cat -1

1) Kleen closure

$$L^* = \bigcup_{i=0}^{\infty} L^i$$

5) 00(0/1)*

8) Three consecutive 0's

10) 3 states (divisible by 3) (divisible by 4) 4 Remainders

(not divisible by 3 q1, q2 final state)

0, 1, 2 remainder

0 → Accepted 1, 2 → Not accepted.

(accept state → Non accept vice versa)

Solution:

DFA

R 0, 1, 2

3 % 3 = 0

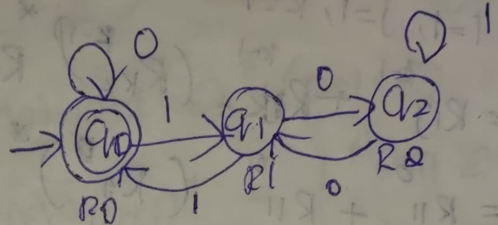
4 % 3 = 1

5 % 3 = 2

6 % 3 = 0

7 % 3 = 1

:



Binary

	R
0 →	0
1 →	1
10 →	2
11 →	0
100 →	1
101 →	2

1111 - 15

110 - 6

20/9/2022

DFA → RE

Regular equation method ($R_{ij}^{(k)}$):

R_{ij}^{k} - R → RE
k → k number of regular expression (intermediate steps)

Case (i)

$R_{ij}^{0} = \{a \in \Sigma \mid \delta(q_i, a) = q_j\}$ if $i \neq j$

case (ii)

$R_{ij}^{0} = \{a \in \Sigma \mid \delta(q_i, a) = q_j\} \cup \{\varepsilon\}$ if $i = j$

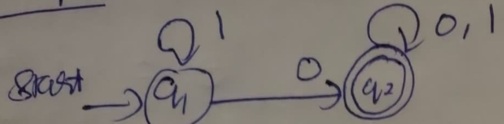
Induction:

$$R_{ij}^{k} = R_{ik}^{k-1} (R_{kk}^{k-1})^* R_{kj}^{k-1} + R_{ij}^{k-1}$$

self loop represents * $(R_{kk}^{k-1})^*$

(name it as q_1, q_2, q_3 and so on)

Example:



$k = 2$ $\quad$ 1, 2 $\quad \begin{cases} 11 \\ 12 \\ 21 \\ 22 \end{cases}$

Solution:

$k = 0$ $\quad i=1, j=1, k=0$

$R_{11}^0 = 1 + \varepsilon \qquad R_{12}^0 = 0 \qquad R_{21}^0 = \phi \qquad R_{22}^0 = 0+1+\varepsilon$

$k = 1$ $\quad i=1, j=1, k=1$

$R_{11}^1 = R_{ij}^{k-1} + R_{ik}^{k-1} (R_{kk}^{k-1})^* R_{kj}^{k-1}$

$R_{11}^1 = R_{11}^0 + R_{11}^0 (R_{11}^0)^* R_{11}^0$

$= (1+\varepsilon) + (1+\varepsilon)(1+\varepsilon)^*(1+\varepsilon)$

$= (1+\varepsilon) + (1+\varepsilon)\, 1^* (1+\varepsilon)$

$= (1+\varepsilon) + 1^*(1+\varepsilon)$

$\boxed{R_{11}^1 = (1+\varepsilon) + 1^*}$

$(1+\varepsilon)^* = (1^*\varepsilon^*)^*$
$= (1^* + \varepsilon^*)^*$
$= (1^*)^* = 1^*$

$(1+\varepsilon)1^* = 1^*$

$R_{12}^1 = R_{12}^0 + R_{11}^0 (R_{11}^0)^* R_{12}^0$

$= 0 + \underbrace{(1+\varepsilon)(1+\varepsilon)^*}_{1^*}\, 0$

$= 0 + 1^*0$

$= 0(\varepsilon + 1^*)$

$R_{12}^1 = 1^*0$

$(1+\varepsilon)^* = (1^*\varepsilon^*)^*$
$= (1^* + \varepsilon^*)^*$
$= (1^*)^* = 1^*$

$R_{21}^1 = R_{21}^0 + R_{21}^0 (R_{11}^0)^* R_{11}^0$

$= \phi + \phi\, (1+\varepsilon)^* (1+\varepsilon)$

$R_{21}^1 = \phi$

$R_{22}^{1} = R_{22}^{0} + R_{21}^{0} (R_{11}^{0})^* R(12)^0$

$= (0+1+\varepsilon) + \phi (1+\varepsilon)^* (0)$

$= 0+1+\varepsilon + \phi$

$R_{22}^{1} = 0+1+\varepsilon$

$K=2$ (If $K=2$ then solve directly)

$R_{11}^{2} = R_{11}^{1} + R_{12}^{1} (R_{22}^{1})^* R_{21}^{1}$

$= ((1+\varepsilon)+1^*) + (1^*0)(0+1+\varepsilon)^* \phi$

1 2
FS FS

[$K=2$ equivalent to no. of states]

