

22/8/20

U2  
=

## Grammar - Introduction

\* Context Free Grammar and denoted by  
 $G_L = (V, T, P, S)$

\* The main objective is to frame the rules for certain languages.

$$G_L = (V, T, P, S)$$



V  $\Rightarrow$  Variables or non terminal

T  $\Rightarrow$  Terminals or small letters or special characters

P  $\Rightarrow$  Production rule

S  $\Rightarrow$  starting symbol

\* The language generated by CRG is CFL (Context Free language).

2)

Let Grammar  $G_L = \{ S^y, Sa, b^y, P, S^y \}$

$$P = (S \rightarrow aSa$$

$$S \rightarrow bSb$$

$$S \rightarrow e )$$

Find the lang. generated by this grammar.

$w = aa$

$$\text{I. } S \xrightarrow{\quad} aSa \\ S \xrightarrow{\quad} a\epsilon a \quad [S \xrightarrow{\quad} \epsilon]$$

$$aSa \xrightarrow{\quad} aa$$

↑ step of 1st move of S

II.

$$S \xrightarrow{\quad} aSa \\ S \xrightarrow{\quad} abSba \\ S \xrightarrow{\quad} ab\epsilon ba$$

$$\xrightarrow{\quad} \frac{abba}{w w^R}$$

$$L = \{ww^R / w \in \{a|b\}^*\}$$

→ L is a context-free lang. because it is accepted by NFA

→ L is a regular lang. because it is accepted by DFA

Type ③ → if grammar has exactly one NT & it is accepted by its generative veg. lang. & it is accepted by

FAs. & type ③ grammar must have a

single non-terminal on RHS.

$$(T-3) \quad x \xrightarrow{\quad} a \\ x \xrightarrow{\quad} a y$$

$$x, y \in N(V)$$

$$a \in T$$

→ if grammar has exactly one NT

(and) exactly one rule right R

T-2

$$\boxed{T-2} \quad \boxed{\alpha A \beta \rightarrow \alpha' \gamma \beta'}$$

(Accepted by LBA)

$$A \rightarrow Y$$

$$A \in N$$

$$Y \in (TUN)^*$$

$$A \in N, \alpha, \beta, Y \in (TUN)^*$$

$\alpha \neq \beta \rightarrow$  may be empty

$Y \rightarrow$  must not be empty

T-0

$$\boxed{\alpha \rightarrow \beta}$$

$\rightarrow$  It is a phrase structured grammar or recursively enumerable language.

$\alpha \rightarrow$  terminals and non-terminals with atleast one non-terminal and ' $\alpha$ ' can't be null.

$\beta \rightarrow$  Both terminals and non-terminals.

$$T_3 \subseteq T_2 \subseteq T_1 \subseteq T_0$$

Derivations and Languages :-

1. Recursive inference

2. Derivations.

Derivation types:-

1. left-most derivations (LMD)

2. right-most derivations (RMD)

\* LMD

→ In each step in derivation, the left-most variable is replaced by its production.

→ represented by ;

$$\xrightarrow[\text{Lm}]{\quad} \xrightarrow[\text{Lm}]{\quad} \xrightarrow[\text{Lm}]{\quad}$$

Ex:

$$S \rightarrow aA \quad S/a$$

$$A \rightarrow ab$$

$$w = aaba$$

$$S \rightarrow aAS$$

$$\xrightarrow[\text{Lm}]{\quad} aabs \quad [A \rightarrow ab]$$

$$\xrightarrow[\text{Lm}]{\quad} aaba \quad [S \rightarrow a]$$

\* RMD

$$S \rightarrow aAS$$

$$S \rightarrow aAa \quad [S \rightarrow a]$$

$$S \rightarrow aaba \quad [A \rightarrow ab]$$

"aabbaabba"

