a\boxed{X}b \mid Yb \mid a$

$Y \rightarrow b\boxed{X}ef \mid \boxed{X}m \mid \varepsilon$

$Follow(X) = \{b, e, m\}$

$= \{t \mid t$ is a terminal derived as 1^{st} symbol after $X\}$

FIRST SET

FIRST(X) = It is set of terminals (may include ε) where each terminal is derived as 1st symbol **from** X

First (abc) = {a}

First (de) = {d}

Note:
we need X rules for computing FIRST(X)

FOLLOW SET

Follow(X) = It is set of terminals where every terminal is derived as **1st symbol** **after** X.

Note: we need whole CFG where production contain X helps to compute Follow(X)

If S is start symbol then Follow(S) should contain \$.

$\boxed{\text{\$ is special end terminal}}$

Example:

$$S \rightarrow S\underline{a} \mid S\underline{b} \mid c \qquad \boxed{S}\boxed{\$}$$

$$\text{Follow}(\underline{S}) = \{\$, a, b\}$$

start   must

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

$\text{Follow}(S) = \{\$\}$

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

---

② $S \rightarrow S\boxed{a} \mid \varepsilon$

$\text{Follow}(S) = \{\$, a\}$

$\text{First}(S) = \{\varepsilon, a\}$

$S \Rightarrow \boxed{a}aa$

$S \Rightarrow \boxed{a}a$

$S \Rightarrow \boxed{a}$

$S \Rightarrow \varepsilon$

$S\boxed{a} \Rightarrow a\ldots$

(3)

$S \rightarrow \boxed{Ab} \mid cde$

$A \rightarrow ae \mid \varepsilon \mid fg$

Follow(S) = { \$ }

start

Follow(A) = { b }

First(S) = { c, a, b, f }

First(A) = { a, ε, f }

(4)

$S \rightarrow \boxed{A}$

$A \rightarrow \varepsilon$

Follow(S) = { \$ }

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

First(S) = { ε }

First(A) = { ε }

Note:

$$\underset{LHS}{X} \longrightarrow \alpha \boxed{Y}$$

$$X\ \underline{a}$$

$$\Downarrow$$

$$\alpha Y \underline{a}$$

same as Follow(X)

$$\text{Follow}(Y) = \underset{LHS}{\text{Follow}(X)}$$

$$\ldots\ \boxed{X}\,\boxed{a}\ \ldots$$

$$\ldots\ \alpha\boxed{Y}\,\boxed{a}\ \ldots$$

If you want to find after Y, it is same as whatever after X

FIRST(X) :

$X \rightarrow \bigcirc \mid \bigcirc \mid \bigcirc$

1st symbol (terminal) 1st 1st

Follow(X) :

Whole CFG Take Rule which contain X in RHS

$LHS \rightarrow \boxed{X}$

(5) $S \to AB$

$A \to a$

$B \to b$

(6) $S \to AaBb$

$A \to f$

$B \to f$

(7) $S \to AB$

$A \to ab$

$B \to \varepsilon$

(8) $S \to AB$

$A \to \varepsilon$

$B \to ef$

(9) $S \to SS \mid a$

(10) $S \to SS \mid (S) \mid \varepsilon$

Summary

- → Elimination of Left Recursion
- → Left factoring
- → FIRST & FOLLOW set.

"DEVA SIR PW"

$\underbrace{\hspace{3cm}}$

Telegram group


THANK YOU GW
SOLDIERS !