$R_{12}^{2} = R_{12}^{1} + R_{12}^{1} (R_{22}^{1})^* R_{22}^{1}$

$= (1^*0) + (1^*0)(0+1+\varepsilon)^* (0+1+\varepsilon)$

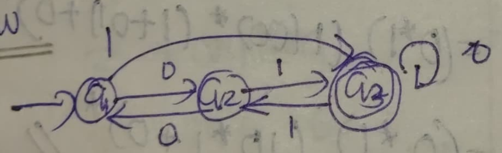
$= \underbrace{(1^*0)}_{R} + \underbrace{(1^*0)}_{P} \underbrace{(0+1)^*}_{Q} = 1^*0 + (\varepsilon + (0+1)^*)$

$\boxed{P(R+Q)}$

$= (1^*0) + (\varepsilon + (0+1)^*)$

$= (1^*0)(\varepsilon + (0+1)^*) = 1^*0(0+1)^*$

H.W



FS: $R_{13}^{3} = R_{13}^{2} + R_{13}^{2} (R_{33}^{2})^* R_{33}^{2}$ $(K=3)$

$(k=2)$ R_{13}^{2} & R_{33}^{2}

solution:

$\boxed{K=0}$ $i=1, j=1, k=0$

$R_{11}^{0} = \varepsilon$ $R_{12}^{0} = 0$ $R_{13}^{0} = 1$ $R_{21}^{0} = 0$ $R_{22}^{0} = \varepsilon$

$R_{23}^{0} = 1$ $R_{31}^{0} = \phi$ $R_{32}^{0} = 1$ $R_{33}^{0} = 0+\varepsilon$

$\boxed{K=1}$ $i=1, j=1, k=1$

$R_{11}^{1} = R_{ij}^{k-1} + R_{ik}^{k-1} (R_{kk}^{k-1})^* R_{kj}^{k-1}$

$= R_{11}^0 + R_{10}^0 (R_{11}^0)^* R_{11}^0$

$= (\ +\varepsilon) + (\ +\varepsilon)(\ +\varepsilon)^* (\ +\varepsilon)$

$R_{11}^1 = (\ +\varepsilon) = \varepsilon$

$R_{12}^1 = R_{12}^0 + R_{11}^0 (R_{11}^0)^* R_{12}^0$

$= 0 + \varepsilon (\varepsilon)^* 0$

$= 0 + \varepsilon = 0$

$R_{13}^1 = R_{13}^0 + R_{11}^0 (R_{11}^0)^* R_{13}^0$

$= 1 + \varepsilon (\varepsilon)^* 1$

$= 1 + \varepsilon = 1$

$R_{21}^1 = R_{21}^0 + R_{21}^0 (R_{11}^0)^* R_{11}^0$

$= 0 + 0 (\varepsilon)^* \varepsilon$

$= 0$

$R_{22}^1 = R_{22}^0 + R_{21}^0 (R_{11}^0)^* R_{12}^0$

$= \varepsilon + 0 (\varepsilon)^* 0$

$= \varepsilon + 00 = 00 + \varepsilon$

$R_{23}^1 = R_{23}^0 + R_{21}^0 (R_{11}^0)^* R_{13}^0$

$= 1 + 0 (\varepsilon)^* 1$

$= 1 + 01$

$R_{31}^1 = R_{31}^0 + R_{31}^0 (R_{11}^0)^* R_{11}^0$

$= \phi + \phi = \phi$

$R_{32}^1 = R_{32}^0 + R_{31}^0 (R_{11}^0)^* R_{12}^0$

$= 1 + \phi = 1$

$R_{33}^1 = R_{33}^0 + R_{31}^0 (R_{11}^0)^* R_{13}^0$

$= (0+\varepsilon) + \phi = 0 + \varepsilon$

$\boxed{k=2}$

$R_{13}^2 = R_{13}^1 + R_{12}^1 (R_{22}^1)^* R_{23}^1$

$= 1 + 0 (00+\varepsilon)^* (1+01)$

$= 1 + 0 (00)^* (1+01)$

$= 0 (00)^* (1+01) + 1$

$R_{33}^2 = R_{33}^1 + R_{32}^1 (R_{22}^1)^* R_{23}^1$

$= (0+\varepsilon) + 1 (00+\varepsilon)^* (1+01)$

$= (0+\varepsilon) + 1 (00)^* (1+01)$

$= 1 (00)^* (1+01) + 0 + \varepsilon$

$k=3$

$R_{13}^3 = R_{13}^2 + R_{13}^2 (R_{33}^2)^* R_{33}^2$

$= 0(00)^* (1+01) + 1 + 0(00)^* (1+01) + 1 \left((0+\varepsilon) + 1 (00)^* (1+01) \right)^* 1(00)^* (1+01) + 0 + \varepsilon$

$= \left(0(00)^* (1+01) + 1\right) \left(1(00)^* (1+01) + 0\right)^*$

$= (0^* 1) \left(1 (00)^* (1+01) + 0\right)^*$

$= (0^* 1) (10^* 1 + 0)^*$ //

Identify the type:

