PR 6 TBFO

AHMAD NADIL
13521024
K3

Ex 5.1.1.

a.) $G = \{(S), \{0,1\}, S, P\}$

dengan P adalah production rule ;

$S \rightarrow 0S1 \mid 01$

b.) Set $\{ a^i b^j c^k \mid i \neq j \text{ or } j \neq k \}$

$G = \{(S,A,B,C,D,E), \{a,b,c\}, S, P\}$

dgn P adalah :

$S \rightarrow AB \mid CD$
$A \rightarrow aA \mid \varepsilon$
$B \rightarrow bBc \mid E \mid cD$
$C \rightarrow aCb \mid E \mid aA$
$D \rightarrow cD \mid c$
$E \rightarrow bE \mid b$

c.) Set of all strings of a's and b's that are not of the form ww, that is, not equal to any string repeated.

$G = \{(S,T,A,B), (a,b), S, P\}$

P :

$S \rightarrow AB \mid BA \mid T$
$T \rightarrow aTb \mid bTa \mid aTa \mid bTb \mid a \mid b$
$A \rightarrow aAb \mid bAa \mid aAa \mid bAb \mid a$
$B \rightarrow aBb \mid bBa \mid aBa \mid bBb \mid b$

d.) Set of all strings with twice as many 0's as 1's

$G = \{(S), (0,1), S, P\}$

P :

$S \rightarrow \varepsilon \mid SS \mid 00S1 \mid 1S00 \mid 01S0$

Ex 5.1.2

following grammar generates the language of regular expression $0^*1(0+1)^*$

$$S \to A1B$$
$$A \to 0A \mid \varepsilon$$
$$B \to 0B \mid 1B \mid \varepsilon$$

Give leftmost and rightmost derivations :

a) 00101

LM : $S \to A1B \to 0A1B \to 00A1B \to 001B \to 0010B$
$\to 00101B \to 00101$

RM : $S \to A1B \to 0A10B \to 0A101B \to 0A101 \to 00A101 \to 00101$

b.) 1001

LM : $S \to A1B \to 1B \to 10B \to 100B \to 1001B \to 1001$

RM : $S \to A1B \to A10B \to A100B \to A1001B \to A1001 \to 1001$

c.) 00011

LM : $S \to A1B \to 0A1B \to 00A1B \to 000A1B \to 0001B$
$\to 00011B \to 00011$

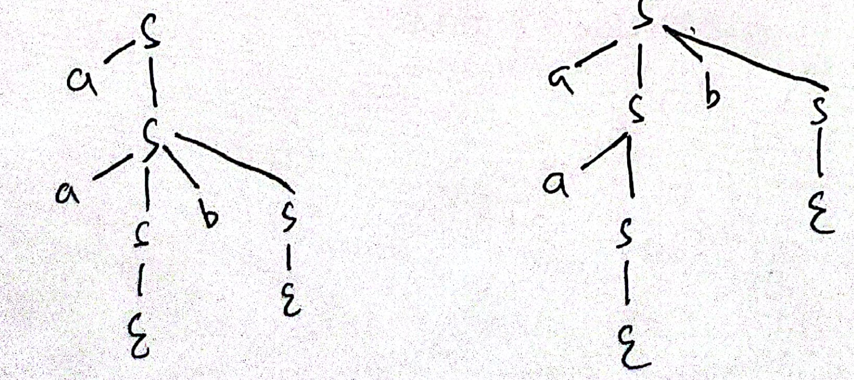
RM : $S \to A1B \to A11B \to A11 \to 0A11 \to 00A11$
$\to 000A11 \to 00011$

Ex 5.4.1

$$S \to aS \mid aSbS \mid \epsilon$$

This grammar is ambiguous. Show in particular that the string aab has two :

a.) Parse trees



Leftmost derivations

b.) #1 $S \to aS \to aaSbS \to aab$

\# $S \to aSbS \to aaSbS \to aabS \to aab$

c.) Rightmost derivations

\#1 $S \to aS \to aaSbS \to aaSb \to aab$

\#2 $S \to aSbS \to aSb \to aaSb \to aab$

Ex 5.4.7

The following grammar generates prefix expressions with operands x and y and binary operators +, −, and *

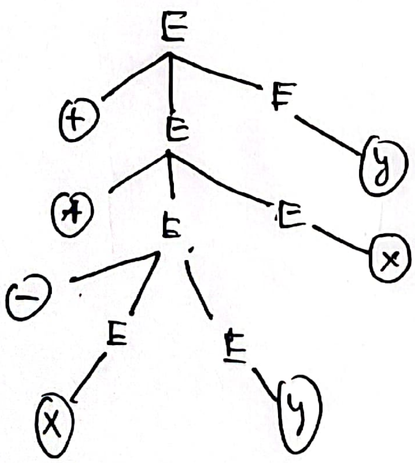
$$E \to +EE \mid *EE \mid -EE \mid x \mid y$$

a.) find LM RM derivations, and a derivation tree for the string $+*-xyxy$

LM : $E \to +EE \to +*EEE \to +*-EEEE \to +*-xEEE$
$\to +*-xyEE \to +*-xyxE \to +*-xyxy$

RM : $E \to +EE \to +Ey \to +*EEy \to +*Exy \to +*-EExy$
$\to +*-Eyxy \to +*-xyxy$

Derivation tree :



b.) Pada grammar ini, setiap production rule memiliki awalan yg berbeda (+, *, −, x, y). Untuk string yg termasuk dalam CFL ini, apabila diambil LM var. E, langkah berikutnya bergantung dengan simbol yg ingin dimunculkan dari w. Karena awalan tiap rule berbeda, maka dijamin hanya terdapat satu rule yg dapat dipakai pada saat itu. Maka dijamin hanya ada satu LM derivation untuk w dan grammar terbukti tidak ambigu.