$$S \rightarrow aB/bA$$

$$S \rightarrow a/aS/bAA$$

$$B \rightarrow b/bS/aBB$$

LMD

$$S \rightarrow aB \quad \cancel{aA}$$

$$S \xrightarrow[\text{Lm}]{\quad} aABb \quad [B \rightarrow aBB]$$

$$S \xrightarrow[\text{Lm}]{\quad} aabB \quad [B \rightarrow b]$$

$$S \xrightarrow[\text{Lm}]{\quad} aabbS \quad [B \rightarrow bS]$$

$S \xrightarrow{Im} aabbS [B \rightarrow bS]$

$S \xrightarrow{Im} aabbAB [S \rightarrow aB]$

$S \xrightarrow{Im} aaBBabS [B \rightarrow bS]$

$S \xrightarrow{Im} aaBBabbA [S \rightarrow bA]$

$S \xrightarrow{Im} aabbabba [A \rightarrow a]$

$S \xrightarrow{Im} aabbabba$



Derivation

RMD

aabbabba

$S \rightarrow aB$

$S \xrightarrow{RMD} aaB\underline{B}$

$S \xrightarrow{RMD} aaBbb\underline{A}$

$S \xrightarrow{RMD} aaB\underline{bba}$

$S \xrightarrow{RMD} aab\underline{S} bba$

$S \xrightarrow{RMD} aab\underline{bA} bba$

$S \xrightarrow{RMD} aabb\underline{babba}$

Derive the string 1000111 for left & leftmost derivation using CFG

$G_1 = (V, T, P, S)$  where

$$V = \{S, T\}$$

$$T = \{0, 1\}$$

$$P: S \rightarrow T00T$$

$$T \rightarrow 0T / \cdot \quad |T$$

$$(S \rightarrow T) \cdot \cdot \cdot S * S \cdot \cdot \cdot$$

Ambiguity

$$(S \rightarrow T) \cdot \cdot \cdot T * S * S \cdot \cdot \cdot$$

Ambiguous Grammar

$$E \rightarrow E + E \mid E^* E \mid id.$$

Construct ambiguous grammar for the following

$$E \rightarrow I \quad E \rightarrow E^* E$$

$$E \rightarrow E + E \quad E \rightarrow (E)$$

restrictions: no two terms with parentheses \(\downarrow\) (i)

$$w = 3 * 2 + 5$$

restriction: two terms with parentheses \(\downarrow\) (ii)

LHD:

~~any start symbol must have a leftmost derivation with 0~~

$$E \rightarrow I * E \quad (E \rightarrow I)$$

$$E \rightarrow I * E + E \quad (E \rightarrow E + E)$$

$$E \rightarrow I * I + E \quad (E \rightarrow I)$$

$$E \rightarrow I * I + I \quad (E \rightarrow I)$$

$$E \rightarrow E + E$$

$$E \rightarrow E * E + E \quad (E \rightarrow E * E)$$

$$E \rightarrow I * E + E \quad (E \rightarrow I)$$

$$E \rightarrow I * I + E \quad (E \rightarrow I)$$

$$E \rightarrow I * I + I \quad (E \rightarrow I)$$

$$E \rightarrow I * I + I \quad (I \rightarrow 3)$$

$$E \rightarrow 3 * I + I \quad (I \rightarrow 2)$$

$$E \rightarrow 3 * 2 + I \quad (I \rightarrow 5)$$

$$E \rightarrow 3 * 2 + 5 \quad (I \rightarrow 5)$$

$$\underline{3 * 2 + 5 \leftarrow E}$$



parallel to the removal of useless transitions  
Simplification of CFG

$$3 * 2 + 5 \leftarrow I \leftarrow 3$$

(i)  $\hookrightarrow$  Eliminating useless production

(ii)  $\hookrightarrow$  Removing / eliminating the epsilon production

(iii)  $\hookrightarrow$  Removing / eliminating the unit production.

The productions form a grammar that can never take part in any derivations.

$$\text{Ex: } S \rightarrow aSbS/c/e \quad (S \rightarrow aSbS) \rightarrow S \rightarrow c/e$$

$$C \rightarrow aC \quad (C \rightarrow aC) \rightarrow C \rightarrow a$$

$$\underline{\text{Solutions}}) \quad S \rightarrow a^+b^+c^+e^+ \quad (S \rightarrow a^+b^+c^+) \rightarrow S \rightarrow e^+$$

Q) eliminate the useless symbols and useless production  
are from  $G = (V, T, P, S)$  where

$$V = \{S, A, B, C\}$$

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

$$S \in V$$

$$P:$$

$$S \rightarrow aS/A/C$$

(present) (initial) (final)

$$B \rightarrow aa$$

$$C \rightarrow aC$$

to make all the  $\alpha$  in  $\beta$  in  $\alpha\beta$  has  $A$ .

