Context Free Grammars

Shakir Ullah Shah

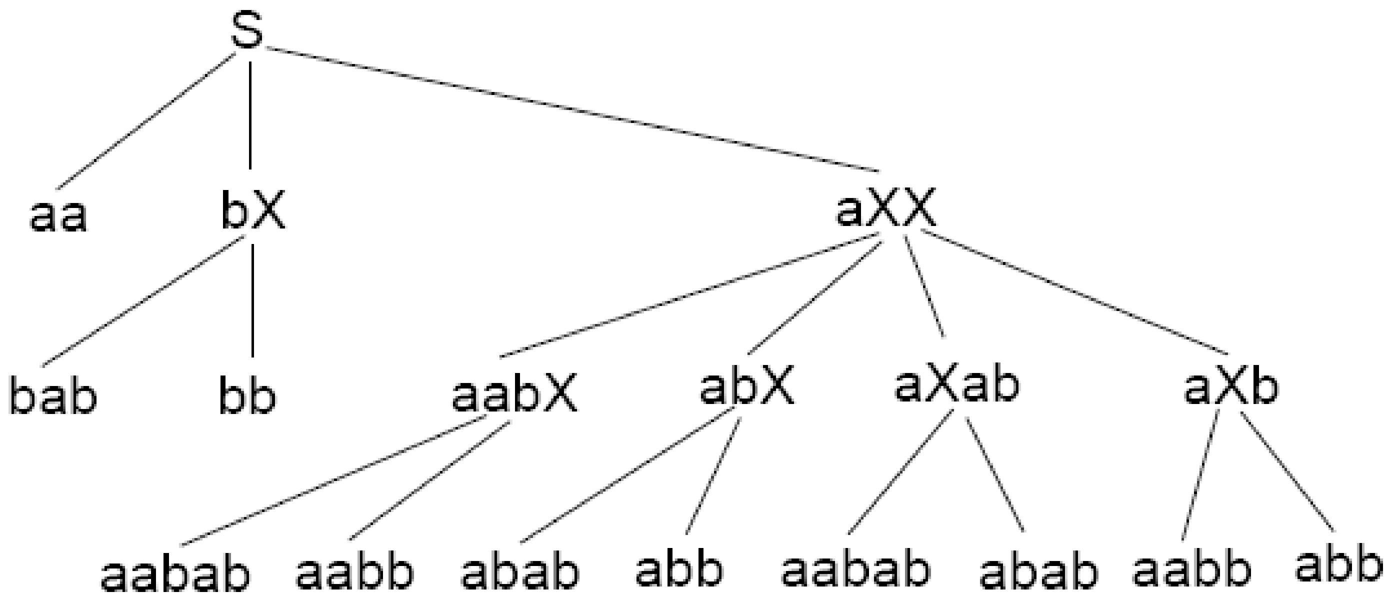
Total language tree

- For a given CFG, a tree with the start symbol S as its root and whose nodes are working strings of terminals and non-terminals.

- The descendants of each node are all possible results of applying every production to the working string. This tree is called **total language tree**. Following is an example of total language tree

Example

- Consider the following CFG
- S □ aa|bX|aXX
- X □ ab|b, then the total language tree for the given CFG may be



