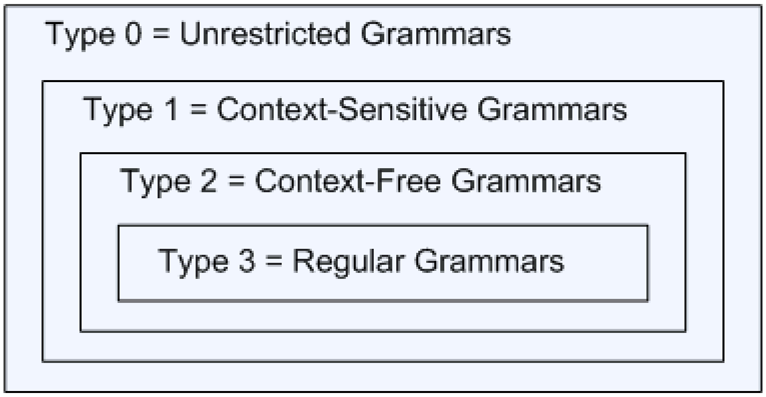
GRAMMARS

DEFINITION

- A **grammar** is a 4-tuple G=(N, T, S, P)
  - N = set of **non-terminal** symbols
  - T = set of **terminal** symbols = set of tokens
  - $N \cap T = \varnothing$
  - S = **start** symbol
  - P = set of **productions** $\subset (N \cup T)^+ \times (N \cup T)^*$
- Notation: $(\alpha, \beta) \in P$ is written as $\alpha \rightarrow \beta$

CHOMSKY HIERARCHY



Type 0 = Unrestricted Grammars

Type 1 = Context-Sensitive Grammars

Type 2 = Context-Free Grammars

Type 3 = Regular Grammars

- Type 0 = **unrestricted** grammars describe **recursively enumerable** languages
  - $P \subset (N \cup T)^+ \times (N \cup T)^*$
  - Recognised by Turing Machines
- Type 1 = **context-sensitive** grammars describe context-sensitive languages
  - $P \subset [\ (N \cup T)^* \, N \, (N \cup T)^*\ ] \times (N \cup T)^*$
  - Recognised by linear bounded automata
- Type 2 = **context-free** grammars describe context-free languages
  - $P \subset N \times (N \cup T)^*$
  - Recognised by Push-Down Automata (PDA)
- Type 3 = **regular** grammars describe regular languages
  - $P \subset N \times T(N \cup \{\varepsilon\})^*$
  - Recognised by Finite State Automata (FSA)
  - Every regular expression can be described by a regular grammar and vice-versa.