



# SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY  
(U/S 3 OF THE UGC ACT, 1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

# CSE211 – Formal Languages and Automata Theory

## Unit 1-L3: Grammars and Dérivations

**Dr. P. Saravanan**

School of Computing

SASTRA Deemed University

# Agenda

- Why Grammars for a language?
- What is a grammar?
- What is derivation?
- What is sentential form?
- How to find Language generated by grammar?
- How to find Grammar to generate language?

# Grammar

- A formal grammar is **a set of rules** for rewriting strings, along with a "start symbol" from which rewriting starts
- It is a mechanism **to describe** languages
- A grammar for the **English language** tells us whether a particular sentence is well-formed or not
- A typical rule of English grammar is “**a sentence can consist of a noun phrase followed by a predicate.**”
- More concisely we write this as

$$\langle sentence \rangle \rightarrow \langle noun\_phrase \rangle \langle predicate \rangle$$

# Example: Grammar for Sentence

- The set of grammar rules for the sentence formation in English language

$$\langle sentence \rangle \rightarrow \langle noun\_phrase \rangle \langle predicate \rangle$$
$$\langle noun\_phrase \rangle \rightarrow \langle article \rangle \langle noun \rangle$$
$$\langle predicate \rangle \rightarrow \langle verb \rangle$$
$$\langle article \rangle \rightarrow a \quad \langle noun \rangle \rightarrow cat \quad \langle verb \rangle \rightarrow runs$$
$$\langle article \rangle \rightarrow the \quad \langle noun \rangle \rightarrow dog \quad \langle verb \rangle \rightarrow sleeps$$

# Derivation of String in English

- Consider “the dog sleeps”

$\langle sentence \rangle \Rightarrow \langle noun\_phrase \rangle \langle predicate \rangle$   
 $\Rightarrow \langle noun\_phrase \rangle \langle verb \rangle$   
 $\Rightarrow \langle article \rangle \langle noun \rangle \langle verb \rangle$   
 $\Rightarrow the \langle noun \rangle \langle verb \rangle$   
 $\Rightarrow the \ dog \langle verb \rangle$   
 $\Rightarrow the \ dog \ sleeps$

# Derivation of String in English

- Consider “the cat eats”

$\langle sentence \rangle \Rightarrow \langle noun\_phrase \rangle \langle predicate \rangle$

$\Rightarrow \langle noun\_phrase \rangle \langle verb \rangle$

$\Rightarrow \langle article \rangle \langle noun \rangle \langle verb \rangle$

$\Rightarrow the \langle noun \rangle \langle verb \rangle$

$\Rightarrow the \ cat \langle verb \rangle$

$\Rightarrow the \ cat \ eats$

# Language of the grammar

$\langle sentence \rangle \rightarrow \langle noun\_phrase \rangle \langle predicate \rangle$

$\langle noun\_phrase \rangle \rightarrow \langle article \rangle \langle noun \rangle$

$\langle predicate \rangle \rightarrow \langle verb \rangle$

$L = \{ \text{"a cat runs", "a cat sleeps",}$   
 $\text{"the cat runs", "the cat sleeps",}$   
 $\text{"a dog runs", "a dog sleeps",}$   
 $\text{"the dog runs", "the dog sleeps" } \}$

# Production of Grammar

Sequence of  
Terminals (symbols)

$\langle \textit{noun} \rangle \rightarrow \textit{cat}$

$\langle \textit{sentence} \rangle \rightarrow \langle \textit{noun\_phrase} \rangle \langle \textit{predicate} \rangle$