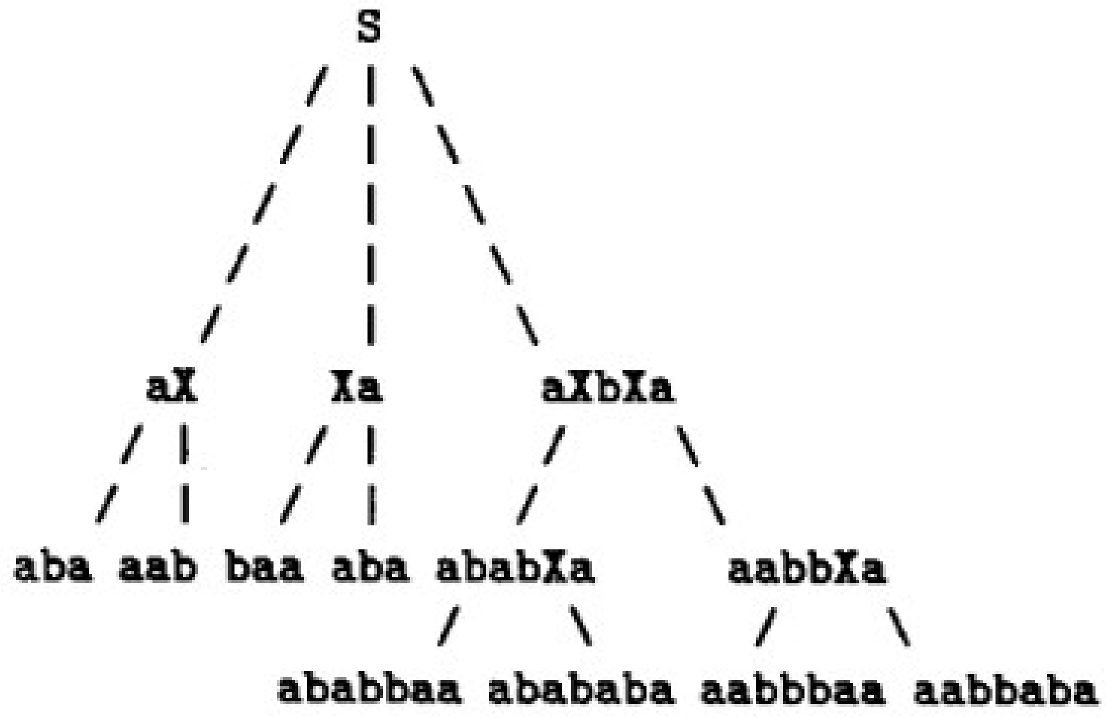
Example continued …

- It may be observed from the previous total language tree that dropping the repeated words, the language generated by the given CFG is
- {aa, bab, bb, aabab, aabb, abab, abb}

Example 2

$$S \rightarrow aX \mid Xa \mid aXbXa$$
$$X \rightarrow ba \mid ab$$

This CFG has total language tree as follows:



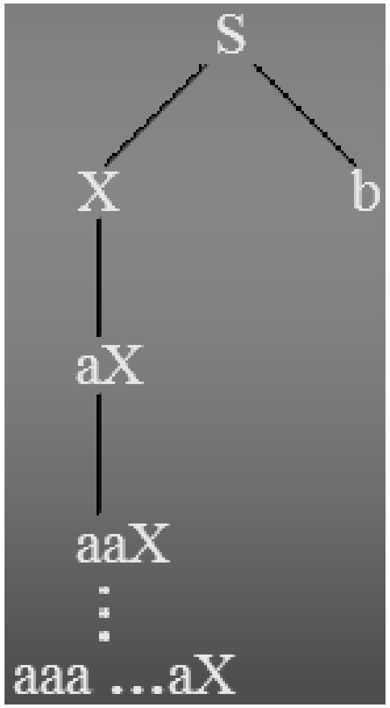
The CFL is finite.

Example

- Consider the following CFG
- $S \rightarrow X|b$, $X \rightarrow aX$
- then following will be the total language tree of the above CFG

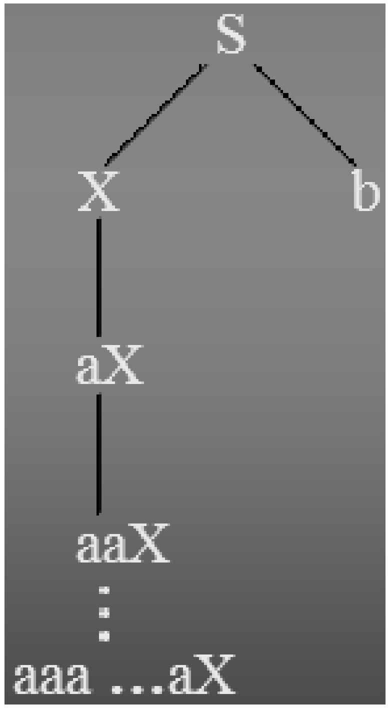
Example

- Consider the following CFG
- S X|b , X aX
- then following will be the total language tree of the above CFG



Example

- Consider the following CFG
- S X|b , X aX
- then following will be the total language tree of the above CFG



- Note: It is to be noted
- that the only word in
- this language is **b**.

