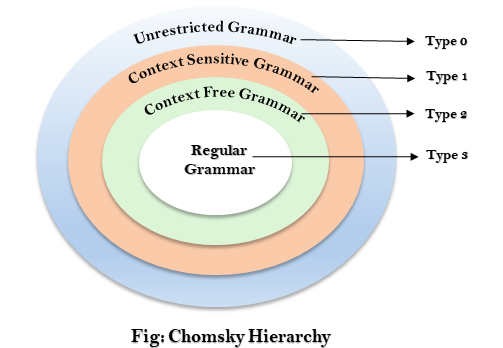
# **Chomsky Hierarchy**

Chomsky Hierarchy represents the class of languages that are accepted by the different machine. The category of language in Chomsky's Hierarchy is as given below:

1. Type 0 known as Unrestricted Grammar.
2. Type 1 known as Context Sensitive Grammar.
3. Type 2 known as Context Free Grammar.
4. Type 3 Regular Grammar.



This is a hierarchy. Therefore every language of type 3 is also of type 2, 1 and 0. Similarly, every language of type 2 is also of type 1 and type 0, etc.

### **Type 0 Grammar:**

Type 0 grammar is known as Unrestricted grammar. There is no restriction on the grammar rules of these types of languages. These languages can be efficiently modeled by Turing machines.

**For example:**

1. bAa → aa
2. S → s

### **Type 1 Grammar:**

Type 1 grammar is known as Context Sensitive Grammar. The context sensitive grammar is used to represent context sensitive language. The context sensitive grammar follows the following rules:

* The context sensitive grammar may have more than one symbol on the left hand side of their production rules.
* The number of symbols on the left-hand side must not exceed the number of symbols on the right-hand side.
* The rule of the form A → ε is not allowed unless A is a start symbol. It does not occur on the right-hand side of any rule.
* The Type 1 grammar should be Type 0. In type 1, Production is in the form of V → T

Where the count of symbol in V is less than or equal to T.

**For example:**

1. S → AT
2. T → xy
3. A → a

### **Type 2 Grammar:**

Type 2 Grammar is known as Context Free Grammar. Context free languages are the languages which can be represented by the context free grammar (CFG). Type 2 should be type 1. The production rule is of the form

1. A → α

Where A is any single non-terminal and is any combination of terminals and non-terminals.

**For example:**

1. A → aBb
2. A → b
3. B → a

### **Type 3 Grammar:**

Type 3 Grammar is known as Regular Grammar. Regular languages are those languages which can be described using regular expressions. These languages can be modeled by NFA or DFA.

Type 3 is most restricted form of grammar. The Type 3 grammar should be Type 2 and Type 1. Type 3 should be in the form of

1. V → T\*V / T\*

**For example:**

1. A → xy

# CHOMSKY CLASSIFICATION

In this chapter you will learn the four classes of formal languages,introduced by Noam Chomsky.

Most famous classification of grammars and languages ​​introduced by Noam Chomsky is divided into four classes:

* **Recursively enumerable grammars** –recognizable by a Turing machine
* **Context-sensitive grammars** –recognizable by the linear bounded automaton
* **Context-free grammars**- recognizable by the pushdown automaton
* **Regular grammars** –recognizable by the finite state automaton

**Interesting…**

**Noam Chomsky**,is an American linguist,philosopher,cognitive scientist and social activist. Chomsky is well known in the academic and scientific community as one of the fathers of modern linguistics and a major figure of analitic philosophy.

## 0 –***Recursively enumerable grammar***

**Type-0 grammars** (unrestricted grammars) include all formal grammars. They generate exactly all languages that can be recognized by a Turing machine. These languages are also known as the recursively enumerable languages. Note that this is different from the recursive languages which can be decided by an always-halting Turing machine.

Class 0 grammars are too general to describe the syntax of programming languages ​​and natural languages​​.

## 1 –***Context-sensitive grammars***

**Type-1 grammars** generate the context-sensitive languages. These grammars have rules of the form α A β → α γ β with A a nonterminal and α,β and γ strings of terminals and nonterminals. The strings α and β may be empty,but γ must be nonempty. The languages described by these grammars are exactly all languages that can be recognized by a linear bounded automaton.

Example:

AB → CDB

AB → CdEB

ABcd → abCDBcd

B → b

## 2 –Context-free grammars

**Type-2 grammars** generate the context-free languages. These are defined by rules of the form A → γ with A a nonterminal and γ a string of terminals and nonterminals. These languages are exactly all languages that can be recognized by a non-deterministic pushdown automaton. Context-free languages are the theoretical basis for the syntax of most programming languages.

Example:

A → aBc

## 3 –Regular grammars

**Type-3 grammars** generate the regular languages. Such a grammar restricts its rules to a single nonterminal on the left-hand side and a right-hand side consisting of a single terminal,possibly followed (or preceded,but not both in the same grammar) by a single nonterminal. The rule S → ε is also allowed here if S does not appear on the right side of any rule. These languages are exactly all languages that can be decided by a finite state automaton. Additionally,this family of formal languages can be obtained by regular expresions. Regular languages are commonly used to define search patterns and the lexical structure of programming languages.  
Example:

A → ε

A →  a

A →  abc

A →  B

A →  abcB

**Interesting…**

If languages L1 and L2 are regular,are also regular the following languages:

L1 ∪ L2

L1 ∩ L2

L1 \*  L2 ={xy:x ∈ L1 ∧ y ∈ L2}

L\* ={ε} ∪ L ∪ L2 ∪ …

## Questions

* What is the classification of grammar introduced by Chomsky?
* Which class of grammars corresponds to Turing machines?

If you do not know the answer to these questions,read again the above content.