**What is a grammar?**

**grammar** is the set of [structural](http://en.wikipedia.org/wiki/Structural) rules governing the composition of [clauses](http://en.wikipedia.org/wiki/Clause_(linguistics)), [phrases](http://en.wikipedia.org/wiki/Phrase), and [words](http://en.wikipedia.org/wiki/Words) in any given [natural language](http://en.wikipedia.org/wiki/Natural_language)

a **grammar** (when the context is not given, often called a **formal grammar** for clarity) is a set of [production rules](http://en.wikipedia.org/wiki/Production_(computer_science)) for [strings](http://en.wikipedia.org/wiki/String_(computer_science)) in a [formal language](http://en.wikipedia.org/wiki/Formal_language)

What is a language?

A language is the set of string generated by a grammar.

What is a regular grammar?

a **regular grammar** is a [formal grammar](http://en.wikipedia.org/wiki/Formal_grammar) that is right-regular or left-regular.

**A formal grammar is**

A formal grammar is a set of rules for rewriting strings, along with a "start symbol" from which rewriting starts.

What is a context free grammar?

**context-free grammar** (**CFG**) is a [formal grammar](http://en.wikipedia.org/wiki/Formal_grammar) in which every [production rule](http://en.wikipedia.org/wiki/Production_(computer_science)) is of the form

*V* → *w*

where *V* is a *single* [nonterminal](http://en.wikipedia.org/wiki/Nonterminal) symbol, and *w* is a string of [terminals](http://en.wikipedia.org/wiki/Terminal_and_nonterminal_symbols) and/or nonterminals (*w* can be empty). A formal grammar is considered "context free" when its production rules can be applied regardless of the context of a nonterminal. No matter which symbols surround it, the single nonterminal on the left hand side can always be replaced by the right hand side.

Be able to derive sentences from a grammar. What are leftmost and rightmost derivations? Be able to derive both.

**eftmost and Rightmost Derivations**

At any stage during a parse, when we have derived some sentential form (that is not yet a sentence) we will potentially have two choices to make:

1. which non-terminal in the sentential form to apply a production rule to
2. which production rule for that non-terminal to apply

Eg. in the above example, when we derived $E\; OP\; E$, we could then have applied a production rule to any of these three non-terminals, and would then have had to choose among all the production rules for either $E$ or $OP$.

The first decision here is relatively easy to solve: we will be reading the input string from left to right, so it is our own interest to derive the leftmost terminal of the resulting sentence as soon as possible. Thus, in a top-down parse we always choose the leftmost non-terminal in a sentential form to apply a production rule to - this is called a **leftmost derivation**.

If we were doing a bottom-up parse then the situation would be reversed, and we would want to do apply the production rules in reverse to the leftmost symbols; thus we are performing a **rightmost derivation** in reverse.

For example, a bottom-up rightmost derivation would look like:

Be able to build parse trees from a grammar.

What is an ambiguous grammar? Be able to demonstrate that grammar is ambiguous.

 an **ambiguous grammar** is a [context-free grammar](http://en.wikipedia.org/wiki/Context-free_grammar) for which there exists a [string](http://en.wikipedia.org/wiki/String_(computer_science)) that can have more than one [leftmost derivation](http://en.wikipedia.org/wiki/Leftmost_derivation), while an **unambiguous grammar** is a context-free grammar for which every valid string has a unique leftmost derivation. Many languages admit both ambiguous and unambiguous grammars, while some languages admit only ambiguous grammars. Any non-empty language admits an ambiguous grammar by taking an unambiguous grammar and introducing a duplicate rule or synonym (the only language without ambiguous grammars is the empty language)

### ****Topics****

**These are a comprehensive, but not complete list of topics.**

Chomsky Hierarchy.

In [computer science](http://en.wikipedia.org/wiki/Computer_science), an **ambiguous grammar** is a [context-free grammar](http://en.wikipedia.org/wiki/Context-free_grammar) for which there exists a [string](http://en.wikipedia.org/wiki/String_(computer_science)) that can have more than one [leftmost derivation](http://en.wikipedia.org/wiki/Leftmost_derivation), while an **unambiguous grammar** is a context-free grammar for which every valid string has a unique leftmost derivation. Many languages admit both ambiguous and unambiguous grammars, while some languages admit only ambiguous grammars. Any non-empty language admits an ambiguous grammar by taking an unambiguous grammar and introducing a duplicate rule or synonym (the only language without ambiguous grammars is the empty language).

What is a language?

What is a grammar?

Grammars define a language. A language is the set of strings generated by a grammar.

Languages and grammars - regular, context free

Language generators generate the strings of a language. Grammars are generators.