Name: Avinash Shrivastava
Roll no: 22512

TOC Assignment 5: Context free Grammar

4.) Consider the CFG

$S \to SS \mid XaXaX \mid \Lambda$
$X \to bX \mid \Lambda$

(i) Prove that X can generate any b^*.

Since, $X \to \Lambda$,
$\Rightarrow$ we can terminate X with Λ
and second production,
$X \to bX$
can be used to generate b^n by applying it n times and terminating it with first production.

Hence, b^* can be generated from X.

(ii) To prove: XaXaX can generate $b^*ab^*ab^*$.

From proof of (i), we know that
X can generate b^*
ie, $X \to b^*$
By using the above production,
Let $S' \to \underline{X}aXaX$
$\to b^*a\underline{X}aX$ $[X \to b^*]$
$\to b^*ab^*a\underline{X}$ $[X \to b^*]$
$\to b^*ab^*ab^*$ $[X \to b^*]$

Hence, XaXaX can generate $b^*ab^*ab^*$. Proved.
ie, $XaXaX \to b^*ab^*ab^*$

(iii) To Prove $S \to (b^*ab^*ab^*)^*$

~~Since, $S \to xaxax$ [Given]~~

$S \to \Lambda$ [Given]

Given, $S \to SS$ [Given]

$S \to xaxax\,S$ [$S \to xaxax$]

$S \to b^*ab^*ab^*S$ [$xaxax \to b^*ab^*ab^*$ (from proof (ii))]

$S \to b^*ab^*ab^*SS$ [$S \to SS$]

$S \to b^*ab^*ab^*(xaxax)S$ [$S \to xaxax$]

$S \to (b^*ab^*ab^*)(b^*ab^*ab^*)S$ [$xaxax \to b^*ab^*ab^*$]

⋮

and so on.

We can keep on using the above productions and generate $(b^*ab^*ab^*)^*$

Hence, proved.

(iv) To Prove:

The language of this CFG is set of all words in $(a+b)^*$ with an even no. of a's with exception that Λ to have even no. of a's and of the words with no b's only Λ can be generated.

Proof:

From proof (iii), we know that

$S \to (b^*ab^*ab^*)^*$

Every words in this language is generated by repetition of the factor $(b^*ab^*ab^*)$ which every introduces even no. of a's i.e, (exactly 2 a's ~~every~~ time) with arbitrary no. of b's.

5.) Given CFG

$S \rightarrow XbaaX \mid aX$

$X \rightarrow Xa \mid Xb \mid \Lambda$

Since, $X \rightarrow Xa$; $X \rightarrow Xb$ & $X \rightarrow \Lambda$

$\Rightarrow X \rightarrow (a+b)^*$

The production

$S \rightarrow \underline{X}baa\underline{X}$ generates

$S \rightarrow (a+b)^* baa (a+b)^*$ $[\because X \rightarrow (a+b)^*]$

$\Rightarrow$ S generates all words that contains 'baa'.

Also, $S \rightarrow a\underline{X}$

$\Rightarrow S \rightarrow a(a+b)^*$ $[\because X \rightarrow (a+b)^*]$

clearly, S generates all words that begins with a.

Hence, the language generated by this CFG is the language of all words that either begins with a or contain substring baa or both.

Now lets generate abaa $\in$ language in two different ways.