Solution:  $S \rightarrow aS/A/a/aa$

(d) initial  $\leftarrow$  (or)  $A$

$$S \rightarrow aS/A/a/A/B/aa$$

$$\leftarrow B \leftarrow C \quad ?$$

ii) Any production of CFG in the form of  $A \rightarrow \epsilon$

$A \rightarrow \epsilon$  is called epsilon production

Q)  $S \rightarrow aS \cancel{A} \epsilon$

$$1.) S \rightarrow aB$$

$$B \rightarrow B\epsilon$$

$$\text{I} \quad \text{Variable} = \{B\}$$

$$\text{II} \quad S \rightarrow abB/ab$$

$$a) B \rightarrow Bb$$

Q)  $S \rightarrow BabC$

$$C \rightarrow b\epsilon$$

$$B \rightarrow a\epsilon$$

$$\Sigma \text{ Vnullabn} = \{B, C\}$$

$$\text{ii) } S \rightarrow Babc | abc | Bab | ab$$

$$c \rightarrow b \text{ (removal of c)}$$

$$B \rightarrow a$$

Step 2:  $c \rightarrow b$

(iii) Unit Productions: (Eliminating / Removing)

A production in a CFG in the form of

$$A \rightarrow B$$

where

$$A, B \rightarrow \text{variable (V)}$$

Step 3:  $A \rightarrow B \rightarrow C \rightarrow \dots$

$$\text{Q) } S \rightarrow AaB|C$$

$A \rightarrow a|bc|B$  is part of removing part Q

conditions:  $C \rightarrow a$

$$B \rightarrow A|bb$$

Step 4:  $C \rightarrow a$

$$\Sigma, S \rightarrow c$$

$$A \rightarrow B$$

$$B \rightarrow A$$

Step 5:  $S \rightarrow AaB|a$

$$A \rightarrow a|bc|bb$$

$$C \rightarrow a$$

$$B \rightarrow a|bc|bb$$

Q) unit production (eliminate all the useless production and  
null)

$$S \rightarrow aA/aBB$$

$$A \rightarrow aaA/\epsilon$$

$$B \rightarrow bB/bbC$$

$$C \rightarrow B$$

Step 1: Eliminate  $\epsilon$  production

Step 2: Eliminate null production

Step 3: Eliminate useless production

Final simplified grammar

I. eliminating  $\epsilon$  production

$$V_{\text{nullish}} = \{A\}$$

$$S \rightarrow a/aBB/a$$

$$A \rightarrow aaA/aa$$

$$B \rightarrow bB/bbC$$

$$C \rightarrow B$$

$$S \rightarrow aA/aBB/a$$

$$A \rightarrow aaA/aa$$

$$B \rightarrow bB/bbC$$

$$C \rightarrow bB/bbC$$

Step 1: Eliminate  $\epsilon$  production

Step 2: Eliminate null production

Step 3: Eliminate useless production

iii removing productions:

$$\boxed{\overline{S} \rightarrow \overline{aA} \overline{a}}$$

$$S \rightarrow aA/a$$

$$A \rightarrow aaA/aa$$

$$\boxed{\overline{aA} \overline{a} \mid \overline{aaA} \leftarrow a}$$

$$\boxed{\overline{aaA} \leftarrow a}$$

$$\boxed{\overline{a} \mid \overline{aa} \leftarrow a}$$

CNF : Chomsky Normal Form.

$$N.T \rightarrow NT.NT$$

$$N.T \rightarrow T$$

Reducing CPG into CNF :-

To reduce a CPG into CNF, there are some

following steps.

- Step 1 i) eliminate  $\epsilon$   
ii) eliminate null

iii) \* useless production.

Q1) Derivations

$$S \rightarrow AAC$$

$$A \rightarrow aAb/b$$

$$C \rightarrow aC/a$$

connect into CNF:

I v-nullable =  $\{a\}^3$

$$S \rightarrow AAC | AC | C$$

$$S \rightarrow A \rightarrow aAb/b$$

$$S \rightarrow C \rightarrow aC/a$$

II removing unit production  $S \rightarrow C$

$$S \rightarrow AAC | AC | aC | a$$

$$A \rightarrow aAb/b$$

$$C \rightarrow aC/a$$

$$\boxed{C \rightarrow aC/a}$$

III There is no useless production.