Variables

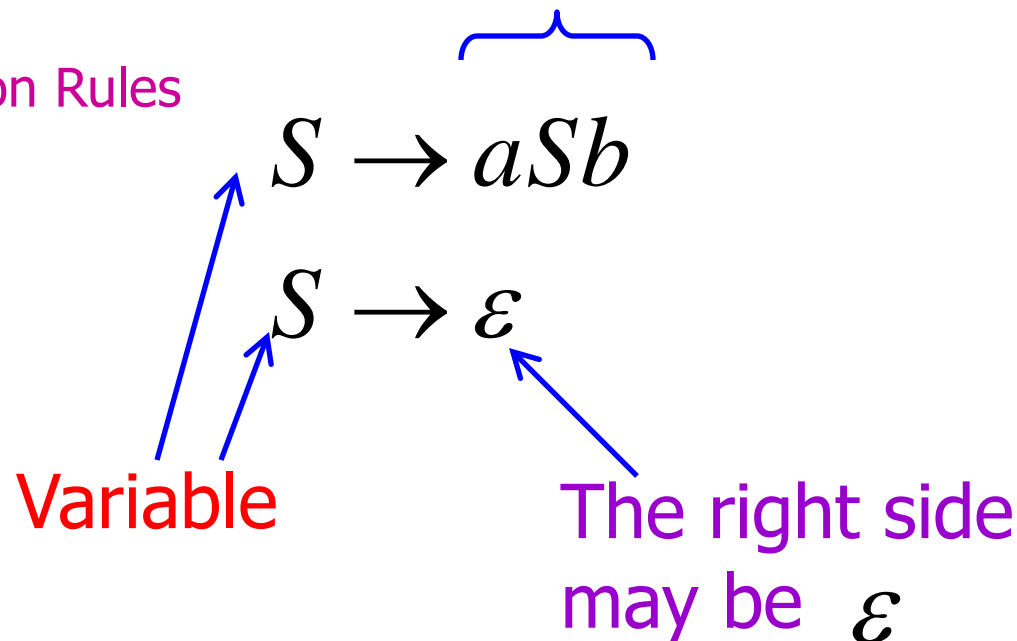
Sequence of Variables



# Production of Grammar ...

Sequence of  
terminals and variables

Production Rules



# Derivation of String

- Production rules:

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

- Derivation of string :  $ab$

$$S \Rightarrow aSb \Rightarrow ab$$


$$\begin{array}{ccc} & S \Rightarrow aSb \Rightarrow ab & \\ \nearrow & & \nwarrow \\ S \rightarrow aSb & & S \rightarrow \lambda \end{array}$$

# Derivation of String...

- Production:  $S \rightarrow aSb$

$$S \rightarrow \lambda$$

- Derive the string:

*aaabbb*

String is a part of the language generated by the grammar

$$S \Rightarrow aSb$$

$$[\because S \rightarrow aSb]$$

$$\Rightarrow aaSbb$$

$$[\because S \rightarrow aSb]$$

$$\Rightarrow aaaSbbb$$

$$[\because S \rightarrow aSb]$$

$$\Rightarrow aaabbb$$

$$[\because S \rightarrow \lambda]$$

# Derivation of String...

- Production rules:

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

- Derivation of string :  $aabb$

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$$

Diagram illustrating the derivation of the string  $aabb$  from the start symbol  $S$  using the production rules  $S \rightarrow aSb$  and  $S \rightarrow \lambda$ . The derivation steps are shown as  $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$ . Pink arrows indicate the application of the rules: the first arrow points from  $S \rightarrow aSb$  to the first  $aSb$  in the sequence; the second arrow points from  $S \rightarrow aSb$  to the second  $aSb$  in the sequence; and the third arrow points from  $S \rightarrow \lambda$  to the final  $aabb$  in the sequence.

# Derivation of String...

- Productions:  $S \rightarrow aSb$

$$S \rightarrow \lambda$$

- Derive the string:

*aaabb*

String is not a part of the language generated by the grammar

$$S \Rightarrow aSb$$

$$[\because S \rightarrow aSb]$$

$$\Rightarrow aaSbb$$

$$[\because S \rightarrow aSb]$$

$$\Rightarrow \text{incomplete}$$

# Language Accepted by Grammar

Productions:  $S \rightarrow aSb$

$$S \rightarrow \lambda$$

Language accepted by the grammar:

$$L = \{\lambda, ab, aabb, aaabbb, aaaabbbb, \dots\}$$

Language accepted by the grammar in general:

$$L = \{a^n b^n : n \geq 0\}$$

# Formal Definition: Grammar

- A grammar  $G$  is defined as a quadruple

$$G = (V, T, S, P),$$

where  $V$  is a finite set of objects called **variables**,  
 $T$  is a finite set of objects called **terminal symbols**,  
 $S \in V$  is a special symbol called the **start variable**,  
 $P$  is a finite set of **productions**.

- It will be assumed that the sets  $V$  and  $T$  are nonempty and disjoint.

# Summary

- Why Grammars for a language
- What is a grammar?
- What is derivation?
- What is sentential form?
- Language generated by grammar
- Grammar to generate language



# References

- John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Pearson, 3<sup>rd</sup> Edition, 2011.
- Peter Linz, *An Introduction to Formal Languages and Automata*, Jones and Bartle Learning International, United Kingdom, 6<sup>th</sup> Edition, 2016.

---

Next Class:

Examples for Grammar and  
Derivation

**THANK YOU.**