Semi Word

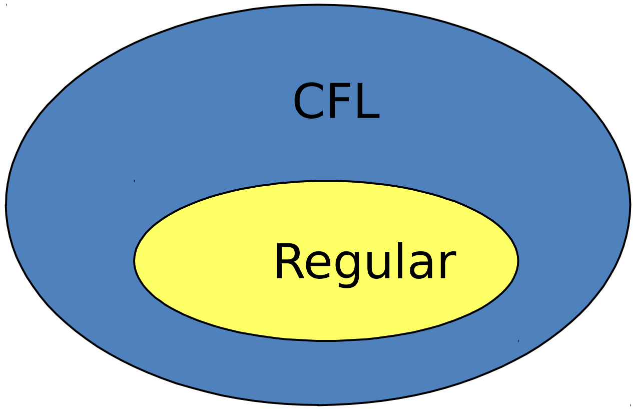
- For a given CFG, semiword is a string of terminals (may be none) concatenated with with exactly one non-terminal (on the right).
- In general semiword has the shape
- **(terminal) (terminal)….(terminal) (Non-Terminal)**
- e.g. aaaX   abcY   bbY

A word is a string of terminals only (zero or more terminals)
Λ is also a word.

Regular Grammar

Given an FA, there is a CFG that generates exactly the language accepted by the FA.

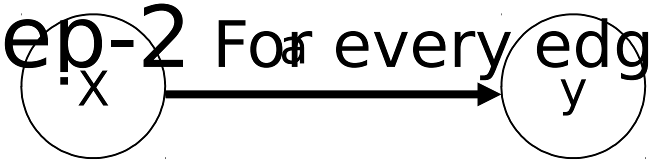
- In other words, all regular languages are CFLs

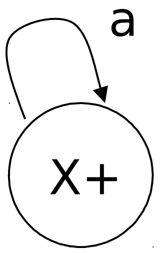


Creating a CFG from an FA

Step-1 The Non-terminals in CFG will be all names of the states in the FA with the start state renamed S.

Step-2 For every edge



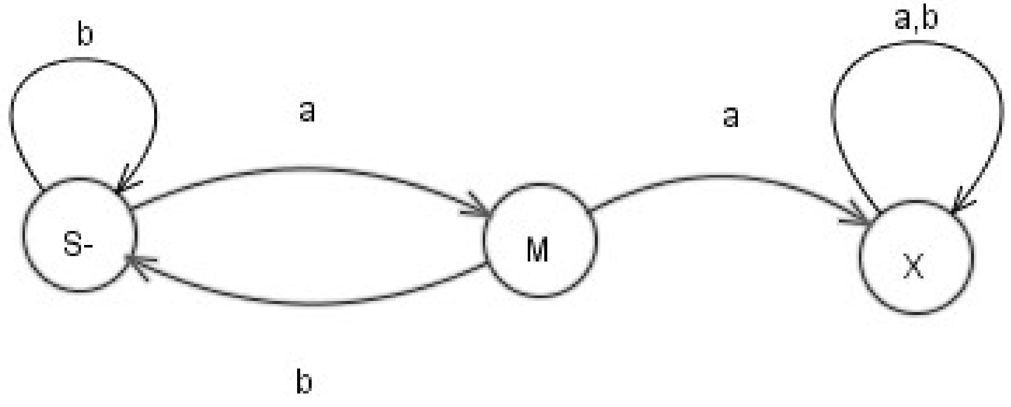


Create productions X aY or X aX
Do the same for b-edges

Step-3 For every final-state X, create the production

X Λ

Example



$S \to aM$
$S \to bS$
$M \to aX$
$M \to bS$
$X \to aX$
$X \to bX$
$X \to \Lambda$

Note: It is not necessary that each C
has a corresponding FA. But each FA
has an equivalent CFG.

Theorem

- If every production in a CFG is one of the following forms
  1. Nonterminal → semiword
  2. Nonterminal → word

  then the language generated by that CFG is **regular**.

Regular grammar

- **Definition**:

  A CFG is said to be a **regular grammar** if it generates the regular language *i.e.* a CFG is said to be a **regular grammar** in which each production is one of the two forms

  Nonterminal □ semiword

  Nonterminal □ word

Examples

1. The CFG S aaS | bbS | Λ   is a regular grammar. It may be observed that the above CFG generates the language of strings expressed by the RE (aa+bb)*.
2. The CFG S aA|bB , A aS|a , B bS|b is a regular grammar. It may be observed that the above CFG generates the language of strings expressed by RE $(aa+bb)^{+}$.

