The Chomsky Hierarchy

Spencer Hirsch, Thomas Johnson, Tyler Gutowski, Remington Greko

January 27, 2023

Noam Chomsky is a reowned linguist whose ideas were recognized and applied by computer scientists to lay foundations of our discipline. I admire him for his support of justice.

- Read about Noam Chomsky on Wikipedia and summarize his salient contributions to our computing discipline.
- Outline the Chomsky Hierarchy giving what we now call
  - Regular Languages
  - Context-Free Languages
  - Context-Sensitive Languages
  - Unrestricted Languages
- Read about Grammars Wikipedia. Report on what you find interesting.

1 Chomsky

Avram Noam Chomsky born December 7th 1928 is an American linguist, philosopher, and political activist. Dubbed as the father of modern linguistics, Chomsky helped establish the field of cognitive science.

Chomsky's early works saw him studying at the University of Pennsylvania to which he began studying at Harvard Society of Fellows during his post graduation. From Harvard he earned his doctorate in 1955 for his theory on transformational grammar, and soon released the book Syntactic Structures in 1957 which helped redevelop the study of language. In the years following Chomsky helped create the universal grammar theory. He then went on to help create the minimalist program, and lead a life of activism. He has since settled down to teach at the University of Arizona as of 2017.

In the context of computer science Chomsky has a few very notable works. His book **Syntactic Structures** contains his theory on **transformational-generative grammar** (originally published in his 1955 dissertation), and the **Chomsky hierarchy**.

Syntactic Structures covers the formality of syntax, and the symbols and rules comprising it. He does this via something he refers to as phase structures. He then applies his new rules called transformations. Without going too in depth these transformations through mathematical notation can generate all grammatical sentences of a language. This is the basis of formal languages, and how they may be constructed via grammars.

The Chomsky Hierarchy is a way of grouping all logical possible phrase-structure grammars into nested subsets with these grammars becoming more complex as we move up. This hierarchy contributed greatly to the study of formal language theory, programming language theory, compiler construction, and automata theory.

2 Chomsky Hierarchy

The Chomsky Hierarchy is a containment hierarchy of classes of formal grammars. A containment hierarchy is a type of nested hierarchy, where certain classes *are of* a similar class, but are *not equivalent* to the similar class.

A good example that is listed on Wikipedia is: **Square is a Quadrilateral is a Polygon is a Shape**

The Chomsky Hierarchy is split into four different grammar types, type-0, 1, 2 and 3. Type-0 encapsulates Type-1, which encapsulates Type-2, which encapsulates Type-3.

2.1 Regular Languages

Type-3 generates **regular grammar**, which are exactly all languages that can be accepted by a finite state automaton. Type 3 should only be given in the form of A → a. This means that the left side must have a single non-terminal symbol, and the right side must have a single terminal signal, which may be followed by a single non-terminal symbol. The most obvious example is regex, which offers the ability to define tokens. This allows a company to parse resumes searching for specific terms, such as "python", "java" and "object oriented", while discarding resumes that do not contain important keywords.

2.2 Context-Free Languages

Type-2 generates **context-free languages**. Of course, Type-2 must also be in Type-1 form. The left-hand side of production can only have one variable, and there is no restriction on β. The form is A → α. A is a single non-terminal symbol, and β is a string of symbols. Most speech falls into this category. Since nouns and their plurals can be recognized in a single non-deterministic finite automaton, then combined. An example is the singular form of ("The professor lectures to the class." → Professor / Lectures) as opposed to the plural form of ("The professors lecture to the class." → Professors / Lecture).

2.3 Context-Sensitive Languages

Type-1 generates **context-sensitive languages**. Of course, Type-1 grammar should also be in Type-0 form, and the grammar production must be in the form of $\alpha A\beta \rightarrow \alpha\gamma\beta$ That is, the count of the symbol in α is less than or equal to that in β. This also means that strings α and β may be empty, but similarly to Type-0, γ cannot be empty. Although most languages are context-free, there are many situations where a list of matching tokens is necessary, such as the statement "The square of 2, 3, and 4 are 4, 9 and 16, respectively." Here 2 and 4 are matched, 3 and 9 are matched, and 4 and 16 are matched.

2.4 Unrestricted Languages

Type-0 is in the form of **unrestricted languages**, which means it includes *all* formal grammars. They generate exactly all languages that can be recognized by a Turing machine. Note that this is different from Turing complete, which is designated by programming languages, and *not* formal languages. The form is $\alpha \rightarrow \gamma$. In Type-0, there must be at least one variable on the left side of production, meaning that γ must not be empty. An unrestricted language isn't beneficial towards communication, which means there are no good speech examples.

3 Grammars

Grammar can be used to describe behavior of groups of speakers, grammar can be used to demonstrate the language of not only an entire group of individuals but also subsets within that group. It's interesting how grammar changes depending on what sample of people you are looking at. Geography has an influence on the way that individuals speak.

With regards to history, it is interesting that the "earliest known grammar handbook is the *Art of Grammar*" was written by an ancient Greek scholar in 90 BC. Throughout history grammar has been based off of the grammar of other languages, each society basing it's grammar off of an older language, demonstrating that grammar that is used today have evolved from ancient languages. Grammar has also been a topic of study throughout many time periods including the Middle Ages.

Grammar evolves with time and usage most notabily through the observation and documentation. Sentences that have slight variations in language can have rather large differences based on context. It is interesting to think that grammar has defined rules however, the evolution of language changes the definition of such rules. Demonstrating that grammar rules are dicated by those who break grammar rules.

There is a difference in grammar in an academic setting and in the real world, vernacular dialects and standard language respectively.