IV

P<sub>1</sub>:  $S \rightarrow AAC$  is not a CNF T/F

$$S \rightarrow AAD,$$

$$D_1 \rightarrow AC$$

$$\boxed{A \rightarrow BC \\ A \rightarrow a}$$

P<sub>2</sub>:  $S \rightarrow AC$

$P_3 : S \rightarrow aC$

$S \rightarrow D_2 C$

$D_2 \rightarrow a$

$P_4 : A \rightarrow aAb$

$=$

$A \xrightarrow{Ad_2} D_2 Ab \quad A \xrightarrow{\text{iii}} A \rightarrow D_2 D_3 / D_2 D_4$

$D_3 \rightarrow AD_4$

$A \xrightarrow{\text{iii}} A$

$D_4 \rightarrow b$

$A \xrightarrow{\text{iii}} A$

$P_5 :$

$C \rightarrow aC/a$

$C \rightarrow D_2 C/a$

Q) consider a  $CFG$

$CFG \quad L = \{ s, A \}, \{ a, b \}, P, S$

$s \rightarrow aAS/a$

$A \rightarrow Sba / SS/ba$

$SL(a) \cap \{aa\}^n \neq \emptyset$

$\{a\}^n \neq \emptyset$

$a \mid ab \in S$

Sol:

$$S \rightarrow aAS$$

~~is RT~~

(i)  $M_1 \rightarrow a$

$S \Rightarrow M_1 AS$

$S \rightarrow M_1 M_2$

$M_2 \rightarrow AS$

(ii)  $S \rightarrow a$

(iii)  $A \rightarrow bba$

$M_3 \rightarrow b$

$A \rightarrow SM_3 A$

$A \rightarrow SM_4$

$M_4 \rightarrow M_3 A$

(iv)  $A \rightarrow SS$

(v)  $A \rightarrow ba$

$A \rightarrow M_3 M_1$

CFG to CNF.

Q)  $S \rightarrow ABA$

$A \rightarrow aA / \epsilon$

$B \rightarrow bB / \epsilon$

Step ①

i) eliminate  $\epsilon$

$$S \rightarrow ABA \mid BA \mid AA \mid AB \mid A \mid B$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow bB \mid b$$

(ii) eliminate unit production!

$$\boxed{S \rightarrow A}$$
$$S \rightarrow B$$

$$S \rightarrow ABA | BA | AA | AB | AA | a | bB | b$$

$$A \rightarrow aa | a$$

$$B \rightarrow bB | b$$

(iii) no useless production.

skip ②  $\Leftrightarrow$  CNF (for S)

i)  $S \rightarrow ABA$

~~M<sub>1</sub> → BA~~

$\boxed{S \rightarrow AM_1}$

ii)  $S \rightarrow BA$

~~M<sub>2</sub> → A~~

$\boxed{S \rightarrow M_2A}$

iii)  $S \rightarrow AA$

~~M<sub>2</sub> → a~~

$\boxed{S \rightarrow M_2A}$

iv)  $S \rightarrow AB$

v)  $S \rightarrow a$

vi)  $S \rightarrow bB$

~~M<sub>3</sub> → B~~

$\boxed{S \rightarrow M_3B}$

(for B)

eliminate S to avoid

$\boxed{B \rightarrow M_3B}$

vii)  $S \rightarrow b$

$\boxed{B \rightarrow b}$

(for A)

ix)  $A \rightarrow aa$

$\boxed{A \rightarrow M_2A}$

x)  $A \rightarrow a$

③ Answer

CNF are

$$S \rightarrow A M_1 | AB | BA | AA | M_2 A | M_3 B | a | b$$

$$A \rightarrow M_2 A | a$$

$$B \rightarrow M_3 B | b$$

$$\text{For } M_1 \rightarrow BA$$

$$M_2 \rightarrow a$$

$$M_3 \rightarrow b.$$

GNF  $\Rightarrow$  Grisbert Normal Forms

A grammar  $g$  is  $(V, T, P, S)$  is said to be a GNF, if every production rule is in the form  $A \rightarrow a\alpha$

where  $a \rightarrow \text{terminal}$

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

Lemma ① Substitution Rule

Lemma ② Elimination of left Recursion.

Lemma ③

$$A \rightarrow a_1 B_1 | a_2 B_2 | \dots | a_n B_n$$

Lemma ④

$$(i) \begin{array}{l} A \rightarrow \beta^i \\ A \rightarrow \beta^i x \\ (i^0 \leq i \leq n) \end{array} \quad \left\{ \begin{array}{l} A \rightarrow \beta_1 / \beta_2 / \dots / \beta_n \\ A \rightarrow \beta_1 x / \beta_2 x / \dots / \beta_n x \end{array} \right.$$

$$(ii) \begin{array}{l} x \rightarrow \alpha_i^i \\ x \rightarrow \alpha_i^i x \\ (i^0 \leq i \leq m) \end{array} \quad \left\{ \begin{array}{l} x \rightarrow \alpha_1 / \alpha_2 / \dots / \alpha_n \\ x \rightarrow \alpha_i x \end{array} \right.$$

