☆ Chomsky hierarchy of Grammars & Languages

↳ Noam Chomsky

G L M applica"

Type 0 : Recursively Enumerated Grammars : REL ↔ Turing Machine (TM) : compute

Type 1 : Context Sensitive Grammar : CSL ↔ Linear Bounded Automata (LBA) : syntax & semantics checking

Type 2 : Context Free Grammar : CFL ↔ PDA (Push Down Automata) FA + stack : syntax check

Type 3 : Regular Grammar : RL ↔ Finite Automata FA : Pattern matching

Type-0 | Recursively enumerated | Unrestricted Grammar

REG ↓ REL / TM

V : Variables
T : Terminals
S : start symbol

$\alpha \to \beta$, $\alpha \in (V+T)^*$

$\beta \in (V+T)^*$

Ex:
$S \to aAB$
$aA \to a|b$
$Ba \to b$

Ex (REL) : $L = \{a^{n!} \mid n \geq 0\}$

Type-1 | Context Sensitive Grammars

CSL / LBA

$\forall\ \alpha \to \beta$, $\alpha \in (V+T)^*$

$\beta \in (V+T)^+$

CSG should satisfy :

1) $|\alpha| \leq |\beta|$

2) $\beta \neq \epsilon \Rightarrow$ CSG can't generate ϵ

Ex :
CSL ⇒ $L = \{a^n b^n c^n \mid n \geq 1\}$

Ex CSG : $S \to aAB$
$aA \to aB|bb$
$Ba \to bb$
$(Ba \to b)$ ✗ not allowed as $|\alpha| \leq |\beta|$

* Type-2 / Context Free Grammar

CFL / PDA

$\alpha \to \beta$, $\alpha \in V$

$\beta \in (V+T)^*$

Ex. (CFG) $S \to aSa \mid \epsilon$

$(bB \to bb)$ ✗ not allowed, $\alpha \in V$, bB ↑

CFL

Ex. $L = \{a^n b^n \mid n \geq 1\}$

* Type-3 / Regular Grammar

$A \to xB$

$A \to x$

$A \to Bx$

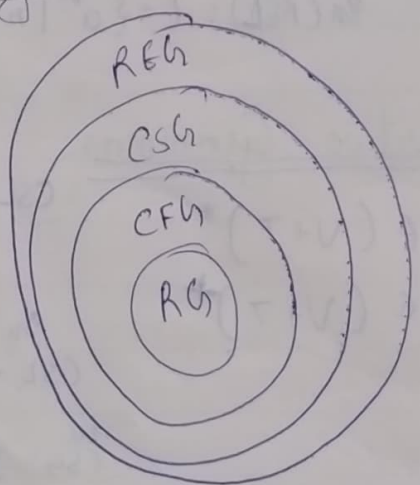
$A, B \in V$

$x \in T^*$ → string of terminals

Ex. ⇒ $S \to 00115 \mid 01 \mid \epsilon$

* By ignoring ϵ's

↙ As CSG can't generate ϵ whereas CFG & RG can



Similar for languages & machines.