The Chomsky Hierarchy

There are other types of grammars.

Regular Grammars

We noted that every regular language has a CFG. In fact, a regular language (without ε) can be generated by a special form:

> A ***regular grammar*** is one where every production is of the form $A \to \mathtt{b}C$ or $A \to \mathtt{a}$ (where $\mathtt{a}$ and $\mathtt{b}$ are some terminals and C some variable).

Regular Languages have Regular Grammars

Theorem. *Every regular language is generated by a regular grammar.*

PROOF. Idea: produce a grammar such that a derivation mimics the operation of the automaton.

Every stage of derivation will have a single variable that is the state of the FA.

Construction of Regular Grammar

Start with DFA for the language. Introduce one variable for each state.

For each transition, add a production:
if $\delta(A, \mathtt{x}) = B$, then add production $A \to \mathtt{x}B$.

For each transition ending at an accept state, add a further production: if B is accept state in above, then also add production $A \to \mathtt{x}$.

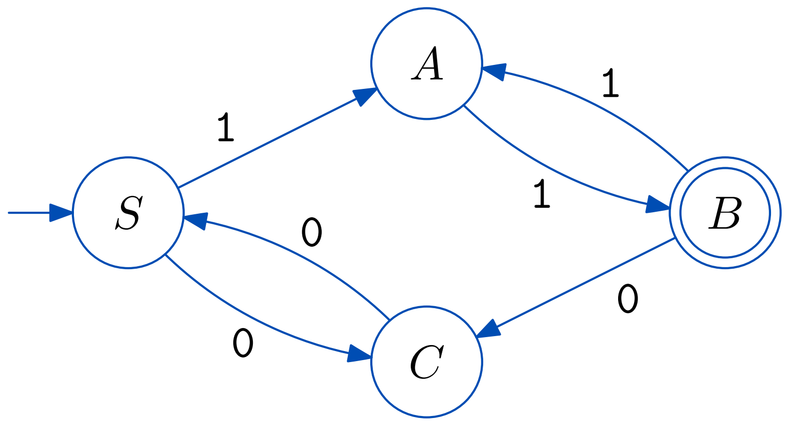
The start variable is the start state.

Example

Consider RE $(11 + 00)^*11$.

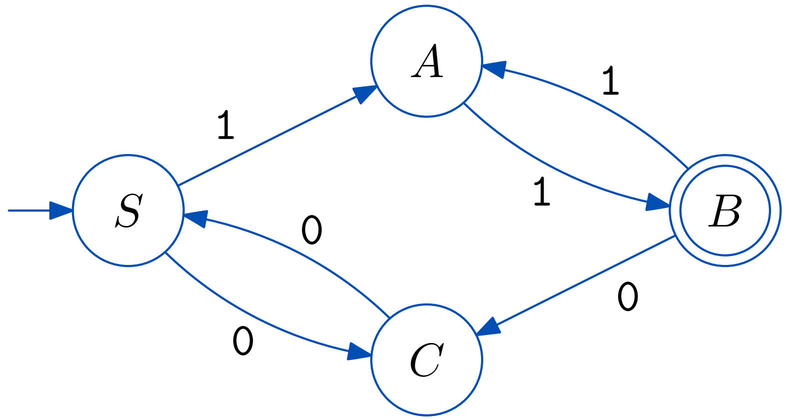
Example

Consider RE $(11+00)^*11$. Here is NFA:



Example

Consider RE $(11+00)^*11$. Here is NFA:



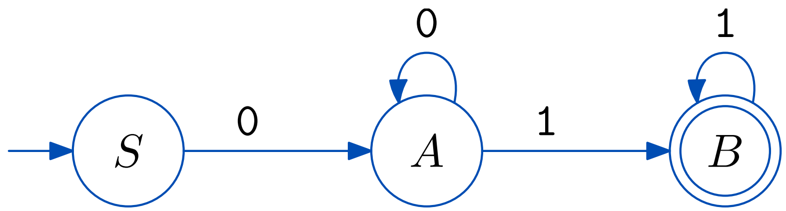
This yields the following regular grammar:

$$
\begin{aligned}
S &\to 0C \mid 1A \\
A &\to 1B \mid 1 \\
B &\to 0C \mid 1A \\
C &\to 0S
\end{aligned}
$$

Practice

Draw an FA, and from there write down a regular grammar, for the language given by the RE `00`*`11`*.

Solution to Practice



$$S \to \texttt{0}A$$

$$A \to \texttt{0}A \mid \texttt{1}B \mid \texttt{1}$$

$$B \to \texttt{1}B \mid \texttt{1}$$

Unrestricted & Context-Sensitive Grammars

In ***unrestricted grammars***, productions have form $u \to v$ where u and v are any strings of terminals and/or variables.

In ***context-sensitive grammars***, productions have form $xAz \to xyz$ where x, y and z are strings of terminals and/or variables, and A is a variable.

Context-Sensitive Example

A context-sensitive grammar for $\texttt{0}^n\texttt{1}^n\texttt{2}^n$ is not obvious!

$$S \rightarrow \texttt{0}BS\texttt{2} \mid \texttt{012}$$
$$B\texttt{0} \rightarrow \texttt{0}B$$
$$B\texttt{1} \rightarrow \texttt{11}$$

(Try to derive $\texttt{000111222}$.)

The Chomsky Hierarchy

Chomsky introduced the hierarchy of grammars in his study of natural languages.

0. Unrestricted grammars.
1. Context-sensitive grammars.
2. Context-free grammars.
3. Regular grammars.

We have seen that regular grammars are accepted by FAs, and that CFGs are accepted by PDAs. We will see later machines for the other two types.

Simplifying CFGs: Usable & Nullable Variables

A variable is ***usable*** if it generates some string of terminals. A variable is ***nullable*** if it generates the empty string.

Example: In the following, A and B are usable but only B is nullable.

$$
\begin{aligned}
A &\to \mathtt{0}A \mid \mathtt{1}B \mid \mathtt{2}C \\
B &\to \mathtt{0}B \mid \varepsilon \\
C &\to \mathtt{1}C
\end{aligned}
$$

Algorithm for Nullable Variables

Identification of nullable variables. *Initialize all variables as not-nullable.*
Repeat:
go through all productions, and if any has RHS empty or all entries nullable,
then mark the LHS variable as nullable;
Until no increase in the set of nullable variables.

A similar procedure can be used to determine the usable variables.

Summary

A regular grammar is one where every production has the form $A \to \texttt{b}C$ or $A \to \texttt{a}$. The Chomsky hierarchy also includes context-sensitive grammars and unrestricted grammars.