First check from Type 3

① $S \to Aa$ = Type 3

② $A \to c \mid BaB$

$A \to c$ = Type 3

$A \to BaB$ = Type 2

Type 2 is the higher level so it is the context free grammar

③ $B \to abc$ = Type 2

---

① $S \to ASB \mid d$

$S \to ASB$ = Type 2

$S \to d$ = Type 3

② $A \to aA$ = Type 3

context free grammar (Type 2)

---

① $S \to 0SA2$ = Type 2

② $S \to 012$ = Type 2

③ $2A \Rightarrow A2$ = Type 1

④ $1A \to 11$ = Type 1

context sensitive grammar (Type 1)

If we add

⑤ $A \to \varepsilon$

Un Restricted grammer (Type 0)

---

① $S \to aBBC \mid aBc$

$S \to aBBC$

$S \to aBC$

② $CB \to BC$

③ $aB \to ab$

④ $bB \to bb$

⑤ $bC \to bc$

⑥ $cC \to cc$

context sensitive grammer (Type 1)

⑦ $C \to \varepsilon$

unrestricted grammer. (Type 0)

---

① $S \to aBa \mid bSb$

$S \to aBa$

$S \to bSb$

$S \to a \mid b$

$S \to \lambda$

context free grammer (Type 2)

29/9/2022

Context Free Grammer:

$\rightarrow$ production

$\Rightarrow$ derivation

$G = (N, T, P, S)$

$\alpha \rightarrow \beta$

$\beta \rightarrow (N \cup T)^*$

Context Free Languages:

DERIVATION:

① id+id

$E \Rightarrow E+E$

$\xrightarrow{lmd}$ id+E

$\xrightarrow{lmd}$ id+id

$E \overset{*}{\Rightarrow}$ id+id

$E \overset{lmd}{\Rightarrow} E*E$

$\overset{lmd}{\Rightarrow}$ E+E*E

$\overset{lmd}{\Rightarrow}$ id+E*E

$\overset{lmd}{\Rightarrow}$ id+id*E

$\overset{lmd}{\Rightarrow}$ id+id*id

② id+id*id

$E \overset{lmd}{\Rightarrow} E+E$

$\overset{lmd}{\Rightarrow}$ id+E

$\overset{lmd}{\Rightarrow}$ id+id (E*E)
NO

$\overset{lmd}{\Rightarrow}$ id+id*E

$\overset{lmd}{\Rightarrow}$ id+id*id

lmd - left most derivation

Right most (Rmd)

$E \overset{rmd}{\Rightarrow} E+E$

$E \overset{rmd}{\Rightarrow} E+id$

$E \overset{rmd}{\Rightarrow} id+id$

③ $E \overset{rmd}{\Rightarrow} E+E*E$

$\overset{rmd}{\Rightarrow}$ E+E*id

$\overset{rmd}{\Rightarrow}$ E+id*id

$\overset{rmd}{\Rightarrow}$ id+id*id

④ $E \overset{rmd}{\Rightarrow} E*E$

$\overset{rmd}{\Rightarrow}$ E*id

$\overset{rmd}{\Rightarrow}$ E+E*id

$\overset{rmd}{\Rightarrow}$ E+id*id

$\overset{rmd}{\Rightarrow}$ id+id*id

$E \xrightarrow{d} w$

w forms a language

$$\boxed{CFL = \{ w \mid w \in T^*, S \xrightarrow{d} w \}}$$

Terminal

sentinal Form:

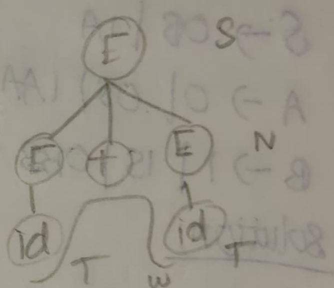
① right sentinal form
② left sentinal form.

⇒ LMD is a left sentinal form.

⇒ RMD is a Right sentinal Form

PARSE TREE:

$$\left.\begin{aligned} E &\Rightarrow E+E \\ &\Rightarrow id+E \\ &\Rightarrow id+id \end{aligned}\right\}$$



→ start symbol will be the root of the parse tree

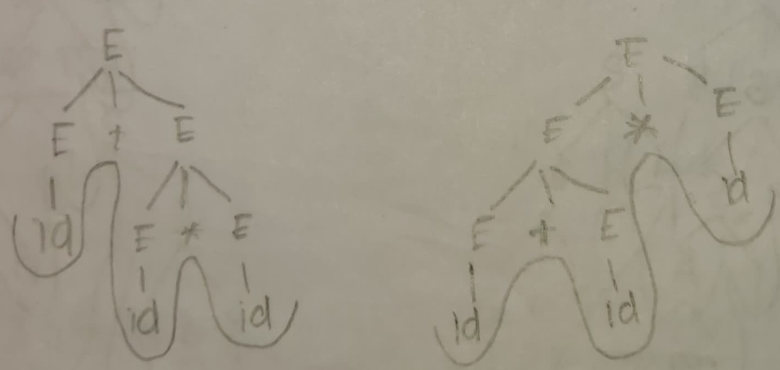
→ Root node always start symbol.

