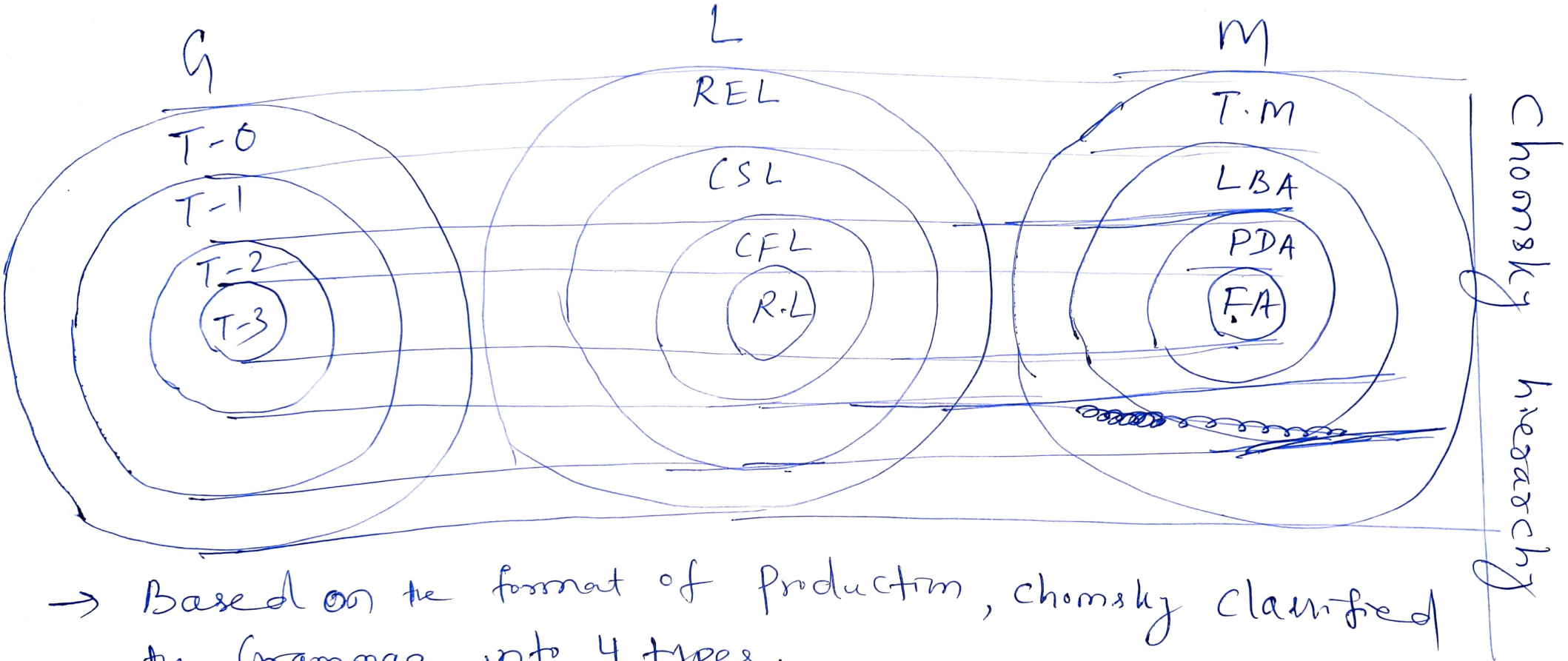

→ Based on the format of production, chomsky classified the Grammar into 4 types.

① Type(0) or REG
② Type(1) or CSG
③ Type(2) or CFG
④ Type(3) or RG

Type-0 Grammar - Recursive enumerable grammar/unrestricted Grammar

→ used to generate Recursive enumerable Language (REL) which is accepted by Turing Machine (T.M)

R.E.G
↓
R.E.L
↓
T.M

The Grammar G is said to be Type (0) or REG if the production is in the form

$\alpha \longrightarrow \beta$

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

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

Type-1 Grammar - Context sensitive grammar/length increasing grammar

→ It generates context sensitive language which is accepted by Linear bounded Automata (LBA)

→ The Grammar G is said to be Type(1) or CSG if the production is in the form

$\alpha \longrightarrow \beta$

$|\alpha| \leq |\beta|$ (Length of α is less than or equal to β)

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

$\beta \in (V+T)^+ \{ \beta \neq \epsilon \}$

$AaB \rightarrow bb$ ✗

$Aa \rightarrow bb$ ✓

Note: In CSG we donot have any production to Generate empty string ϵ. i.e the Language which is generated by CSG does not contain ϵ.

Type-2 Grammar - Context Free Grammar

→ It generates Context Free Language which is accepted by Push down automata.

→ The Grammar G is said to be Type(2) or CFG if the production is in the form

$\alpha \rightarrow \beta$

$\alpha \in V$, $|\alpha| = 1$ no. of variable will be 1

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

So at LHS only one variable, but at RHS no restriction.

~~Type-3 Grammar~~

CFG
↓
CFL
↓
PDA

Type-3 Grammar - Regular Grammar

→ It generates regular language which is accepted by finite automata.

Left Linear Grammar

$A \rightarrow a \mid Ba$

$A, B \in V, |A| = |B| = 1$

$a \in T^*$

Right Linear Grammar

$A \rightarrow a \mid aB$

$A, B \in V, |A| = |B| = 1$

$a \in T^*$

RG
↓
RL
↓
FA