planetmath.org

Math for the people, by the people.

Chomsky hierarchy

Canonical name	ChomskyHierarchy
Date of creation	2013-03-22 16:27:04
Last modified on	2013-03-22 16:27:04
Owner	CWoo (3771)
Last modified by	CWoo (3771)
Numerical id	13
Author	CWoo (3771)
Entry type	Definition
Classification	msc 68Q15
Classification	msc 03D55
Synonym	Chomsky-Schützenberger hierarchy
Synonym	Chomsky-Schutzenberger hierarchy
Synonym	unrestricted grammar
Defines	type-0 grammar
Defines	type-0 language

The *Chomsky hierarchy* or *Chomsky-Schützenberger hierarchy* is a way of classifying formal grammars into four types, with the lower numbered types being more general.

Recall that a formal grammar $G = (\Sigma, N, P, \sigma)$ consists of an alphabet Σ, an alphabet N of non-terminal symbols properly included in Σ, a non-empty finite set P of productions, and a symbol $\sigma \in N$ called the start symbol. The non-empty alphabet $T := \Sigma - N$ is the set of terminal symbols. Then G is called a

Type-0 grammar if there are no restrictions on the productions. Type-0 grammar is also known as an *unrestricted grammar*, or a *phrase-structure grammar*.

Type-1 grammar if the productions are of the form $uAv \to uWv$, where $u, v, W \in \Sigma^*$ with $W \neq \lambda$, and $A \in N$, or $\sigma \to \lambda$, provided that σ does not occur on the right hand side of any productions in P. As A is surrounded by words u, v, a type-1 grammar is also known as a context-sensitive grammar.

Type-2 grammar if the productions are of the form $A \to W$, where $A \in N$ and $W \in \Sigma^*$. Type-2 grammars are also called context-free grammars, because the left hand side of any productions are "free" of contexts.

Type-3 grammar if the productions are of the form $A \to u$ or $A \to uB$, where $A, B \in N$ and $u \in T^*$. Owing to the fact that languages generated by type-3 grammars can be represented by regular expressions, type-3 grammars are also known as regular grammars.

It is clear that every type-i grammar is type-0, and every type-3 grammar is type-2. A type-2 grammar is not necessarily type-1, because it may contain both $\sigma \to \lambda$ and $A \to W$, where λ occurs in W. Nevertheless, the relevance of the hierarchy has more to do with the languages generated by the grammars. Call a formal language a type-i language if it is generated by a type-i grammar, and denote $\mathscr{L}_i$ the family of type-i languages. Then it can be shown that

$$\mathscr{L}_3 \subset \mathscr{L}_2 \subset \mathscr{L}_1 \subset \mathscr{L}_0$$

where each inclusion is strict.

Below is a table summarizing the four types of grammars, the languages they generate, and the equivalent computational devices accepting the languages.

grammar	language family	abbreviation	automaton
type-0	recursively enumerable	$\mathscr{L}_0$ or $\mathscr{E}$	turing machine
type-1	context-sensitive	$\mathscr{L}_1$ or $\mathscr{S}$	linear bounded automaton
type-2	context-free	$\mathscr{L}_2$ or $\mathscr{F}$	pushdown automaton
type-3	regular	$\mathscr{L}_3$ or $\mathscr{R}$	finite automaton