1. $S \rightarrow \underline{X}baaX$

$S \rightarrow \cancel{X}abaa\underline{X}$ $[\because X \rightarrow Xa]$

$S \rightarrow abaa$ $[X \rightarrow \Lambda]$

2. ~~$S \rightarrow Xa$~~

~~$S \rightarrow Xaa$~~ ~~$[\because X \rightarrow Xa]$~~

~~$S \rightarrow Xbaa$~~ ~~$[X \rightarrow Xb]$~~

~~$S \rightarrow Xabaa$~~ ~~$[X \rightarrow Xa]$~~

~~$S \rightarrow abaa$~~ ~~$[X \rightarrow \Lambda]$~~

$S \rightarrow a\underline{X}$

$S \rightarrow a\underline{X}a$ $[X \rightarrow Xa]$

$S \rightarrow a\underline{X}aa$ $[X \rightarrow Xa]$

$S \rightarrow aXbaa$ $[X \rightarrow Xb]$

$S \rightarrow abaa$ $[X \rightarrow \Lambda]$

7.) Find CFG for,

(i) ab^*

$S \rightarrow aX$

$X \rightarrow bX$

$X \rightarrow \Lambda$

(ii) a^*b^*

$\cancel{S \rightarrow \Lambda}$

$\cancel{S \rightarrow aSS}$

$\cancel{S \rightarrow a}$

$\cancel{S \rightarrow b}$

$S \rightarrow XY$

$X \rightarrow aX$

$X \rightarrow \Lambda$

$Y \rightarrow bY$

$Y \rightarrow \Lambda$.

(iii) $(baa + abb)^*$

$S \rightarrow SS$

$S \rightarrow baa$

$S \rightarrow abb$

$S \rightarrow \Lambda$

8. Find CFGs for the following languages over the alphabet $\Sigma = \{a, b\}$

(i) All words in which letter b is never tripled.

$S \rightarrow aS \mid bX \mid \Lambda$
$X \rightarrow aS \mid bY \mid \Lambda$
$Y \rightarrow aS \mid \Lambda$

(ii) All words that have exactly two or three b's.

$S \rightarrow aS \mid bY$
$X \rightarrow aX \mid bY$
$Y \rightarrow aY \mid bZ \mid \Lambda$
$Z \rightarrow aZ \mid \Lambda$

(iii) All words that do not have substring ab.

$S \rightarrow bS \mid aX \mid \Lambda$
$X \rightarrow aX \mid \Lambda$

(iv) All words that do not have the substring baa.

$S \rightarrow aS \mid bX \mid \Lambda$
$X \rightarrow bX \mid aY \mid \Lambda$
$Y \rightarrow bX \mid \Lambda$

(v) All words that have different first and last letter.

$S \rightarrow aX \mid bY$
$X \rightarrow aX \mid bW$
$Y \rightarrow bY \mid aZ$
$W \rightarrow bW \mid aX \mid \Lambda$
$Z \rightarrow aZ \mid bY \mid \Lambda$

(9) consider the CFG

$S \to AA$

$A \to AAA$

$A \to bA \mid Ab \mid a$

Prove that the language generated by these productions is the set of all words with an even no. of a's but not no a's. Contrast this grammar with CFG in Problem 4.

Proof :

Given the productions from nonterminal A:

1. Each A in the working string must eventually be replaced by exactly one a.
2. The production $A \to bA$ and $A \to Ab$ do not change the no. of A's only adding the terminal b.
3. The production $A \to AAA$ adds two A's at a time, ensuring the parity (odd/even) of A's remain even.
4. Since, A must eventually terminates as a, the final string will contain an even no of a's.
5. The grammar cannot produce a word without any a's as every A must resolve into a.
6. Thus, the language consists of words with an even no. of a's excluding the empty string Λ.

The difference b/w the language defined in Problem 4 and this is that the language defined in problem 4 without Λ.

11 Write a CFG to generate the language MOREA of all strings that have more a's than b's (not necessarily only one more, as with non terminal A for the language EQUAL, but any no. of more a's than b's).

MOREA = {a aa aab aba baa, ... }

Required CFG :-

$S \to SS \mid EXE$
$X \to aX \mid a$
$E \to$ EQUAL.

(10) Describe the language defined by CFG:

$S \to SS$
$S \to XXX$
$X \to aX \mid Xa \mid b$

This language is the language of all words containing a positive no. of b's divisible by 3.

(13) We have,

TRAILING-COUNT = { $sa^{length(s)}$ for all s in $(a+b)^*$ }

is non regular. Show however that it is context free and generated by

$S \to aSa \mid bSa \mid \Lambda$

Proof:

The grammar $S \to aSa \mid bSa \mid \Lambda$ generate the language as

- The nonterminal S on the left of each production becomes the letters of the word s.

- The a's on the right side of S ensures that the count of a's matches the length of s.

Thus, for ~~such~~ each string s, the grammar produces $sa^{|s|}$, showing the language is context-free.

