SNS COLLEGE OF TECHNOLOGY

Coimbatore-36.

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++'
Grade Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

COURSE CODE& NAME : *19CSB301* &AUTOMATA THEORY AND COMPILER DESIGN

III YEAR/ V SEMESTER

UNIT – I FINITE AUTOMATA AND REGULAR LANGUAGES

Topic: Types of Grammar

Dr.B.Vinodhini

Associate Professor

Department of Computer Science and Engineering

Types of Grammar

- ***Grammar in Automata***
  - **G = (V, T, P, S)**
  - V – Non-Terminals / Variables /Auxillary Symbols (A,B,C,….)
    - Takes part in generation of sentence (Not a part of sentence)
  - T – Terminals (small-case letters a,b,c,….)
  - P – Production Rules
  - S – Start Symbol

Example1

V = {S}

T = {a, b}

P = {S→aSbS, S→bSaS, S→ ∈}

S = {S}

Example2

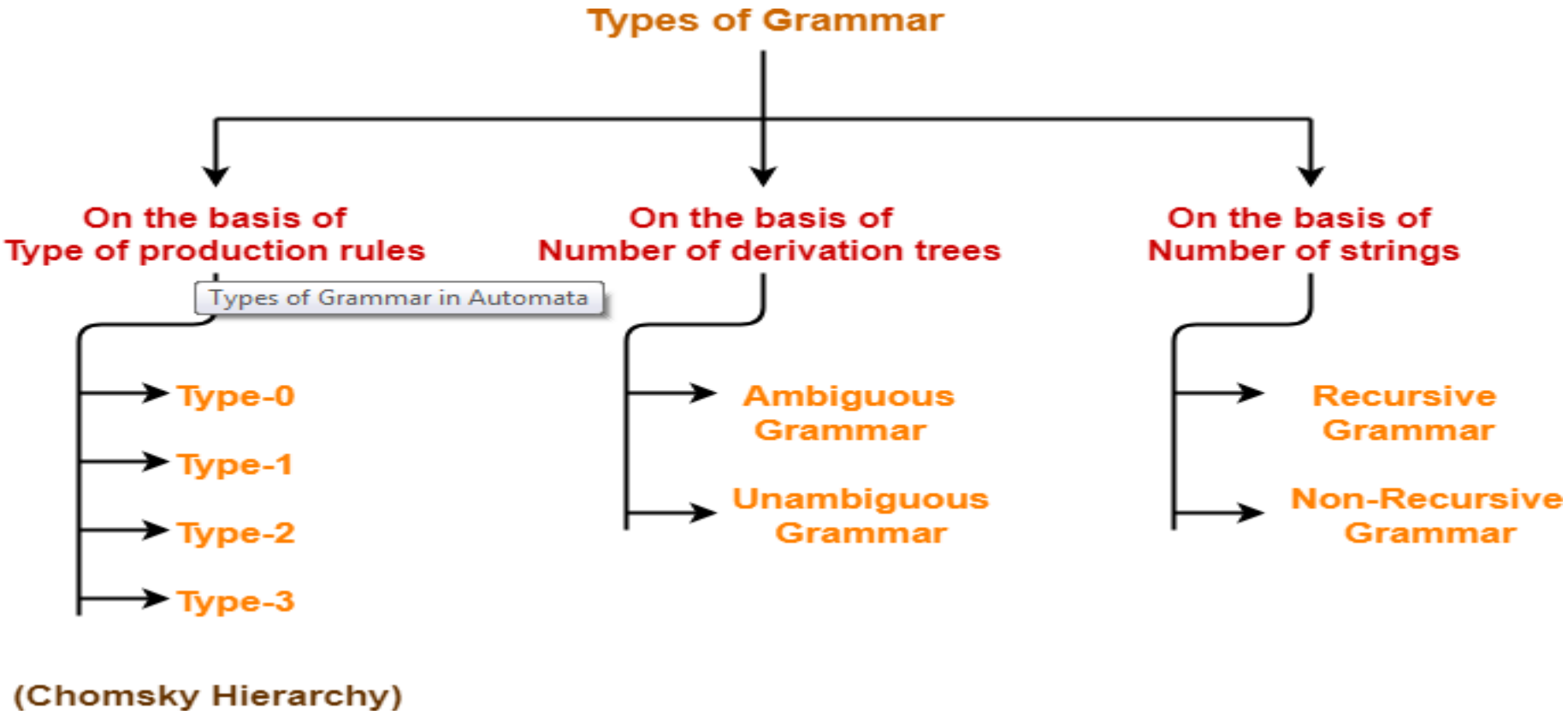
V = {S,A,B}

T = {a,b}

P = {S→ABA, A→BB, B→ab, AA→b}

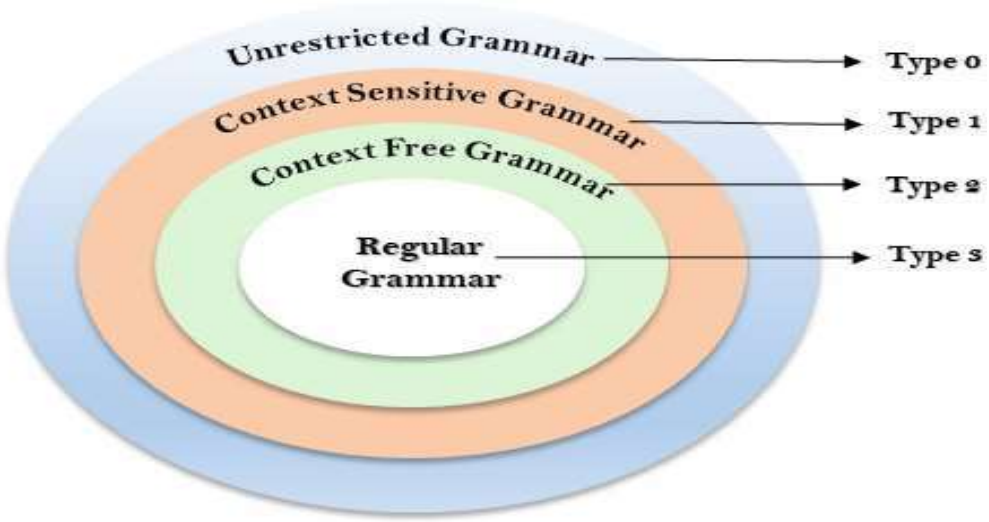
S = {S}

T ypes of Grammar



Chomsky Hierarchy

Grammar Type	Grammar Accepted	Language Accepted	Automaton
Type 0	Unrestricted grammar	Recursively enumerable language	Turing Machine
Type 1	Context-sensitive grammar	Context-sensitive language	Linear-bounded automaton
Type 2	Context-free grammar	Context-free language	Pushdown automaton
Type 3	Regular grammar	Regular language	Finite state automaton