### GNF Rules :-

(1) Check whether it is CNF  
 (2) If not eliminate the  $\epsilon$ , unit, useless

(3) Rename the variables

Ex:  $A, A_2, \dots$  with  $S = A,$

(4)  $A_i \rightarrow A_j \alpha$  apply the following

(i)  $i < j \rightarrow$  leave the pds as it is

(ii)  $i > j \rightarrow$  substitution rule

(iii)  $i = j \rightarrow$  eliminating of L.R.

(5)  $A_i \rightarrow A_j \alpha$  where  $i < j$  apply sub.rule if  $A_j$  is in GNF

Example  $B, A \leftarrow S = i \cup b = j$

Q) GNF for the "Gr" grammar

$$S \rightarrow AA / a$$

$$A \rightarrow SS / b$$

Sol: (i) Given Gr is CNF

(ii) Rename the variable

$$S \rightarrow A_1 \text{ & } A \rightarrow A_2$$

$$A_1 \rightarrow A_2 A_2 / a$$

$$A_2 \rightarrow A_1 A_1 / b$$

(iii) check  $i^o$  &  $j^o$

$$A_1 \rightarrow A_2 A_2 / a \quad \left\{ i^o < j^o - \text{name as it is} \right.$$

$$\begin{bmatrix} i^o = 1 \\ j^o = 2 \end{bmatrix}$$

$$A_2 \rightarrow A_1 A_1 / b \quad \left\{ i^o = 2, j^o = 1; i^o > j^o \right.$$

applying Substitution rule;  $i^o < j^o$   
not in GNF

$$A_2 \rightarrow A_1 A_1 / b$$

$$A_2 \rightarrow \overline{A_2} \overline{A_2} \overline{A_1} / a \overline{A_1} / b \quad \left. \begin{array}{l} \text{not in} \\ \text{GNF} \end{array} \right]$$

$i^o = 2, j^o = 2 \rightarrow$  Apply Lemma (2).

(iv)  $A_2 \rightarrow aA_1/b$  and  $aA_1/b$  both are strings of length 3 in G.

$A_2 \rightarrow aA_1 z_2/bz_2$

$z_2 \rightarrow A_2 A_1 \quad \{ \alpha \text{ terms } \}$

$z_2 \rightarrow A_2 A_1 z_2$

$\boxed{A_2 \rightarrow aA_1/b/aA_1 z_2/bz_2} \text{ GNF}$

(v)  $z_2 \rightarrow A_2 A_1$

$z_2 \rightarrow \boxed{aA_1 A_1/bA_1/aA_1 z_2 A_1/bz_2 A_1} \text{ GNF}$

positions of  $b$  &  $d$  go additional strings. G.

(vi)  $z_2 \rightarrow A_2 A_1 z_2$  &  $aA_1 z_2 A_1 z_2$

$\boxed{z_2 \rightarrow aA_1 A_1 z_2/bA_1 z_2/aA_1 z_2 A_1 z_2/bz_2 A_1 z_2} \text{ GNF}$

(vii)  $A_1 \rightarrow aA_2 A_2/a$

$\boxed{A_1 \rightarrow aA_1 A_2/bA_2/aA_1 z_2 A_2/bz_2 A_2/a} \text{ GNF}$

get additional strings

$aA_1 A_2/bA_2/a \left\{ \begin{array}{l} aA_1 A_2 \\ bA_2 \end{array} \right. \left. \begin{array}{l} aA_1 A_2 \\ bA_2 \end{array} \right\} \in A \quad \{ aA_1 A_2 \in A \}$

$aA_1 A_2/bA_2/a \left\{ \begin{array}{l} aA_1 A_2 \\ bA_2 \end{array} \right. \left. \begin{array}{l} aA_1 A_2 \\ bA_2 \end{array} \right\} \in A \quad \{ aA_1 A_2 \in A \}$

