Lecture 12

Context Free Grammar :

$L = \{0^n 1^n \mid n \geq 0\}$

→ To implement this in a machine, the machine must be able to count/keep track of how many 0s & 1s are in the machine.

But FSM can not count or keep track of arbitrary amounts of information. Hence the language can not be represented by FSM.

So its not a Regular Language.

[The languages that are accepted by any FSM are regular languages]

Grammar:

set of production rules to generate strings of a language.

Example of grammars:

- Grammar for string "01"

$$\left.\begin{aligned} S &\rightarrow 0A \\ A &\rightarrow 1B \\ B &\rightarrow \epsilon \end{aligned}\right\} \text{production rules}$$

start symbol — S
Variables / non-terminals — A, B
terminals — 0, 1, ϵ

$S \rightarrow 0\underline{A} \longrightarrow 01\underline{B} \longrightarrow 01\epsilon \longrightarrow 01$

$S \xrightarrow{*} 01$ [S derives "01"]

Regular Grammar

↳ generates strings of regular language using production rules described in the grammar.

↳ A grammar is regular if in the production rule the variables are always on the left or always on the right.

Right Regular Grammar:

$$S \to A$$
$$A \to 1A \mid 0A \mid 010B$$
$$B \to 1B \mid 0B \mid \epsilon$$

Left Regular Grammar:

$$S \to A0$$
$$A \to A0 \mid A1 \mid \epsilon$$

A regular grammar can not keep count.

Context Free Grammar:

↳ in production rule, variable can be anywhere.

- $L = \{0^n 1^n \mid n \geq 0\}$

  $S \rightarrow 0S1 \mid \epsilon$

Definition

A *context-free grammar* is a 4-tuple (V, Σ, R, S), where

1. V is a finite set called the ***variables***,
2. Σ is a finite set, disjoint from V, called the ***terminals***,
3. R is a finite set of ***rules***, with each rule being a variable and a string of variables and terminals, and
4. $S \in V$ is the start variable.

Note:

- *R* is a set of productions or substitution rules of the form-

  $$A \rightarrow \alpha$$

  where A is a symbol in *V* and a is a string over $V \cup T$

- Usually Capital letter used for non-terminals and Small letters for terminals.

Example 1/

$S \to 0B \mid 1A$

$A \to 0 \mid 0S \mid 1AA \mid \epsilon$

$B \to 1 \mid 1S \mid 0BB$

generate 00110101

Derivation:

Following Right most derivation. Left most derivation can also be followed.

$S \to 0B$

$\to 00BB$

$\to 00B1S$

$\to 00B10B$

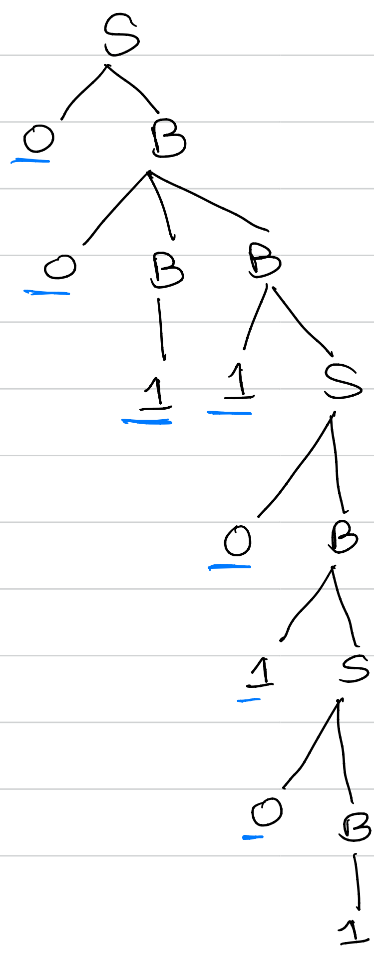
$\to 00B101S$

$\to 00B1010B$

$\to 00B10101$

$\to 00110101$

Parse tree:



Example 2)

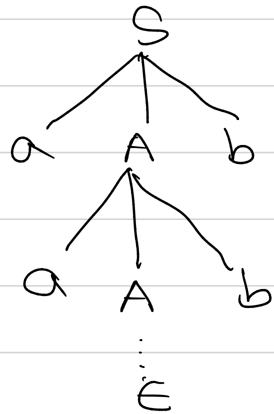
$S \rightarrow aAb$

$A \rightarrow aAb \mid \epsilon$

does it generate aabbb ?