Type 3 ⊆ Type2, Type 1, Type 0
Type 2 ⊆ Type 1, Type 0
Type 1 ⊆ Type 0

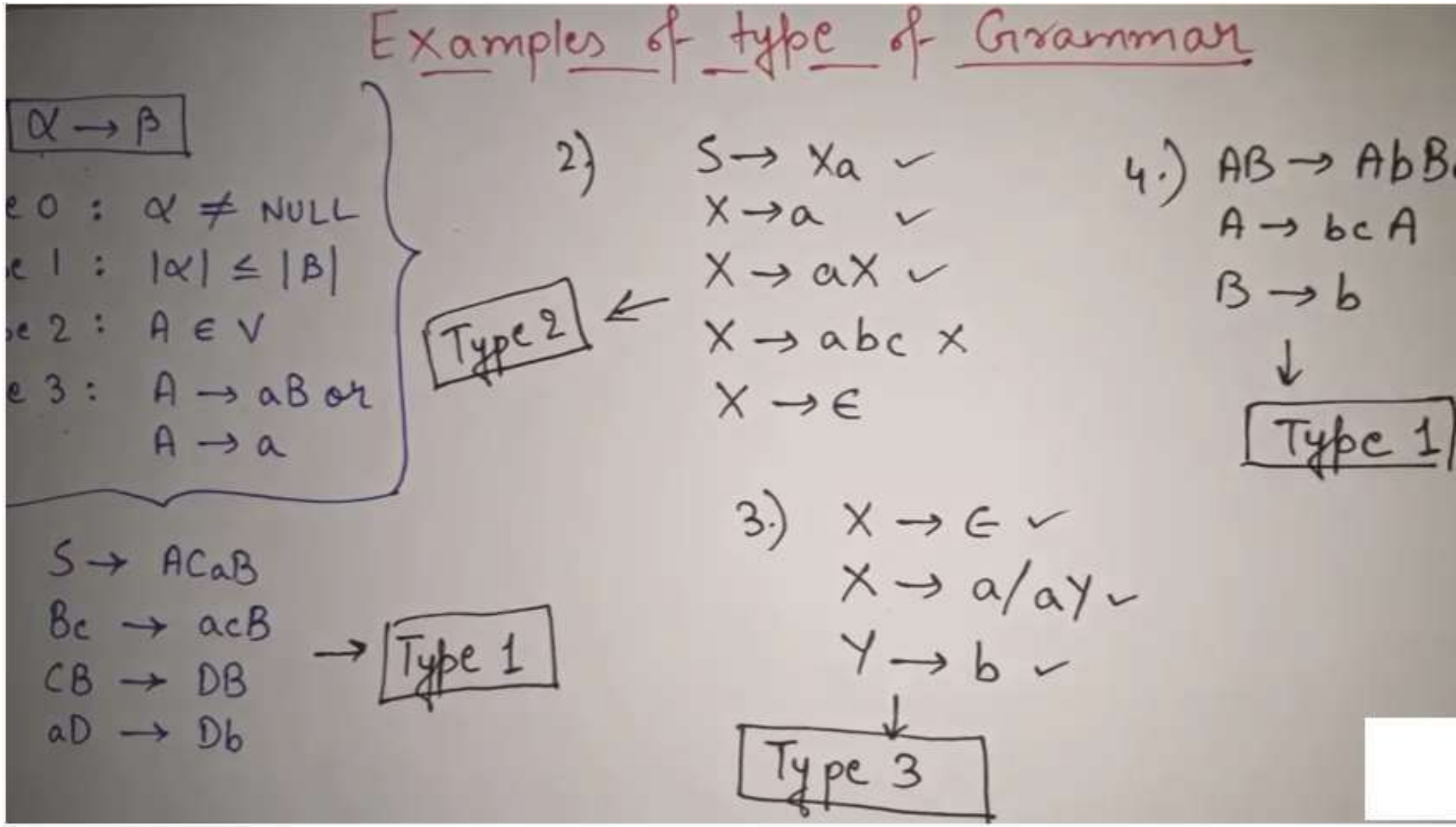
Chomsky Hierarchy

- Type 0 (Unrestricted)
  - $\alpha \rightarrow \beta$
  - $\alpha \in (V+T)^+$ ← excluding $\in$
  - $\beta \in (V+T)^*$ ← including $\in$
  - $\alpha \neq \in$
- Type 1 (Context Sensitive Grammar)
  - $\alpha \rightarrow \beta$
  - $|\alpha| <= |\beta|$
- Type 2 (Context Free Grammar)
  - $\alpha \rightarrow \beta$
  - $\alpha \in V$
  - $\beta \in (V+T)^*$
- Type 3 (Restricted)
  - $V \rightarrow VT^*/T^*$ (Right Regular Language) OR T^*V/T^* (Left Regular language)
  - Example: $A \rightarrow aB$, $A \rightarrow a$

Chomsky Hierarchy

Rules

Type 0 → α ≠ ∈
Type 1 → | α|<= | β|
Type 2 → α ∈ V, β ∈ (V+T)*
Type 3 -> α->aB or α->a



References

- John E. Hopcroft and Rajeev Motwani and Jeffrey D. Ullman, “Introduction to Automata Theory, Languages and Computation”, Second Edition, Pearson Education, New Delhi, (2007) (UNIT-I)
- Linz P. An introduction to formal languages and automata. Sixth edition, Jones and Bartlett Publishers; 2016.(UNIT-I)
- Ramaiah k. Dasaradh “Introduction to Automata and Compiler Design “ First Edition ,Prentice Hall India Learning Private Limited(2011)(UNIT-I to V)


Thank
you