## Formal Languages

The Turing machine is a mathematical model consistent with other models from the theory of computation. A quick review of the theory of computation and automata theory reveals that these models are defined in terms of the types of "languages" that can be recognized.

Let $\Sigma$ be an alphabet (i.e., a finite set of symbols). Then a language $L$, defined over $\Sigma$, is any set of strings made up from the symbols in $\Sigma$ conforming to some set of rules. Let $G$ be a grammar (i.e., the set of rules) for determining if a string is a member of $L$. Then complexity classes of languages can be defined based on how expressive the corresponding language is.

### Regular Languages

A regular language is a language $L$ such that there exists a deterministic finite automaton (DFA) that accepts the language. When examining the grammar of a regular language, we observe that regular languages are either "right linear" or "left linear".

A grammar is said to be right linear if all of the productions (i.e., rules) are of the form

$$A \rightarrow xB$$

$$A \rightarrow x$$

where $A$ and $B$ are non-terminal symbols and $x$ is a terminal symbol from $\Sigma$. In other words, right-linear grammars expand sentences in a left-to-right fashion. Similarly, a grammar is said to be left linear if all of the productions are of the form

$$A \rightarrow Bx$$

$$A \rightarrow x$$

A regular grammar is either left linear or right linear.

### Context Free Grammars

Moving up what is known as the Chomsky hierarchy, a grammar $G$ is context free if all of the productions in the grammar have the following form.

$$A \rightarrow x$$

where $x$ is a string consisting of a combination of terminal and/or non-terminal symbols.

As suggested above, since we have "moved up" the Chomsky hierarchy, every regular grammar can also be shown to be context free. The opposite is not necessarily true. This is because context free grammars are not necessarily going to be linear.

As an interesting piece of "trivia," most modern programming languages are context free. The advantage to this is that parsing is simple. In addition, most constructs (if not all) needed for programming can be handled with context free grammars.

Where regular languages can be recognized by DFAs, the context free grammar requires a deterministic "pushdown" automaton. These automata are like DFAs except they also have an infinite-sized stack.

### The Chomsky Hierarchy

The Chomsky hierarchy relates the expressiveness of the various types of languages. Traditionally, the Chomsky hierarchy consists of four types of languages:

1. Type 0 languages: Also referred to as "unrestricted grammars," type 0 grammars generate all languages that can be recognized by a Turing machine. This type includes the set of recursively enumerable languages.

2. Type 1 languages: These correspond to the set of context sensitive languages and correspond to those languages that can be recognized by a Turing machine with a bounded tape.

3. Type 2 languages: These are the context free languages as defined previously.

4. Type 3 languages: These are the regular languages as defined previously.

In fact, there are more classes of languages that have been defined. The following diagram shows the relationship between these languages within the hierarchy.