• Parse Tree/ Derivation Tree/ Syntax Tree

- Tree representation of derivation
- Leaves: labeled by a Terminal or ∈
- Interior Nodes: labeled by a Variable
- Root: must be labeled by the start symbol
- Children (left to right) are labeled by the body of the production rule

Example 1:

$$\mathsf{A}\to\mathsf{BC}$$

$$B \rightarrow a \mid b$$

$$C \rightarrow \in$$

Example 2: Balanced parenthesis

$$S \rightarrow SS \mid (S) \mid ()$$

Example 3:

$$S \rightarrow 0A1 \mid 0S$$

$$\mathsf{A} \to \mathsf{A1} \mid \; \in$$

Membership checking

- If possible, to draw a Parse Tree then the given string is a member of the Language

Example 4:

$$S \rightarrow 0A1 \mid 0S$$

$$A \rightarrow A1 \mid \in$$

$$w = 0001111$$

• Yield of the Parse Tree:

- The concatenation of the labels of the leaves in left to right order
- Preorder traversal

• Derivation order:

- **Left derivation** if we always derive the left most Variable
- **Right derivation** if we always derive the right most Variable

Example 5:

 $S \rightarrow aB \mid bA$

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

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

w = aabbabba

| Left der | ivation | Right De | rivation |
|----------|---------------------|----------|---------------------|
| S | | S | |
| аВ | $S \rightarrow aB$ | аВ | $S \rightarrow aB$ |
| aaBB | $B \rightarrow aBB$ | aaBB | $B \rightarrow aBB$ |
| aabB | $B \rightarrow b$ | aaBbS | $B \rightarrow bS$ |
| aabbS | $B \rightarrow bS$ | aaBbbA | $S \rightarrow bA$ |
| aabbaB | $S \rightarrow aB$ | aaBbba | $A \rightarrow a$ |
| aabbabS | $B \rightarrow bS$ | aabSbba | $B \rightarrow bS$ |
| aabbabbA | $S \rightarrow bA$ | aabbAbba | $S \rightarrow bA$ |
| aabbabba | $A \rightarrow a$ | aabbabba | $A \rightarrow a$ |

Example 6:

 $S \rightarrow AB \mid \in$

 $A \rightarrow aB$

 $B \rightarrow Sb$

w = abb

| Left de | erivation | Right Derivation | |
|---------|---------------------|------------------|---------------------|
| S | | S | |
| AB | $S \rightarrow AB$ | AB | $S \rightarrow AB$ |
| aBB | $A \rightarrow aB$ | ASb | $B \rightarrow Sb$ |
| aSbB | $B \rightarrow Sb$ | Ab | $S \rightarrow \in$ |
| abB | $S \rightarrow \in$ | aBb | $A \rightarrow aB$ |
| abSb | $B \rightarrow Sb$ | aSbb | $B \rightarrow Sb$ |
| abb | $S \rightarrow \in$ | abb | $S \rightarrow \in$ |

Example 7:

 $S \rightarrow aSb \mid aX$

 $X \rightarrow aY \mid a$

 $Y \rightarrow bY \mid \in$

w1 = aabbb

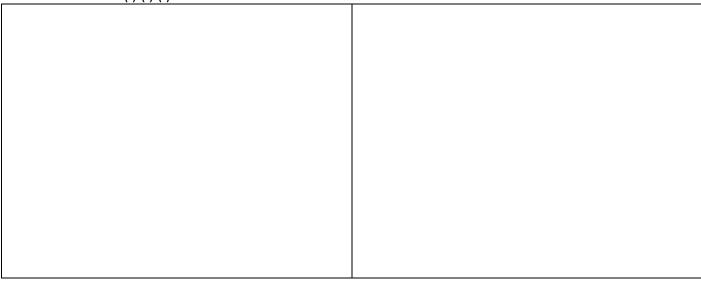
• Ambiguous Grammar:

- A CFG is ambiguous if there is a string in the Language that is Yield of 2 or more Parse Trees
- There is a string in the Language that has 2 different Left derivations
- There is a string in the language that has 2 different Right derivations

| Example 7: Balanced parenthesis |
|--|
|--|

$$S \rightarrow SS \mid (S) \mid ()$$

 $W = ()()()$



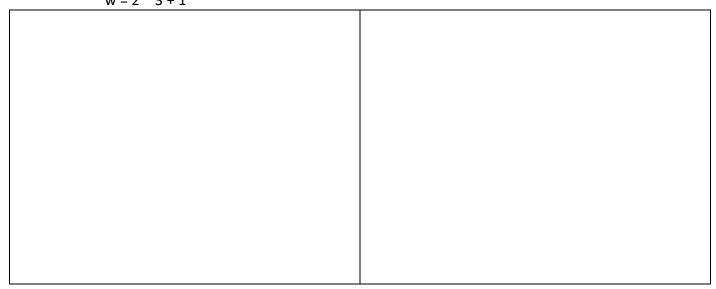
** Unambiguous grammar for Balanced parenthesis:

$$B \rightarrow (RB \mid \in R \rightarrow) \mid (RR \mid R \rightarrow) \mid (RR \mid R \rightarrow R \rightarrow))$$

Example 8:

$$E \rightarrow E + E \mid E * E \mid 1 \mid 2 \mid 3$$

w = 2 * 3 + 1



** Unambiguous grammar:

$$E \rightarrow E + F \mid F$$

 $F \rightarrow F * D \mid D$

$$D \rightarrow 1 \mid 2 \mid 3$$

-
$$w_1 = 2 * 3 + 1$$

-
$$w_2 = 2 + 3 * 2 + 4 + 1$$

-
$$w_3 = 2 + 3 * 2 + 3 + 1$$

- How to generate Grammar Rules:
 - 1. $L = \{0^n 1^n \mid n \ge 0\}$
 - 2. $L = \{0^n 1^n \mid n \ge 1\}$
 - 3. L = 1(0 + 1)*
 - 4. L = (0 + 1)*0
 - 5. L = 0n1m |
 - a. $|n, m \ge 0$
 - b. $| n \ge 0, m > 0$
 - c. $| n > 0, m \ge 0$
 - d. | n > 0, m > 0
 - 6. L = Even palindrome, $\Sigma = \{a, b\}$
 - 7. L = Odd palindrome, $\Sigma = \{a, b\}$
 - 8. L = string holds at least 3 b's, Σ = {a, b}
 - 9. L = all nonempty strings that start and end with the same symbol