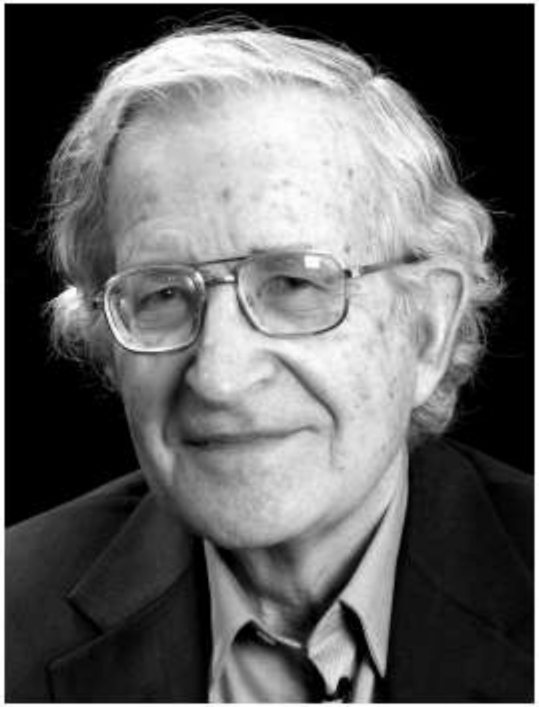
Context Free Languages

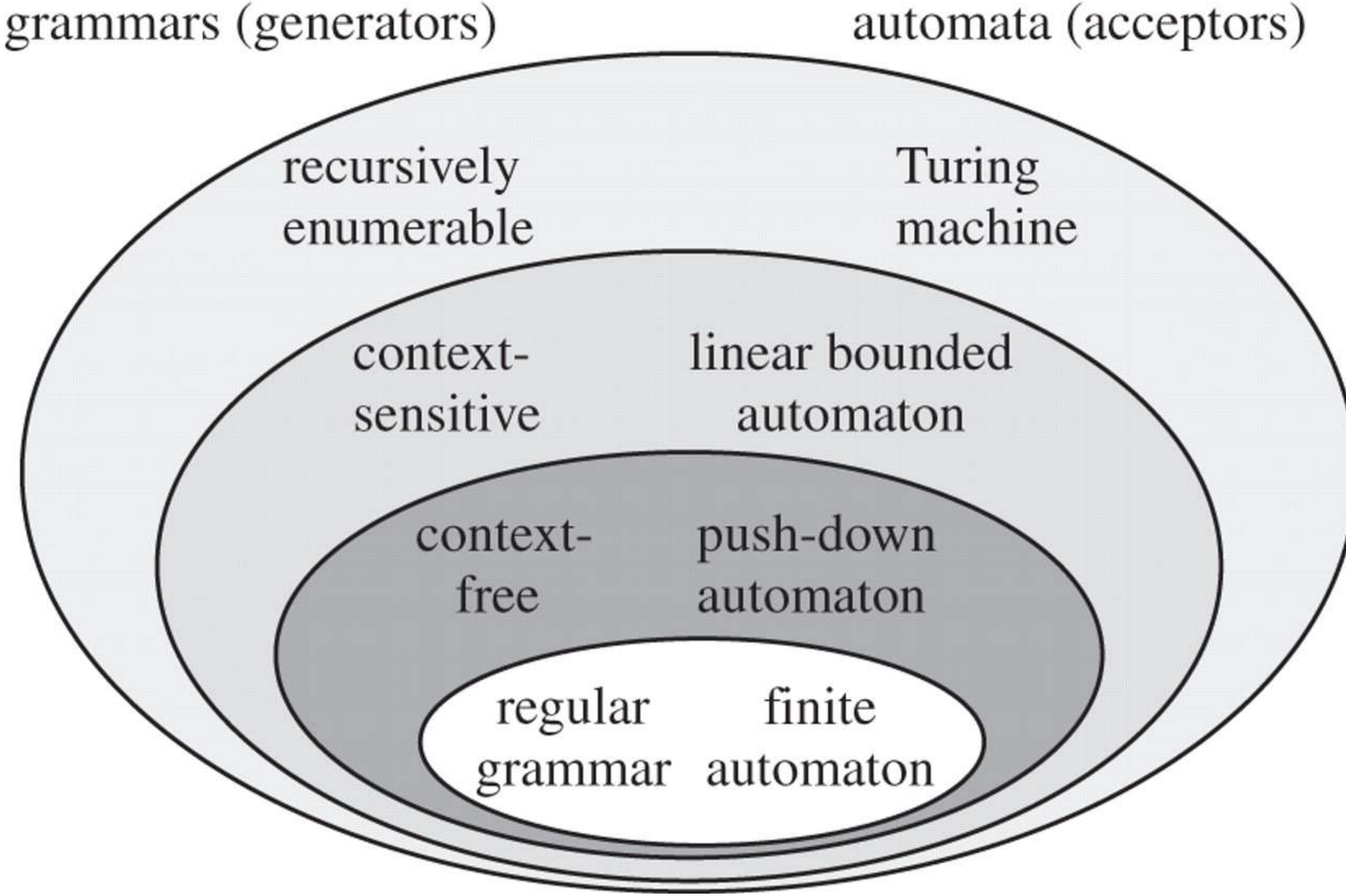
Context-Free Grammars



Noam Chomsky
(linguist, philosopher, logician, and activist)

In the formal languages of computer science and linguistics, the **Chomsky hierarchy** is a **hierarchy** of classes of formal grammars. This **hierarchy** of grammars was described by Noam **Chomsky** in 1956.

Chomsky Hierarchy



The Hierarchy

Class	Grammars	Languages	Automaton
Type-0	Unrestricted	Recursive Enumerable	Turing Machine
Type-1	Context Sensitive	Context Sensitive	Linear-Bound
Type-2	Context Free	Context Free	Pushdown
Type-3	Regular	Regular	Finite

How production rules look like

Type	Grammar	Production rules
Type 0	unrestricted	$\alpha \to \beta$
Type 1	context-sensitive	$\alpha A \beta \to \alpha \gamma \beta$
Type 2	**context-free**	$A \to \gamma$
Type 3	regular	$A \to aB$ or $A \to Ba$

A grammar generates sentences (strings) in a language

Examples

Consider the grammar

$$
\begin{align}
S &\rightarrow AB \tag{1}\\
A &\rightarrow C \tag{2}\\
CB &\rightarrow Cb \tag{3}\\
C &\rightarrow a \tag{4}
\end{align}
$$

where $\{a, b\}$ are terminals, and $\{S, A, B, C\}$ are non-terminals.

Examples

Consider the grammar

$$
\begin{align}
S &\rightarrow AB \tag{1}\\
A &\rightarrow C \tag{2}\\
CB &\rightarrow Cb \tag{3}\\
C &\rightarrow a \tag{4}
\end{align}
$$

where $\{a, b\}$ are terminals, and $\{S, A, B, C\}$ are non-terminals.
We can derive the phrase "ab" from this grammar in the following way:

$$
\begin{aligned}
S &\rightarrow AB, \text{ from (1)}\\