$S \rightarrow aAb$

$\rightarrow aaAbb$



The grammar will always generate strings with equal number of 0s & 1s.

Hence it doesn't generate aabbb

Example 3)

$A \rightarrow 0A1 \mid B$
$B \rightarrow \#$

variables: A, B
terminals: 0, 1, #
start variable: A

- Is the string 00#11 in L? → Yes
- How about 00#111, 00#0#1#11? → No
- The language of this CFG? → Yes.

$L = \{0^n\#1^n : n \geq 0\}$

Practice/

Write the grammar for the following languages-

1. Numbers without leading zeros, e.g., 109, 0 but not 019
2. The language $L = \{a^n b^n c^m d^m \mid n \geq 0, m \geq 0\}$
3. The language $L = \{a^n b^m c^m d^n \mid n \geq 0, m \geq 0\}$
4. Linear equations over x, y, z, like:
   - $x + 5y - z = 9$
   - $11x - y = 2$

1)

$S \rightarrow AB \mid 0$

$A \rightarrow 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$B \rightarrow BD \mid D$

$D \rightarrow 0 \mid A$

2)

$S \rightarrow AB$

$A \rightarrow aAb \mid \epsilon$

$B \rightarrow cBd \mid \epsilon$

3)

$S_1 \rightarrow aS_1d \mid aS_2d \mid S_2$

$S_2 \rightarrow bS_2c \mid \epsilon$

4) Linear equations over x, y, z:

```
Equation -> Expression "=" Number
Expression -> Term | Term "+" Expression | Term "-" Expression
Term -> Coefficient Variable | Coefficient
Coefficient -> Number | Number "x" | Number "y" | Number "z"
Variable -> "x" | "y" | "z"
```

Explanation:

"Equation" is the start symbol, representing the linear equation.
"Expression" represents the left-hand side of the equation, which can be a single "Term" or have multiple "Term"s connected by "+" or "-".
"Term" represents a coefficient with an optional "Variable" (x, y, or z).
"Coefficient" represents a number or a number followed by one of the variables "x", "y", or "z".
"Variable" represents one of the variables x, y, or z.

Every Regular language is Context Free.

Every CFG can not be converted to Regular Expression.

Regular language is a subset of Context free language.

Design CFG/

- $a^n \mid n \geq 0$ — $S \rightarrow aS \mid \epsilon$

- $(a+b)^*$ — $S \rightarrow aS \mid bS \mid \epsilon$

- $(a+b)^+$ — $S \rightarrow aS \mid bS \mid a \mid b$

- $(a+b)^3$ — $S \rightarrow AAA$
  $A \rightarrow a \mid b$

- $(a+b)^2$ strings of length 2 — $S \rightarrow AA$
  $A \rightarrow a \mid b$

- strings of length at least 2 — $S \rightarrow AAB$
  $A \rightarrow a \mid b$
  $B \rightarrow aB \mid bB \mid \epsilon$

- strings of length at most 2 — $S \rightarrow \epsilon \mid A \mid AA$
  $A \rightarrow a \mid b$

- strings starting with a & ending with b

$S \longrightarrow aAb$
$A \longrightarrow aA \mid bA \mid \epsilon$

- strings of even length

$S \rightarrow \epsilon \mid ASA$
$A \longrightarrow a \mid b$

- strings of odd length

$S \rightarrow aA \mid bA$
$A \rightarrow aS \mid bS \mid \epsilon$

- $L = \{ a^n b^n c^m \mid m, n \geq 1 \}$

$S \longrightarrow AB$

$A \longrightarrow aAb \mid ab$

$B \longrightarrow cB \mid c$