Noam Chomsky

By Zach Baker

Noam Chomsky

"I was never aware of any other option but to question everything."

Early life

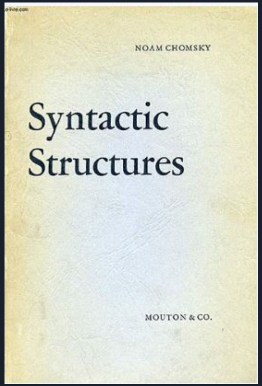
- Avram Noam Chomsky was born on December 7, 1928, in Pennsylvania
- Interested in politics
- Jewish upbringing
- Had a younger brother

Life at M.I.T.

After graduating from the University of Pennsylvania, Chomsky went on to obtain a doctorate in linguistics from Harvard.

From there, he went on to study at M.I.T., where he became an assistant professor.

- Published multiple books, papers, and reviews.
- Argued for the separation of syntax from semantics.
- Argued against B.F. Skinner, who said language is a learned behavior.
- Argued humans must have some “language acquisition device,” since exposing an animal to language only resulted in an animal.



Contributions to Formal Systems

Chomsky developed a formal definition of a grammar, of which there are four types, and further organized them into Chomsky's Hierarchy, where each type of grammar is a subclass of each grammar before it.

Grammar	Languages	Automaton	Production rules (constraints)*	Examples[3]
Type-0	Recursively enumerable	Turing machine	$\gamma \rightarrow \alpha$ (no constraints)	$L = \{w \| w$ describes a terminating Turing machine$\}$
Type-1	Context-sensitive	Linear-bounded non-deterministic Turing machine	$\alpha A \beta \rightarrow \alpha \gamma \beta$	$L = \{a^n b^n c^n \| n > 0\}$
Type-2	Context-free	Non-deterministic pushdown automaton	$A \rightarrow \alpha$	$L = \{a^n b^n \| n > 0\}$
Type-3	Regular	Finite state automaton	$A \rightarrow \mathrm{a}$ and $A \rightarrow \mathrm{a}B$	$L = \{a^n \| n \geq 0\}$

* Meaning of symbols:

- a = terminal
- A, B = non-terminal
- α, β, γ = string of terminals and/or non-terminals
  - α, β = maybe empty
  - γ = never empty

Unrestricted Grammar

An unrestricted (AKA recursively enumerable) grammar is a system which contains a set of possible symbols, a starting symbol, and a list of rules for transforming or substituting said symbols.

A property of such a system is that a turing machine is capable of listing all possible valid strings, if given infinite time.

Context-Sensitive Grammar

A context sensitive grammar contains symbols whose substitutions depend on the context of their placement in a string, so one may have a rule which replaces ‘b’ with ‘c’ - but only when there is a ‘b’ before ‘a’. So, [ab -> c].

- An unrestricted grammar may have an empty substitution, such as [ab ->], while a context-sensitive grammar cannot.

Context-Free Grammar

A context-free grammar is like a context-sensitive grammar, except it’s not context-sensitive. This means that each rule has only one left-hand argument. Ex. [a -> b, b -> bc, c -> ab].

Regular Grammar

A regular grammar has two qualifiers: each production rule must contain at most one non-terminal symbol, and that symbol is either always on the left (left-regular) or always on the right (right-regular).

As such, these grammars appear to only expand only either left or right.