(15.) Below is a set of words and a set of CFGs. For each words, determine whether the word is in the language of each CFG and, if it is, draw a syntax tree to prove it.

(i) ~~word~~ aab

CFGs

CFG 1. $S \rightarrow aSb \mid ab$

CFG 2. $S \rightarrow aS \mid bS \mid a$

CFG 3. $S \rightarrow aS \mid aSb \mid X$

$X \rightarrow aXa \mid a$

CFG 4. $S \rightarrow aAS \mid a$

$A \rightarrow SbA \mid SS \mid ba$

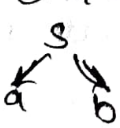
CFG 5. $S \rightarrow aB \mid bA$

$A \rightarrow a \mid aS \mid bAA$

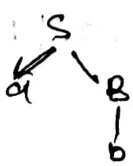
$B \rightarrow b \mid bS \mid aBB$

(i) ab

CFG 1.

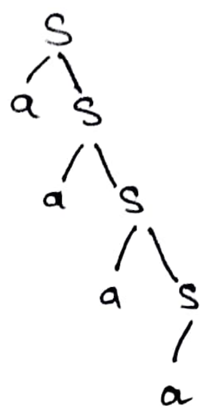


CFG 5

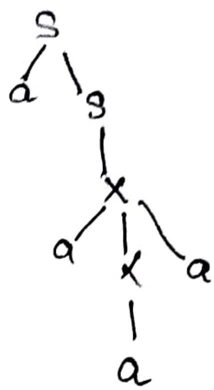


(ii) aaaa

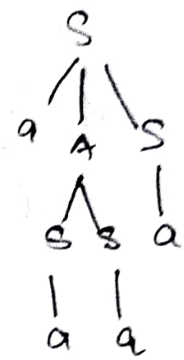
CFG 2



CFG 3



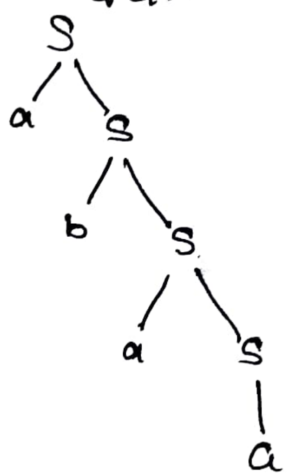
CFG 4



(iv) abaa

~~(iv) abbb~~

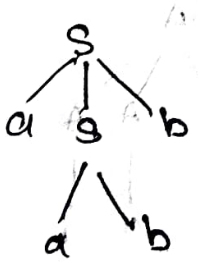
CFG 2.



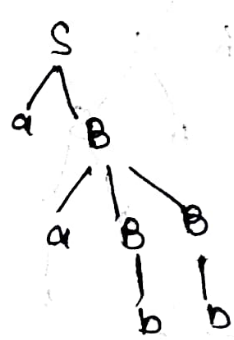
CFG 4

(iii) aabb

CFG 1

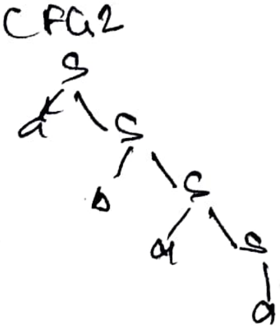


CFG 5.