→ Intermediate node always Non-Terminal

→ Leafnodes are always Terminal.

→ When we go through the leafnodes left to Right it will forms a word.

$E \Rightarrow E+E$ $\qquad$ $E \Rightarrow E*E$



Ambiguous

IN a Grammer if we have more than 1 Left most derivations, more than one right most derivations, more than one parse tree then it is called Ambiguos Grammer.

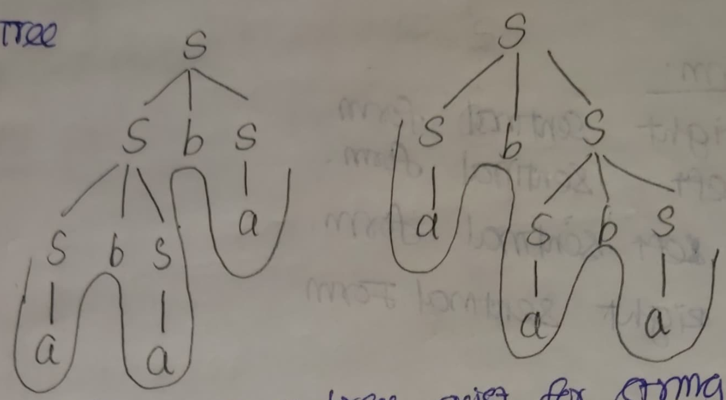
H.W

$S \rightarrow SbS \mid a$ ababa

Check whether the grammer is ambiguous or not?

Solution:

Parse Tree



Since two different parse trees exist for string ababa,

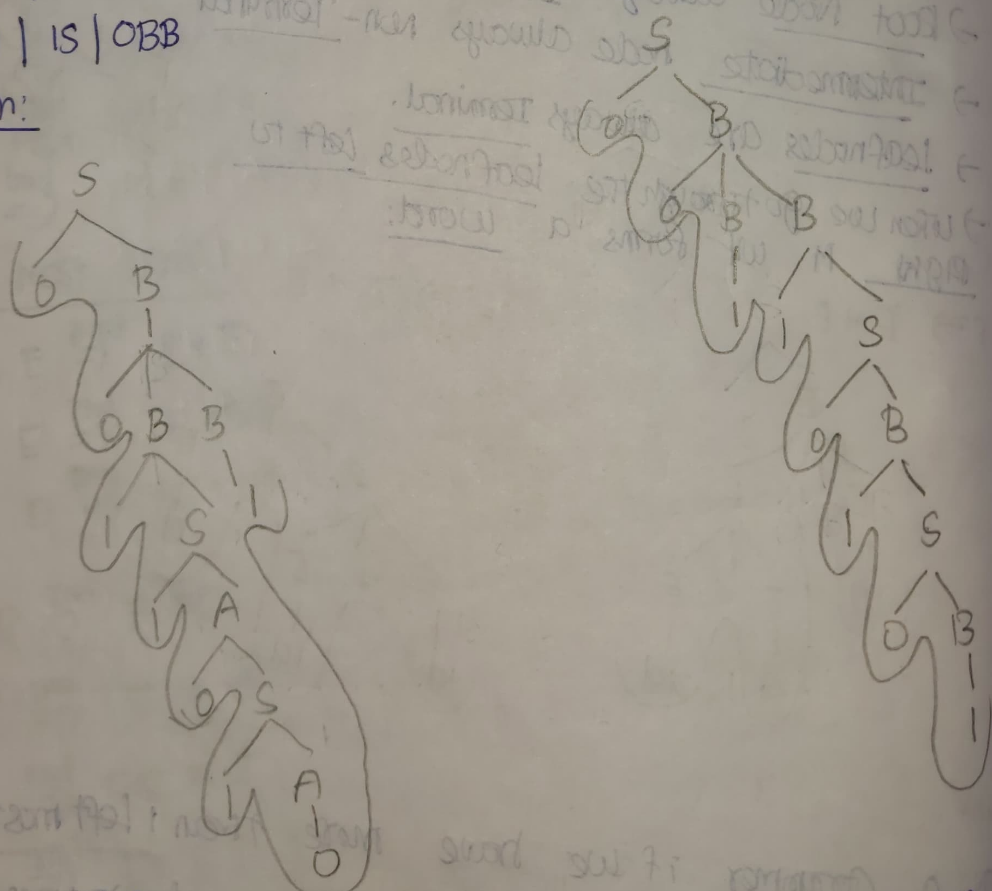
$\therefore$ the given grammer is ambiguous.

$S \rightarrow 0B \mid 1A$

$A \rightarrow 0 \mid 0S \mid 1AA$ $w = 00110101$

$B \rightarrow 1 \mid 1S \mid 0BB$

Solution:



Since two different parse trees exist for string 00110101

$\therefore$ the given grammer is ambiguous.

