Module: 3

## Regular Grammar

- Mathematical model for FA.

* Grammer: It has 4 tuples

finite Automata

- V: Non-terminal (Represented by capital letters)
- t: terminal → Represented by small letters → Stop something
- P: Production rule
- S: Start symbol. If there is more than 1 production rule, S is the 1st starting pdn rule.

Eg: $A \rightarrow aA \mid \epsilon$

- LHS contains only terminal.
- RHS contains combinations of terminal & non-terminal.

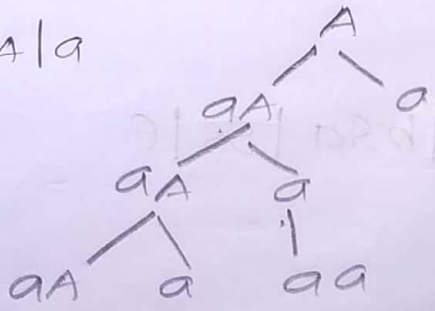
$A \rightarrow a$ : It produce only a
$A \rightarrow aA$ : It produce only aa.

1. Write a grammar to generate $a^+$

$\Sigma^* = \{a, aa, aaa, \ldots\}$

In the rule, we cannot write $\epsilon$ direcly because $\Sigma^*$ doesnot contains $\epsilon$.

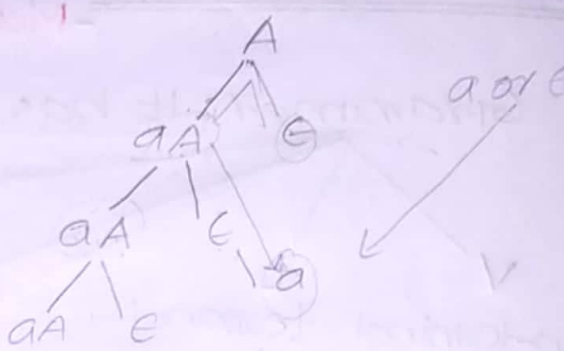
$\therefore A \rightarrow aA \mid a$



2. Write a grammar to generate $a^n | n \geq 0$;

$\Sigma^* = \{\epsilon, a, aa, aaa, \ldots\}$

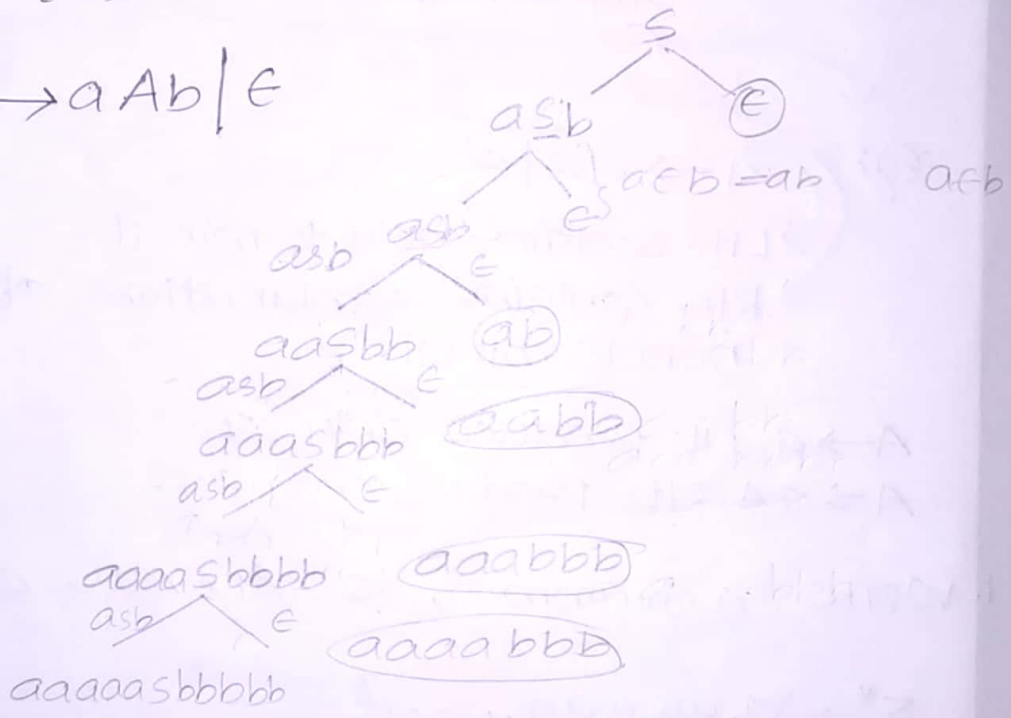
$A \rightarrow aA | \epsilon$




3. Write a grammar to generate $a^n b^n | n \geq 0$

$\Sigma^* = \{\epsilon, ab, a^2b^2, a^3b^3 \ldots\}$

$A \rightarrow aAb | \epsilon$




4. Write a grammer to generate equal no. of a's & b's

$S \rightarrow aSb | bSa | SS | \epsilon$

Not complete

# Different Types of Grammar

Type 0 Grammar : Not possible terminal only combination. The combinations are possible.

$(VUT)^+ \longrightarrow (VUT)^*$ : Unrestricted Grammar

Type 1 Grammar : It accept both V & T but (terminal only) $t \longrightarrow A$ is not possible. ie, $At \longrightarrow A$ is possible

$(VUT)^+ \longrightarrow (VUT)^+$ : Context sensitive

Type 2 Grammar → doesn't accept $\epsilon$

$V \longrightarrow (VUT)^*$ : Context free

Type 3 Grammar → It accept $\epsilon$

$V \longrightarrow (VUT)^+$ : Regular Grammar

Context Free Grammar

Parse Tree / Derevation Tree.