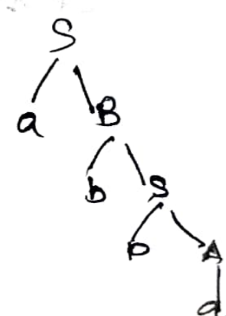


(v) abba

CFG 2

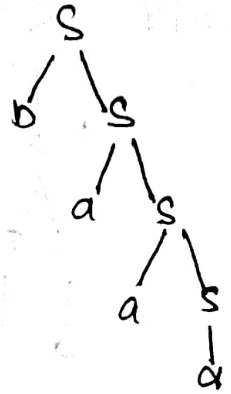


CFG 5



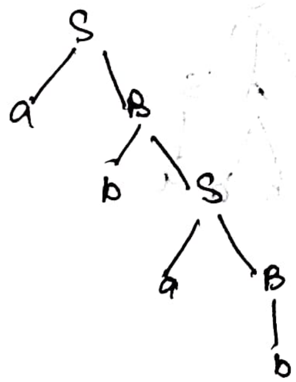
(vi) baaa

CFG 2



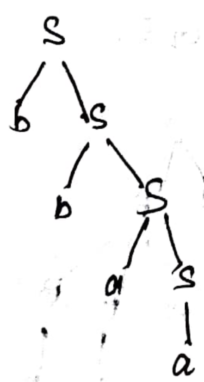
(vii) abab

CFG 5

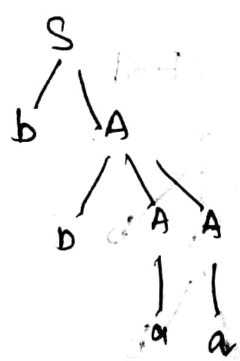


(viii) bbaa

CFG 2

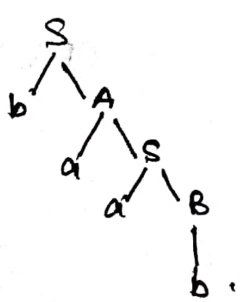


CFG 2



(ix) baab

CFG 5

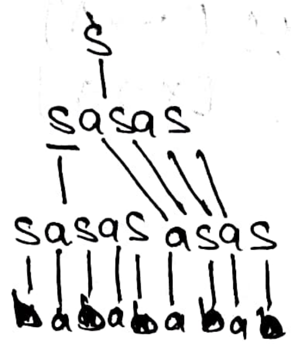


(16.) Show that the following CFGs are ambiguous by finding a word with two distinct syntax trees.

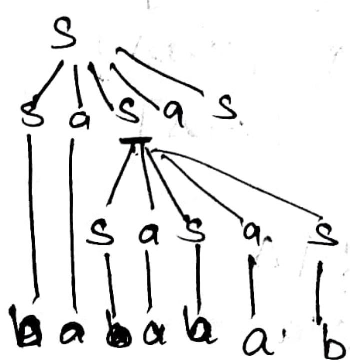
(i) $S \to SaSaS \mid b$

Let the word be babababab

Tree 1

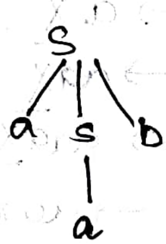


Tree 2

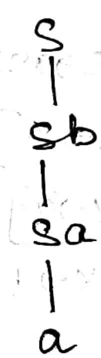


(ii) $S \to$ ~~aSaS|ab~~ $aSb \mid Sb \mid Sa \mid a$

word: ~~abab~~ ~~aababab~~ aab



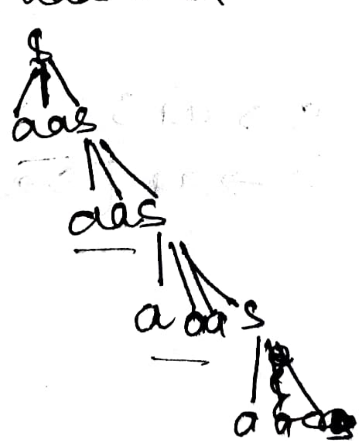
Tree 1



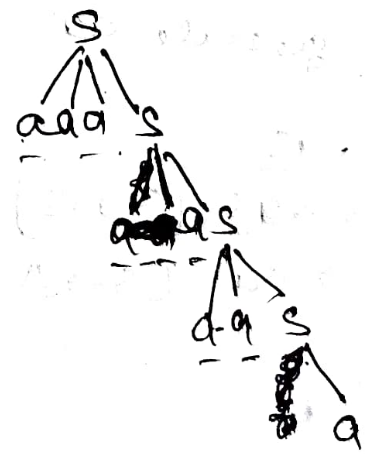
Tree 2

(iii) $S \to aaS \mid aaaS \mid a$

word: aaaaaaa



Tree 1



(14) Show that the language following CFG.s that use Λ are ambiguous.

(i) $S \rightarrow XaX$

$X \rightarrow aX \mid bX \mid \Lambda$

Let's show the ambiguity by forming 'aa' using two ways.