6|10|2022

Tutorial - 2

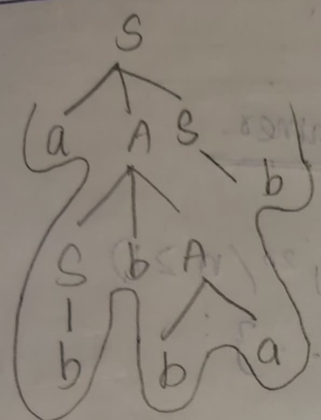
① $G = (\{S, A\}, \{a, b\}, P, S)$

$S \rightarrow aAS \mid b$

$A \rightarrow SbA \mid ba$

$w = abbbab$

Parse Tree:



Derivation:

$S \overset{lm}{\Rightarrow} aAS$

$\overset{lm}{\Rightarrow} aSbAS$

$\overset{lm}{\Rightarrow} abbAS$

$\overset{lm}{\Rightarrow} abbbaS$

$\overset{lm}{\Rightarrow} abbbab$

$S \overset{rm}{\Rightarrow} aAS$

$\overset{rm}{\Rightarrow} aAb$

$\overset{rm}{\Rightarrow} aSbAb$

$\overset{rm}{\Rightarrow} aSbbab$

$\overset{rm}{\Rightarrow} abbbab$

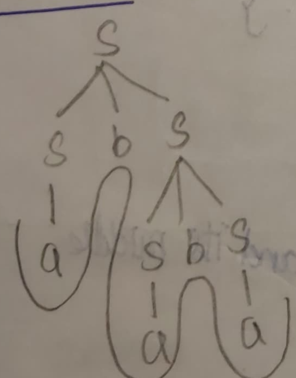
② G is a grammer

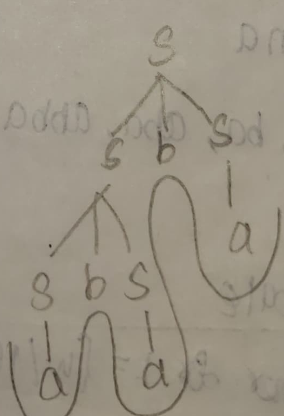
$S \rightarrow SbS \mid a$

$w = ababa$

G is ambiguous ?

Parse Tree:





Derivation:

$S \overset{lm}{\Rightarrow} SbS$

$S \overset{lm}{\Rightarrow} SbSbS$

$S \overset{rm}{\Rightarrow} SbS$

$S \overset{rm}{\Rightarrow} SbSbS$

$S \xRightarrow{lm} SbSbS$ $\quad$ $S \xRightarrow{rm} SbSba$

$\xRightarrow{lm} abSbS$ $\quad$ $S \xRightarrow{rm} Sbaba$

$\xRightarrow{lm} ababS$ $\quad$ $S \xRightarrow{rm} ababa$

$\xRightarrow{lm} ababa$

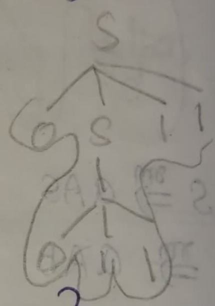
It have 2 parse trees & 2 leftmost & rightmost derivations. It is ambiguous.

8/10/2022

Context Free Grammer

Example

1) construct a grammer for $L = \{0^n 1^{2n} / n \geq 1\}$



$L = \{011, 001111, 000111111 \ldots\}$

G: $S \to 0S11 \mid 011$

2) $L = \{w \mid |w| \text{ is odd over } \{0,1\}\}$

$L = \{0, 1, 001, 010, 111, 000, 011 \ldots\}$

G:
$S \to 0A \mid 1A$
$A \to 0S \mid 1S \mid \varepsilon$

3) construct a grammer to generate the set of all strings over $\Sigma = \{a, b\}$ ending in a

$L = \{a, aa, ba, aba, abba, bba \ldots\}$

G: $S \to Aa$
$A \to Aa \mid bA \mid \varepsilon$

4) construct a grammer for $L = \{w / |w| \text{ is odd and its middle symbol is } 0\}$ over $\{0,1\}$

$L = \{0, 000, 001, 101, 00000 \ldots\}$

G: $S \to 0S0 \mid 0S1 \mid 1S0 \mid 1S1 \mid 0$

... 001111, 00

5) What is language generated by the grammar.

$S \to aSb$

$S \to ab$

$L = \{a^n b^n \mid n \geq 1\}$

---

6) $L = \{w \mid w$ that starts & ends with the same symbol$\}$ over $\{0,1\}$

$L = \{0, 1, 00, 11, 010, 101, 1001 \ldots\}$

G: $S \to 0A0 \mid 1A1 \mid 0 \mid 1$, $A \to 1A \mid 0A \mid \varepsilon$

---

7) $L = \{(ab)^n / n \geq 1\} \cup \{(ba)^n / n \geq 1\}$

$L = L_1 L_2$

G1 G2

G1 G2

A B

A →

B →

G: $S \to AB$

$S \to A \mid B$

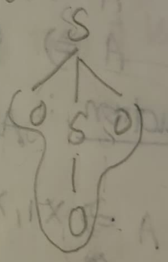
G1 {A G2 {B

---

8) construct grammer for set of all palindrome over the alphabet $\Sigma = \{0,1\}$

000

$L = \{0110, 1001, 1111, 0000, 010 \ldots\}$



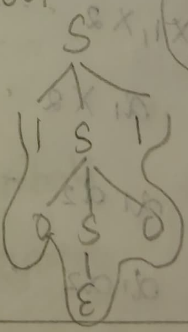
1001

G: $S \to 0S0 \mid 1S1$

$S \to 0$

$S \to 1$

$S \to \varepsilon$



Derivation:

The sequence of replacements of nonterminals is called derivation.