– Following is a method of building TG corresponding to the regular grammar.

TG for Regular Grammar

- For every regular grammar there exists a TG corresponding to the regular grammar. Following is the method to build a TG from the given regular grammar.

1. Define the states, of the required TG, equal in number to that of nonterminals of the given regular grammar. An additional state is also defined to be the final state. The initial state should correspond to the nonterminal S.
2. For every production of the given regular grammar, there are two possibilities for the transitions of the required TG as follows

Method continued …

(i)
If the production is of the form nonterminal → semiword, then transition of the required TG would start from the state corresponding to the nonterminal on the left side of the production and would end in the state corresponding to the nonterminal on the right side of the production, labeled by string of terminals in semiword.

Method continued …

(ii)

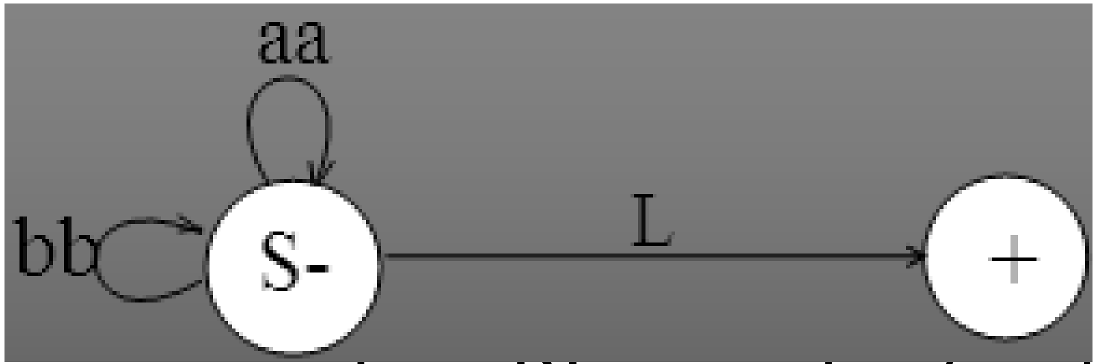
If the production is of the form nonterminal word, then transition of the TG would start from the state corresponding to nonterminal on the left side of the production and would end on the final state of the TG, labeled by the word. Following is an example in this regard

Example

- Consider the following CFG

  S aaS | bbS | Λ

  The TG accepting the language generated by the above CFG is given below



- The corresponding RE may be (aa+bb)*.

