Al-Hamdaniya University

College of Education

Computer Science

Stage: 2nd

Regular Grammar

Regular grammar generates regular language. They have a single non-terminal on the left-hand side and a right-hand side consisting of a single terminal or single terminal followed by a non-terminal.

The productions must be in the form:

```
A → xB
A → x
A → Bx
where A, B ∈ Variable(V) and x ∈ T*  i.e. string of terminals.
```

Types of regular grammar:

- Left Linear grammar(LLG)
- Right linear grammar(RLG)

1. **Left linear grammar(LLG):**
   In LLG, the productions are in the form if all the productions are of the form

```
A → Bx
A → x
where A,B ∈ V and x ∈ T*
```

2. **Right linear grammar(RLG):**
   In RLG, the productions are in the form if all the productions are of the form

$$A \dashrightarrow xB$$
$$A \dashrightarrow x$$
$$\text{where } A,B \in V \text{ and } x \in T^*$$

Example: Derived a^4 from the following grammar: $A \longrightarrow aA \mid a$

$$A \longrightarrow aA$$
$$\longrightarrow aaA$$
$$\longrightarrow aaaA$$
$$\longrightarrow aaaa$$
$$\longrightarrow a^4$$

Note: We note in these rules that it is possible to repeat in any number of steps and stop at any stage of derivation, and the general form of the words resulting from these rules is:

$\{a^n, n \geq 1$ by n steps$\}$

Phrase Structure Grammar (PSG)

Note: These are the rules that appear in the tree form in the form of intermediate nodes that eventually turn into the terminal symbols, and they also consist of four groups:

PSG = (N, T, P, S)

- N = a finite set of non-terminals.
- T= a finite set of terminals.
- P= a finite phrase structure rules.
- S= a start symbol representing the whole sentences.

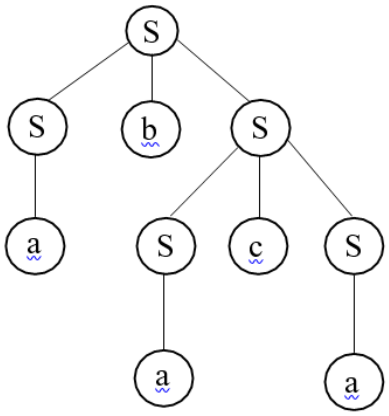
Left Most Derivation

Example: Let G(L) = ({S}, {a, b, c}, P, S), where P is:

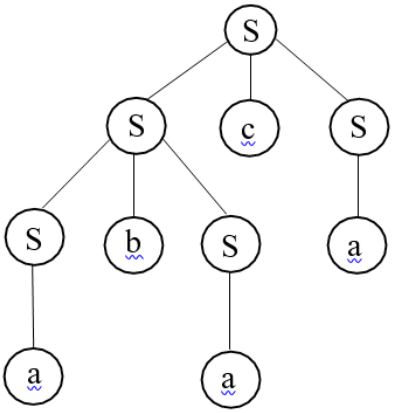
S → SbS | ScS | a

Find the string "abaca"

S → SbS
→ abS
→ abScS
→ abacS
→ abaca Accept

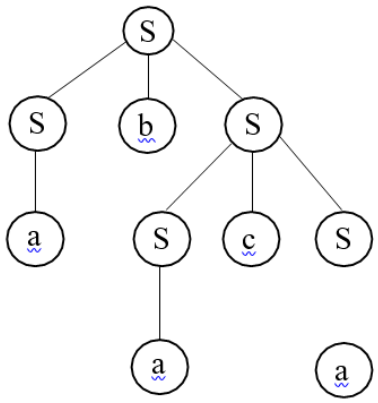


S → ScS
→ SbScS
→ abScS
→ abacS
→ abaca Accept

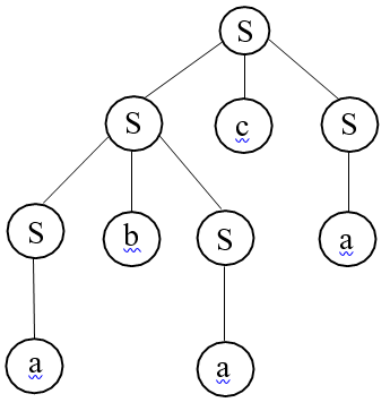


Right Most Derivation

$S \rightarrow SbS$
$\rightarrow SbScS$
$\rightarrow SbSca$
$\rightarrow Sbaca$
$\rightarrow abaca$ Accept



$S \rightarrow ScS$
$\rightarrow Sca$
$\rightarrow SbSca$
$\rightarrow Sbaca$
$\rightarrow abaca$ Accept



Note: *In the previous example, we noticed that the second solution is in the same direction as the first solution, consisting of a derivation from the right and a derivation from the left, and it has the same tree.*

We conclude from this that when there is more than one solution on the left side or more than one solution on the right side, the rule is called ***ambiguous***