1) Left most derivation $\overset{lm}{\Rightarrow}$

2) rightmost derivation $\overset{rm}{\Rightarrow}$

Parse Tree (Derivation Tree):

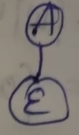
→ The label can be a terminal or non terminal

→ The root should be start symbol.

→ Intermediate nodes are non terminals.

→ Leaf nodes are Terminals.

→ $A \to \varepsilon$ ε will be the one child of A.



Relationship b/w Derivation & Derivation Tree:

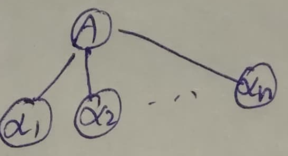
$G = \{N, \Sigma, P, S\} \Rightarrow$ CFG

$S \Rightarrow^* \alpha$ & if and only if there is a derivation tree for G which yield α.

Part-1: If there is parse tree then we have a derivation.

Proof:

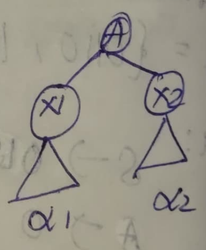
Base: height 1



$A \to \alpha_1, \alpha_2 \dots \alpha_n$

$A \Rightarrow \alpha_1, \alpha_2 \dots \alpha_n$

Induction: Assume Tree with height > 1



$A \Rightarrow X_1, X_2 \dots X_n$

$\overset{*}{\Rightarrow} \alpha_1 X_2 \dots X_n$

$\overset{*}{\Rightarrow} \alpha_1 \alpha_2 \dots X_n$

$\overset{*}{\Rightarrow} \alpha_1 \alpha_2 \dots \alpha_n$

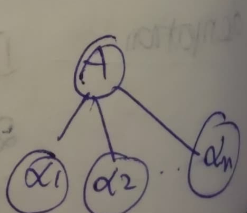
$\boxed{\alpha A \overset{*}{\Rightarrow} \alpha}$

Part-2: If there is a derivation then there is a parse tree.

Proof:

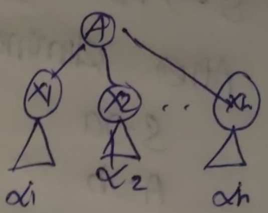
Base: production

$A \to \alpha_1 \alpha_2 \dots \alpha_n$



Induction:

Assume



$A \Rightarrow X_1 X_2 \dots X_n$

$\overset{*}{\Rightarrow} \alpha_1 \; X_2 \dots X_n$

$\overset{*}{\Rightarrow} \alpha_1 . \alpha_2 , \dots X_n$

$\overset{*}{\Rightarrow} \alpha_1 \alpha_2 \dots \alpha_n$

$A \overset{*}{\Rightarrow} \alpha$

10/10/2022

Simplication of context free grammer

Three ways:

Eliminating useless symbols:

Find useful symbols

-> It is a symbol is generating it is useful symboll

=> It is a symbols is reachable It is usefaul symbol.

$X \overset{*}{\Rightarrow} w$

for some $w \in T^*$

-> If a symbol is generating reachable then it is usefull symbol.

-> If not it is a use less symbol.

Algorithm:

-> Eliminate the non generat symbols

-> Eliminate the non reachable symbols

Example

① $S \to AB \mid a$

$A \to b$

Sol:

$A \to b \quad S \to a$ is generating

$A \to aAa \mid aa \mid B \mid C$

$S \to ACA \mid CA \mid AA \mid AC \mid A \mid C \mid \varepsilon$

$C \to \varepsilon$

---

③ $S \to aS \mid SS \mid bA$

$A \to BB$

$B \to ab \mid aAbC \mid aAb \mid CC$

$C \to \varepsilon$

1) $C \to \varepsilon$ (eliminate)

$B \to CC$

$B \to aAbC$

$A \to BB$

$S \to bA$

$S \to aS$

$S \to SS$

$B \to aAb$

2) The new Grammer

$S \to aS \mid SS \mid bA$

$A \to BB$

$B \to \varepsilon \mid ab \mid aAb$

3) eliminate $B \to \varepsilon$ by replacing

$A \to BB$ into $A \to BB, A \to B, A \to \varepsilon$

4) The new grammer is

$S \to aS \mid SS \mid bA$

$A \to BB \mid B \mid \varepsilon$

$B \to ab \mid aAb$

④ $S \to aS \mid SS \mid bA$

$A \to BB \mid B \mid \varepsilon$

$B \to ab \mid aAb$

1) Nullable Non terminals

$A \to \varepsilon$

$B \to aAb$

$S \to bA$

$S \to aS$

$S \to SS$

$A \to BB$

$A \to B$

2) The final CFG is

$S \to aS \mid SS \mid bA \mid b$

$A \to BB \mid B$

$B \to ab \mid aAb$

1. $S \to AB$ CNF
   $S \to \varepsilon$ CNF
   $A \to a$ CNF
   $B \to b$ CNF

Convert a grammar to CNF:

1. No ε productions (other than $S \to \varepsilon$)
   No unit productions
   No useless symbols

2. Eliminate terminals from RHS of productions

$A \to X_1 X_2 \ldots X_m$

where $X_i \in N \cup T$ $\to X_1$

($A \to aBbcd$ for each terminal we have to create a new non terminal)

$\boxed{A \to C_aBC_bCC_d}$
$C_a \to a$
$C_b \to b$
$C_d \to d$

3. Eliminate productions with long RHS

$A \to X_1 X_2 \ldots X_m$

$A \to X_1 D_1$
$D_1 \to X_2 D_2$
$D_2 \to X_3 D_3$
$\vdots$
$D_{m-2} \to D_{m-1} D_m$

$A \to C_aBC_bCC_d$
~~$A \to C_aBC_bAC_d$~~
$A \to C_aD_1$
$D_1 \to BD_2$
$D_2 \to C_bD_3$
$D_3 \to AC_d$

Example:

1. $S \to aAbB$
   $A \to aA \mid a$
   $B \to bB \mid b$

I) ε production ×
   Unit ×
   Useless ×

(4) $S \to aSa \mid bSb \mid c$

L = {palindrome middle element is c)

1) No ε productions, No useless symbols, No unit production x

2) eliminate terminals

$c \to C_c$, $a \to C_a$, $b \to C_b$

$S \to C_a S C_a \mid C_b S C_b \mid c$

3) Eliminate productions with long RHS

It is alredy in CNF

So, the answer is

$S \to C_a S C_a \mid C_b S C_b \mid c$

(5) No chain rules, useless symbols, or ε-productions

$S \to aXYZ \mid a$ $\quad X \to aX \mid a$

$Y \to bcY \mid bc$ $\quad Z \to cZ \mid c$

1) $S \to AXYZ \mid a$

$X \to AX \mid a$

$Y \to BCY \mid BC$ $\quad$ $A \to a$
$Z \to CZ \mid c$ $\quad$ $B \to b$
$C \to c$

2) $S \to AF \mid a$
$X \to AX \mid a$ $\quad$ $A \to a$ $\quad$ $F \to XG$
$Y \to BH \mid BC$ $\quad$ $B \to b$ $\quad$ $G \to YZ$
$Z \to CZ \mid c$ $\quad$ $C \to c$ $\quad$ $H \to CY$

12/10/2022

Greibach Normal Form

Convert CFG to GNF:

It defines about the right hand side of the production.

- $A \to a\alpha$ ⇒ non terminal * (N^*) $\quad$ $\alpha \leftarrow \varepsilon$ also
  (a ⇓ Terminal) $\quad$ $\alpha \in N^*$
- $A \to a$
- $S \to \varepsilon$ only if S doesn't appear on any right hand side of any other production. $\underline{\varepsilon \in L}$

Ex:

① $S \to aAB \mid \varepsilon$
$A \to bC$
$B \to b$

GNF only

② $G_1 = \{\{S, A\}, \{a, b\}, S, \{S \to aSA \mid a, A \to aA \mid b\}\}$

P

$S \to aSA$
$S \to a$
$A \to aA$
$A \to b$

} GNF

③ $G_2 =$

④ $S \to AS$ NO

$S \to AAS$ NO

$A \to SA$ NO

$A \to aa$ NO

} not GNF

⑤ $S \to AB$ ×

$A \to aA \mid bB \mid b$ ✓

$B \to b$ ✓

} Not in GNF

⑥ $S \to aAB \mid bBB \mid bB$

$A \to aA \mid bS \mid b$

$B \to b$

} GNF

Theorem: (Immediate left recursion):

① Need to check the grammer ~~is in~~ CNF. If not convert it into CNF and then do the process.

$A_1, A_2 \dots A_n$ with $S = A_1$

② Derive the productions of the form

$A_i \to a\gamma$ or

$A_i \to A_j \gamma$ where $j > i$

$A_1 \to a A_2$

$A_1 \to A_2 A_3 \dots$ } $j > i$

$A_i \to A_i \gamma$

③

$A_i \to a\gamma$ from $A_i \to A_i \gamma$ is called left recursion

$\boxed{\text{mathematical relation for left recursion}}$

$A \to A\alpha_1 \mid A\alpha_2 \dots \mid \beta_1 \mid \beta_2$

* Eliminate left recursion.