$S \rightarrow XaX$

$S \rightarrow aXaX \; [X \rightarrow aX]$

$S \rightarrow a\Lambda aX \; [X \rightarrow \Lambda]$

$S \rightarrow aa \; [X \rightarrow \Lambda]$

$S \rightarrow XaX$

$S \rightarrow XaaX \; [X \rightarrow aX]$

$S \rightarrow aaX \; (X \rightarrow \Lambda]$

$S \rightarrow aa \; [X \rightarrow \Lambda]$

(ii) $S \rightarrow aSX \mid \Lambda$

$X \rightarrow aX \mid a$.

Let's generate aaaa.

$S \rightarrow aSX$

$\rightarrow aaSXX \; [S \rightarrow aSX]$

$\rightarrow aaXX \; [S \rightarrow \Lambda]$

$\rightarrow aaXa \; [X \rightarrow a]$

$\rightarrow aaaa \; [X \rightarrow a$

$S \rightarrow aSX$

$S \rightarrow aX \; [S \rightarrow \Lambda]$

$S \rightarrow aaX \; [X \rightarrow aX]$

$S \rightarrow aaaX \; [X \rightarrow aX]$

$S \rightarrow aaaa \; [X \rightarrow a]$.

(iii) $S \rightarrow aS \mid bS \mid aaS \mid \Lambda$

Let's generate 'aa'

$S \rightarrow aS$

$S \rightarrow aaS \; [S \rightarrow aS]$

$S \rightarrow aa \; [S \rightarrow \Lambda$.

$S \rightarrow aaS$

$S \rightarrow aa \; [S \rightarrow \Lambda]$

(iv) Find unambiguous CFGs that generates these three languages.

(i) $S \to bS \mid aX$

$X \to aX \mid bX \mid \Lambda$

(ii) $S \to aX$

$X \to aX \mid a$

(iii) $S \to aS \mid bS \mid \Lambda$.

(v) For each of these three languages, find an unambiguous grammar that generates exactly the same language except for the word Λ. Do this by not employing the symbol Λ in the CFGs at all.

(i) $S \to bS \mid aX \mid a$

$X \to aX \mid bX \mid a \mid b$.

(ii) $S \to aX$

$X \to aX \mid a$.

(iii) $S \to aS \mid bS \mid a \mid b$

(18) Begin to draw the total language tree for the following CFGs until we can be sure we have found all the words in these languages with one, two, three or four letters. Which of these CFGs are ambiguous?

(i) $S \to aS \mid bS \mid a$.

S — a

as — aa

aas — aaa

aaaS — aaaa

aabs — aaba

abs — aba

abas — abaa

abbs — abba

bs — ba

bas — baa.

~~babs — baba~~

baaS — baaa

babs — baba.

bbs — bba

bbas — bbaa

bbbS — bbba.

clearly, it is not ambiguous.

(ii) S → asas | b

S → b

asas — abas — abab

abaasas

asab — abab

aasasab

aasasas — abbases

aaasasasas

aasasaasas. aasasas — aasaasasab

Clearly, it is ambiguous.

(iii) $S \to aSa \mid bSb \mid a$

S — a
- aSa — aaa
  - aaSaa — aaaaa
  - abSba — ababa.
- bSb — bab
  - baSab — baaab
  - bbSbb — bbabb

Clearly, its not ambiguous.

(iv) $S \to aSb \mid bX$

$X \to bX \mid b$

S — aSb — aaSbb — aaaSbbb
- aabXbb
- abXb — abbb
  - abbXb
- bX — bb
  - bbX — bbb
    - bbbX — bbbb

Not ambiguous.

(V) $S \to bA \mid aB$

$A \to bAA \mid aS \mid a$

$B \to aBB \mid bS \mid b$.

- S → bA — ba
  - baS — babA — baba
    - baaB — baab
  - bbAA — bbaA — abaa.
    - bbAa
    - bbbAAA
    - bbAbAA
    - bbaSA
    - bbAaS
- aB — ab
  - abS — abaB — abab
    - abbA — abba
  - aaBB. — aabB — aabb
    - aabB — aabb
    - aaaBBB
    - aaBaBB
    - aabSB
    - aaBbS

clearly, this CFG is ambiguous.

(19) – Not part of syllabus.