Ambiguous and Unambiguous Grammar

Ambiguous Grammar

$\forall X: X \in L(G)$ *if there are two distinct derivations on the right side or two distinct derivations on the left side, then the rule is called* ***ambiguous***

Unambiguous Grammar

$\forall X: X \in L(G)$ *It gives the same tree derivation, whether from the right or the left.*

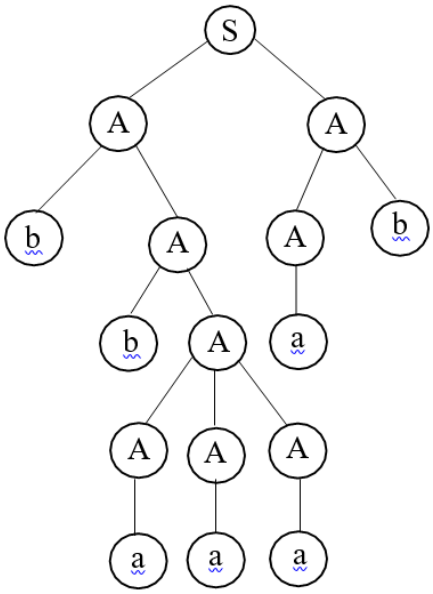
Example: Let G(L) = ({S, A}, {a, b}, P, S), where P is:

S → AA

A → AAA | bA | Ab | a

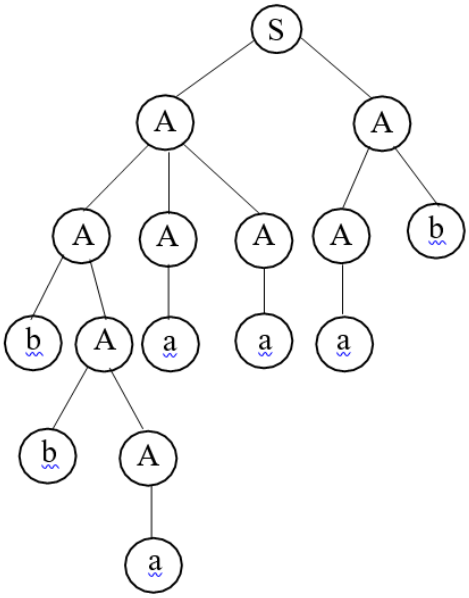
- Is the string "bbaaaab" Accept or not?
- Is the grammar ambiguous or not?

(Left) S → AA
→ bAA
→ bbAA
→ bbAAAA
→ bbaAAA
→ bbaaAA
→ bbaaaA
→ bbaaaAb
→ bbaaaab Accept



(Left) S → AA
→ AAAA
→ bAAAA
→ bbAAAA
→ bbaAAA
→ bbaaAA
→ bbaaaA
→ bbaaaAb
→ bbaaaab Accept

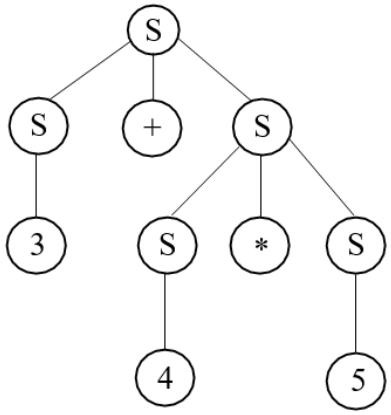
The grammar is ambiguous



Example: Let the following grammar:

$$S \rightarrow S + S \mid S * S \mid \text{digit}$$

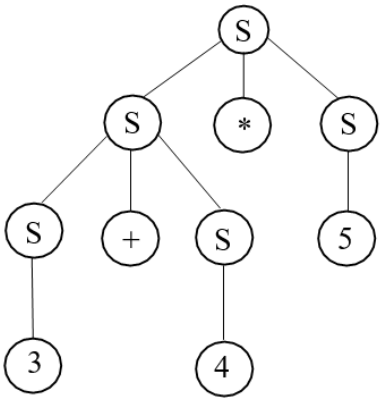
- Is the string "3 + 4 * 5" Accept or not?
- Is the grammar ambiguous or not?



(Left) $S \longrightarrow S + S$
$\longrightarrow 3 + S$
$\longrightarrow 3 + S * S$
$\longrightarrow 3 + 4 * S$
$\longrightarrow 3 + 4 * 5$ Accept

(Left) $S \longrightarrow S * S$
$\longrightarrow S + S * S$
$\longrightarrow 3 + S * S$
$\longrightarrow 3 + 4 * S$
$\longrightarrow 3 + 4 * 5$ Accept

The grammar is ambiguous



Homework: Let the following grammar:

$S \rightarrow aSa \mid bSb \mid a \mid b \mid A$

- Is the string "aabaa" Accept or not?
- Is the grammar ambiguous or not?