① $A \to \beta_1 \mid \beta_2$

② $A \to \beta_1 B \mid \beta_2 B \mid \beta_3 B \dots$

③ $B \to \alpha_1 \mid \alpha_2 \dots$

④ $A_i \to a\gamma$ for $i = 1, 2 \dots n-1$

(5) $B_i \to a\gamma$

(4) $\alpha_1 B \mid \alpha_2 B \; \ldots$

Example:

(1) $S \to AA \mid a$

$A \to SS \mid b$

$A \to AaB \mid Ab \mid a \mid b$

$B \to a$

Solution

1) ✓CNF

$\left.\begin{matrix} S & A_1 \\ A & A_2 \end{matrix}\right\}$

$A \to a \mid b$

$A \to aB_1 \mid bB_1$

$B_1 \to aB \mid b$

$B_1 \to aBB \mid b$

$A_1 \to A_2 A_2 \mid a$

$A_2 \to A_1 A_1 \mid b$

2) $A_i \to a\gamma$

$A_i \to a_j \gamma$

$A_1 \to A_2 A_2 \mid a$

$A_2 A_2 A_1 \mid aA_1 \mid b$

$A_2 \to \underline{A_1} A_1 \mid b$

for A_1 substitute

$A_2 \to A_2 A_2 A_1 \mid aA_1 \mid b$

$A_i \to A_i \gamma$ left recursion

3) $A_2 \to A_2 A_2 A_1$ eliminate left recursion

as the whole set we can eliminate the left set of left recursion then it (get value) recursion

$A_2 \to \underbrace{A_2}_{} \underbrace{A_2 A_1}_{\alpha} \mid \underbrace{aA_1}_{\beta_1} \mid \underbrace{b}_{\beta_2}$

$A_2 \to b$

$A_2 \to aA_1$

$A_2 \to A_2 A_2 A_1$

$A_2 \to aA_1 \mid b$

$A_2 = A_1 B_2 \mid bB_2$

$B_2 \to A_2 A_1$

$B_2 \to A_2 A_1 B_2$

When we eliminate left recursion we get the relation in all.

$A_2 \to$ GNF

1) $A_1 \to A_2 A_2 | a$

$A_2 \to a A_1$
$A_1 \to \lambda \ldots \lambda | a$
$A_2 \to a A_1 | b$
$A_2 \to a A_1 B_2 | b B_2$

① $A_2 \to a b_1 | b$ ..
② $A_2 \to a b$, B_{us} $(b B_2$
③ $a A_1 A_1 | b A_2 | a A f_2 A_2 |$
$b | b_1 A_3] \varepsilon$

③ Modify B_i - productions:

$B_2 \to A_2 b | A B_2$

$B_2 \to$ 8 productions

$B_2 \to a A_1 A_2 | b A_1 | a A_1 B_2, A | b A_2 A_1$
$a_1 A_1 B_2 A | B b$;

② $S \to AB$

$A \to BS | b$

$B \to SA | a$

Convert this CFG to GNF:

Solution:

1) CNF

2) $S \quad A_1$
$A \quad A_2$
$B \quad A_3$

$A_1 \to A_2 A_3 \quad j > i$

$A_2 \to A_3 A_1 | b$ $j > i$

$A_3 \to A_{1_j} A_2 | a$ $j \not> i$ If not add the states

$A_3 \to A_2 A_3 A_2 | a$ $A_3 \to A_3 A_1 A_3 A_2 | a$

3) $\boxed{A_i \to a\gamma}$ $\boxed{A_i \to \alpha_j \gamma}$

$A_1 \to A_2 A_3$ & $A_2 \to A_3 A_1 | b$ & $A_3 \to A_3 A_2 | a$ ($A_3 A_2$)

3) $\left.\begin{array}{l} A_1 \to A_2 A_3 \\ A_2 \to A_3 A_1 | b \\ A_3 \to A_1 A_2 \end{array}\right\}$ eliminate left recursion

$A_2 \to \underset{\alpha}{A_3} \underset{\beta}{A_1} | b$ $A_3 \to A_3 A_2 | a$ ($A_3 A_2$)

$A_1 - 5$
$B_3 - 10$
$A_2 - 5$
$A_3 = 4$

$A_1 \to A_2 A_3$

$A_3 \to A_1 A_2$

$A_3 \to A_2 A_3 A_2 | a$

$A_3 \to A_3 \underbrace{A_1 A_3 A_2}_{\beta} | b A_3 A_2 | a$ α

αB_3

B_i

$A_3 \to b A_3 A_2 | a$

$A_3 \to b A_3 A_2 B_3 | a B_3$

$B_3 \to A_1 A_3 A_2$

$B_3 \to A_1 A_3 A_2 B_3$

4) $A_1 \to A_2 A_3$

$A_1 \to A_3$

$L = \{01, 0011, 000111, \dots\}$

First symbol always 0

(3 states is the easy method)

$\delta = \{(q_0, a, z)\} = \{(q_0, 0z)\}$

$\delta(q_0, 0, z)$

$\delta($

1) $L = \{0^n 1^{2n} \mid n \geq 0\}$

2) $L = \{a^n b^m c^n \mid m,n \geq 1\}$

→ push 2 0's at a time

→ at least we have 1 b

→ 4 states