Lecture 7

Context Free Grammar   
and Ambiguity

**Context Free Grammar**

Consider the Grammar G4,

G4 = ( {S}, {a , b}, R, S )

The set of rules R is,

S aSb | SS |

This grammar generates the strings abab, aaabbb, aababb

Derivation for abab

S SS

aSbaSb

abab

abab

Derivation for aaabbb

S aSb

aaSbb

aaaSbb

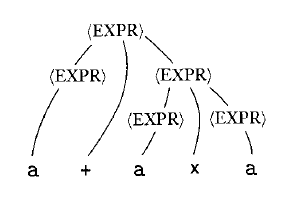
aaabbb

Derivation for aababb

S aSb

aSSb

aaSbaSbb

 aababb

Another Example

G = (N, , P, S) where

N = {S, A}

= {a,b,c}

S aSc

S aAc

A b

Write the language generated by this grammar

Typical Derivation

S aSc

aaAcc

aabcc

S aSc

aaScc

aaaAccc

aaabccc

So, the language generated by this is

{an b cn where n 0}

**Ambiguity**

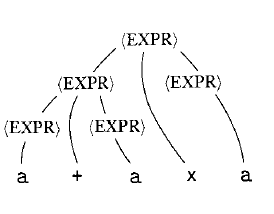
* A string is derived ambiguously in context free grammar G if it has two or more different leftmost derivations.
* Grammar G is ambiguous if it generates some string ambiguously.
* Example:

+

x

()

a

* The grammar generates the string a+axa ambiguously
* Now, see the two different parse tree for that string
* If Grammar G = (V,,R,

) where .

V is {,, }

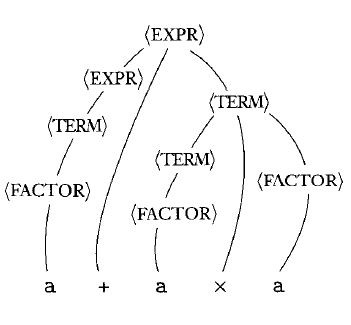
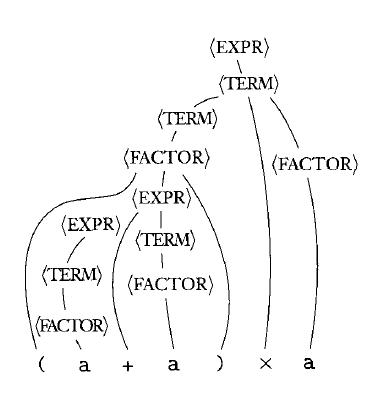
And is { a, +, x, ( , ) }

* The rules are

+ |

x |

() | a

* So, for string a+axa the parse tree will be
* Draw the parse tree for the following string (a+a) x a

**Chomsky Normal Form (CNF)**

A CFG is in CNF if every rule has one of the following form.

A BC B S and C S

A a a

S S is the start variable

**Conversion of CFG to CNF**

CFG to CNF:

Adding a new start variable

Eliminating null production

A

Eliminating Unit Production

A

Converting remaining rules to proper form