Example

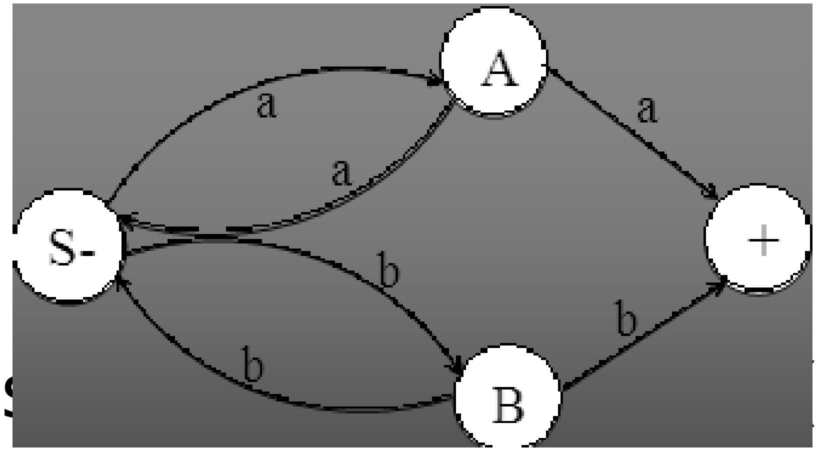
- Consider the following CFG

  S □ aA|bB

  A □ aS|a

  B □ bS|b

  then the corresponding TG will be



The corres (aa+bb)+

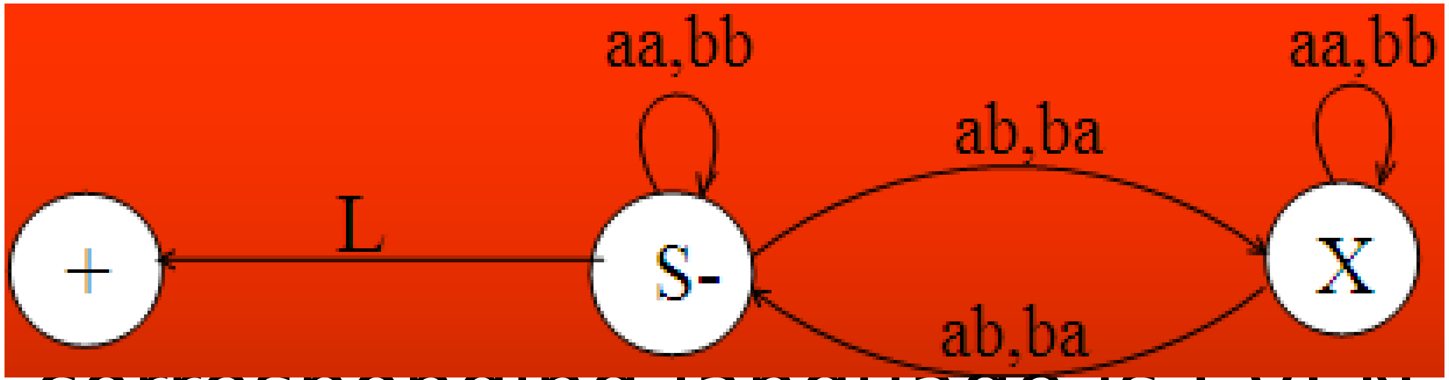
Example

- Consider the following CFG

S → aaS | bbS | abX | baX | Λ

X → aaX | bbX | abS | baS

then the corresponding TG will be



The corresponding language is EVEN-EVEN

FA Conversion to Regular grammar

- Accepts the language of all words with a double a:

  S →aM

  S →bS

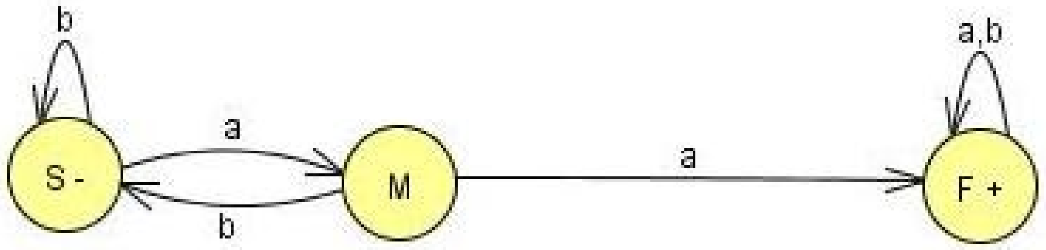
  M →aF

  M →bS

  F →aF

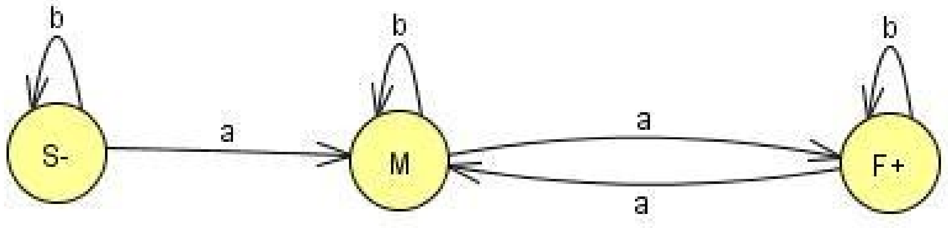
  F →bF

  F →Ʌ



FA Conversion to Regular grammar

- Example
- The language of all words with an even number of a’s is accepted by the FA:



S → aM I bS
M → bM | aF
F → aF I bF | λ

Remarks

- We have seen that some regular languages can be generated by CFGs, and some non-regular languages can also be generated by CFGs.
- ALL regular languages can be generated by CFGs.
- There is some non-regular language that cannot be generated by any CFG.
- Thus, the set of languages generated by CFGs is properly **larger** than the set of regular languages, but properly **smaller** than the set of all possible languages.