&\rightarrow CB, \text{ from (2)}\\
&\rightarrow Cb, \text{ from (3)}\\
&\rightarrow ab, \text{ from (4)}
\end{aligned}
$$

Examples

Consider the grammar

$$\begin{aligned} S &\rightarrow \mathit{NounPhrase} \quad \mathit{VerbPhrase} && (5) \\ \mathit{NounPhrase} &\rightarrow \mathit{SingularNoun} && (6) \\ \mathit{SingularNoun} \quad \mathit{VerbPhrase} &\rightarrow \mathit{SingularNoun} \quad \mathit{comes} && (7) \\ \mathit{SingularNoun} &\rightarrow \mathit{John} && (8) \end{aligned}$$

We can derive the phrase "John comes" from this grammar in the following way:

$$\begin{aligned} S &\rightarrow \mathit{NounPhrase} \quad \mathit{VerbPhrase}, \text{ from (1)} \\ &\rightarrow \mathit{SingularNoun} \quad \mathit{VerbPhrase}, \text{ from (2)} \\ &\rightarrow \mathit{SingularNoun} \quad \mathit{comes}, \text{ from (3)} \\ &\rightarrow \mathit{John} \quad \mathit{comes}, \text{ from (4)} \end{aligned}$$

Type	Grammar	Production rules
Type 0	unrestricted	$\alpha \to \beta$
Type 1	context-sensitive	$\alpha A \beta \to \alpha \gamma \beta$
Type 2	**context-free**	$A \to \gamma$
Type 3	regular	$A \to aB$ or $A \to Ba$

Definition (Context-Free Grammar)

A context-free grammar is a tuple $G = (V, T, P, S)$ where

- V is a finite set of variables (nonterminals, nonterminals vocabulary);
- T is a finite set of terminals (letters);
- $P \subseteq V \times (V \cup T)^*$ is a finite set of rewriting rules called productions,
  - We write $A \rightarrow \beta$ if $(A, \beta) \in P$;
- $S \in V$ is a distinguished start or "sentence" symbol.