GNF depends on two lemmas - Substitution Rule and Elimination of Left Recursion

Lemma ① : Substitution Rule

Let  $G = (V, T, P, S)$  be a CFG. Then  $A \rightarrow a$  is a production in  $G$ . Where  $B \rightarrow B_1 | B_2 | \dots$

is a production in  $G$ . Then we can obtain  $G_1 = (V, T, P_1, S)$

where  $P_1$  consists of

$$A \rightarrow aAB_1 | aB_2 | \dots | aB_n$$

Lemma ② : Elimination of Left Recursion

$G = (V, T, P, S)$  be a CFG, if  $P$  consists of

$$A \rightarrow A\alpha_1 | A\alpha_2 | \dots | A\alpha_m | B_1 | B_2 | \dots | B_n$$

Introduce a new variable say  $X$ ,

Let  $G_1 = (V_1, T, P_1, S)$  where  $V_1 = V \cup \{X\}$

$P_1$  can be formed by replacing the

$A \rightarrow$  production by

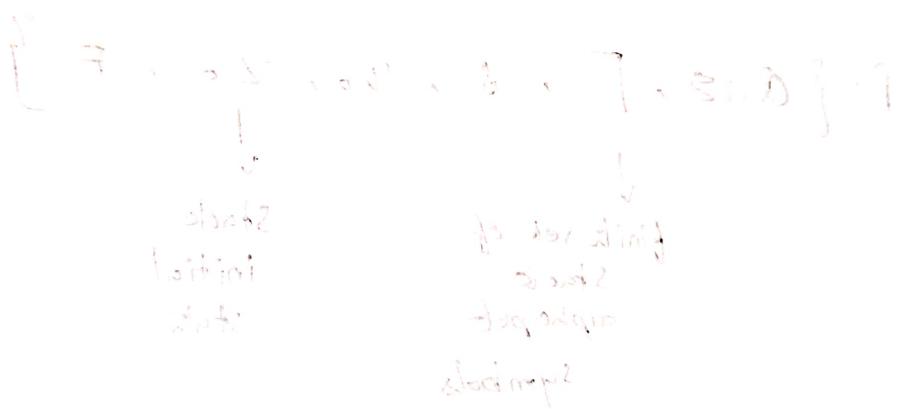
$$\left. \begin{array}{l} (i) A \rightarrow B_i \\ A \rightarrow B_i X \\ (i \leq i \leq n) \end{array} \right\} \begin{array}{l} A \rightarrow B_1 | B_2 | \dots | B_n \\ A \rightarrow B_1 X | B_2 X | \dots | B_n X \end{array}$$

$$\text{iii) } \begin{cases} x \rightarrow \alpha_i \\ x \rightarrow \alpha_i x \\ (i < i \leq m) \end{cases} \quad \left\{ \begin{array}{l} \text{1. } x \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_m \\ \text{2. } x \rightarrow \alpha_1 x \end{array} \right.$$

Weg von einem Wort zu einem anderen  
 Eine Regel kann es möglich machen, dass  
 ein Wort in verschiedene Wörter überführt wird

Erreichbarkeit

Ein Wort ist erreichbar, wenn es durch eine Kette von Regeln aus einem anderen Wort hergestellt werden kann.



$$T_1: S \rightarrow a \quad T_2: S \rightarrow b \quad T_3: S \rightarrow c \quad T_4: S \rightarrow d$$

$$T_5: S \rightarrow e \quad T_6: S \rightarrow f$$

$$T_7: S \rightarrow g \quad T_8: S \rightarrow h$$

$$T_9: S \rightarrow i \quad T_{10}: S \rightarrow j$$

$$T_{11}: S \rightarrow k \quad T_{12}: S \rightarrow l$$

$$T_{13}: S \rightarrow m \quad T_{14}: S \rightarrow n$$

$$T_{15}: S \rightarrow o \quad T_{16}: S \rightarrow p$$

$$T_{17}: S \rightarrow q \quad T_{18}: S \rightarrow r$$

$$T_{19}: S \rightarrow s \quad T_{20}: S \rightarrow t$$

$$T_{21}: S \rightarrow u \quad T_{22}: S \rightarrow v$$

$$T_{23}: S \rightarrow w \quad T_{24}: S \rightarrow x$$

$$T_{25}: S \rightarrow y \quad T_{26}: S \rightarrow z$$

$$T_{27}: S \rightarrow \epsilon$$

Die Erreichbarkeit eines Wortes kann man unterscheiden

Erreichbarkeit (oder nicht erreichbar)

Erreichbarkeit mit endlichem Wortlängenraum (oder nicht erreichbar)

Erreichbarkeit aus endlichem Punkt (oder nicht erreichbar)

Erreichbarkeit mit endlichem Punkt und endlichem Wortlängenraum (oder nicht erreichbar)