Definition (Context-Free Grammar)

A context-free grammar is a tuple $G = (V, T, P, S)$ where

- V is a finite set of variables (nonterminals, nonterminals vocabulary);
- T is a finite set of terminals (letters);
- $P \subseteq V \times (V \cup T)^*$ is a finite set of rewriting rules called productions,
  - We write $A \to \beta$ if $(A, \beta) \in P$;
- $S \in V$ is a distinguished start or "sentence" symbol.

Example: $G_{0^n1^n} = (V, T, P, S)$ where

- $V = \{S\}$;
- $T = \{0, 1\}$;
- P is defined as

$$
\begin{array}{lcl}
S & \to & \varepsilon \\
S & \to & 0S1
\end{array}
$$

- $S = S$.

Palindromes

$G_{pal} = (\{P\}, \{0,1\}, A, P)$

$$
\begin{array}{llll}
1. & P & \rightarrow & \epsilon \\
2. & P & \rightarrow & 0 \\
3. & P & \rightarrow & 1 \\
4. & P & \rightarrow & 0P0 \\
5. & P & \rightarrow & 1P1
\end{array}
$$

A context-free grammar for palindromes

Derivation:

- Let $G = (V, T, P, S)$ be a context-free grammar.
- Let $\alpha A \beta$ be a string in $(V \cup T)^* V (V \cup T)^*$
- We say that $\alpha A \beta$ yields the string $\alpha \gamma \beta$, and we write $\alpha A \beta \Rightarrow \alpha \gamma \beta$ if

$$A \to \gamma \text{ is a production rule in } G.$$

- For strings $\alpha, \beta \in (V \cup T)^*$, we say that α derives β and we write $\alpha \overset{*}{\Rightarrow} \beta$ if there is a sequence $\alpha_1, \alpha_2, \ldots, \alpha_n \in (V \cup T)^*$ s.t.

$$\alpha \Rightarrow \alpha_1 \Rightarrow \alpha_2 \cdots \alpha_n \Rightarrow \beta.$$

$\Rightarrow$ is also called direct derivation.
$\overset{i}{\Rightarrow}$ is to mean that the i th production is used in the direct derivation.
$\overset{*}{\Rightarrow}$ is reflexive and transitive closure of $\Rightarrow$

1.	E	$\rightarrow$	I
2.	E	$\rightarrow$	$E + E$
3.	E	$\rightarrow$	$E * E$
4.	E	$\rightarrow$	(E)
5.	I	$\rightarrow$	a
6.	I	$\rightarrow$	b
7.	I	$\rightarrow$	Ia
8.	I	$\rightarrow$	Ib
9.	I	$\rightarrow$	$I0$
10.	I	$\rightarrow$	$I1$

A context-free grammar for simple expressions

$$
\begin{array}{llll}
1. & E & \rightarrow & I \\
2. & E & \rightarrow & E + E \\
3. & E & \rightarrow & E * E \\
4. & E & \rightarrow & (E) \\
\\
5. & I & \rightarrow & a \\
6. & I & \rightarrow & b \\
7. & I & \rightarrow & Ia \\
8. & I & \rightarrow & Ib \\
9. & I & \rightarrow & I0 \\
10. & I & \rightarrow & I1
\end{array}
$$

A context-free grammar for simple expressions

T is the set of symbols $\{+, *, (,), a, b, 0, 1\}$ and P is the set of productions

$$
\begin{array}{llll}
1. & E & \rightarrow & I \\
2. & E & \rightarrow & E + E \\
3. & E & \rightarrow & E * E \\
4. & E & \rightarrow & (E) \\
\\
5. & I & \rightarrow & a \\
6. & I & \rightarrow & b \\
7. & I & \rightarrow & Ia \\
8. & I & \rightarrow & Ib \\
9. & I & \rightarrow & I0 \\
10. & I & \rightarrow & I1
\end{array}
$$

A context-free grammar for simple expressions

T is the set of symbols $\{+, *, (,), a, b, 0, 1\}$ and P is the set of productions

- Can you find how the following is true.

$$E \overset{*}{\Rightarrow} (a1 + b0 * a1)$$

Compact Notation for Productions

It is convenient to think of a production as "belonging" to the variable of its head. We shall often use remarks like "the productions for A" or "A-productions" to refer to the productions whose head is variable A. We may write the productions for a grammar by listing each variable once, and then listing all the bodies of the productions for that variable, separated by vertical bars. That is, the productions $A \to \alpha_1,\ A \to \alpha_2, \ldots, A \to \alpha_n$ can be replaced by the notation $A \to \alpha_1|\alpha_2|\cdots|\alpha_n$. For instance, the grammar for palindromes from Fig. 5.1 can be written as $P \to \epsilon \mid 0 \mid 1 \mid 0P0 \mid 1P1$.

CFL Definition

The language $L(G)$ accepted by a context-free grammar $G = (V, T, P, S)$ is the set

$$L(G) = \{w \in T^* \ : \ S \overset{*}{\Rightarrow} w\}.$$

Leftmost and Rightmost Derivations

- Derivations are not unique.
- So, to bring uniqueness, we define two special type of derivations, viz., leftmost and rightmost.

$$
\begin{array}{llll}
1. & E & \to & I \\
2. & E & \to & E + E \\
3. & E & \to & E * E \\
4. & E & \to & (E) \\
 & & & \\
5. & I & \to & a \\
6. & I & \to & b \\
7. & I & \to & Ia \\
8. & I & \to & Ib \\
9. & I & \to & I0 \\
10. & I & \to & I1
\end{array}
$$

A context-free grammar for simple expressions

$$E \underset{lm}{\Rightarrow} E * E \underset{lm}{\Rightarrow} I * E \underset{lm}{\Rightarrow} a * E \underset{lm}{\Rightarrow}$$

$$a * (E) \underset{lm}{\Rightarrow} a * (E + E) \underset{lm}{\Rightarrow} a * (I + E) \underset{lm}{\Rightarrow} a * (a + E) \underset{lm}{\Rightarrow}$$

$$a * (a + I) \underset{lm}{\Rightarrow} a * (a + I0) \underset{lm}{\Rightarrow} a * (a + I00) \underset{lm}{\Rightarrow} a * (a + b00)$$

We can also summarize the leftmost derivation by saying $E \overset{*}{\underset{lm}{\Rightarrow}} a * (a + b00)$, or express several steps of the derivation by expressions such as $E * E \overset{*}{\underset{lm}{\Rightarrow}} a * (E)$.

$$
\begin{array}{llll}
1. & E & \rightarrow & I \\
2. & E & \rightarrow & E + E \\
3. & E & \rightarrow & E * E \\
4. & E & \rightarrow & (E) \\
\\
5. & I & \rightarrow & a \\
6. & I & \rightarrow & b \\
7. & I & \rightarrow & Ia \\
8. & I & \rightarrow & Ib \\
9. & I & \rightarrow & I0 \\
10. & I & \rightarrow & I1
\end{array}
$$

A context-free grammar for simple expressions

$$
E \underset{rm}{\Rightarrow} E * E \underset{rm}{\Rightarrow} E * (E) \underset{rm}{\Rightarrow} E * (E + E) \underset{rm}{\Rightarrow}
$$

$$
E * (E + I) \underset{rm}{\Rightarrow} E * (E + I0) \underset{rm}{\Rightarrow} E * (E + I00) \underset{rm}{\Rightarrow} E * (E + b00) \underset{rm}{\Rightarrow}
$$

$$
E * (I + b00) \underset{rm}{\Rightarrow} E * (a + b00) \underset{rm}{\Rightarrow} I * (a + b00) \underset{rm}{\Rightarrow} a * (a + b00)
$$

This derivation allows us to conclude $E \overset{*}{\underset{rm}{\Rightarrow}} a * (a + b00)$. $\square$

Exercise

Consider the following grammar:

$$\begin{aligned} S &\rightarrow AS \mid \varepsilon. \\ A &\rightarrow aa \mid ab \mid ba \mid bb \end{aligned}$$

Give leftmost and rightmost derivations of the string *aabbba*.