1 Chomsky Hierarchy

Grammars for each task

Different types of rules, allow one to describe different aspects of natural language

These grammars form a hierarchy

Grammars in General

All grammars we consider will be of the form \( G = (V, \Sigma, R, S) \)

- \( V \) is a finite set of variables
- \( \Sigma \) is a finite set of terminals
- \( R \) is a finite set of rules
- \( S \) is the start symbol

The different grammars will be determined by the form of the rules in \( R \).

1.1 Regular Languages

Type 3 Grammars

The rules in a type 3 grammar are of the form

\[ A \rightarrow aB \quad \text{or} \quad A \rightarrow a \]

where \( A, B \in V \) and \( a \in \Sigma \cup \{\epsilon\} \).

We say \( \alpha A\beta \Rightarrow_G \alpha\gamma\beta \) iff \( A \rightarrow \gamma \in R \).

\( L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^*_G w \} \)
1.1.1 Type 3 Grammars and Regularity

**Proposition 1.** If $G$ is Type 3 grammar then $L(G)$ is regular. Conversely, if $L$ is regular then there is a Type 3 grammar $G$ such that $L = L(G)$.

**Proof.** Let $G = (V, \Sigma, R, S)$ be a type 3 grammar. Consider the NFA $M = (Q, \Sigma, \delta, q_0, F)$ where

- $Q = V \cup \{q_F\}$, where $q_F \notin V$
- $q_0 = S$
- $F = \{q_F\}$
- $\delta(A, a) = \{B \mid A \rightarrow aB \in R\} \cup \{q_F \mid A \rightarrow a \in R\}$ for $A \in V$. And $\delta(q_F, a) = \emptyset$ for all $a$.

$L(M) = L(G)$ as $\forall A \in V, \forall w \in \Sigma^*, A \Rightarrow_G w$ iff $A \xrightarrow{w,M} q_F$.

Conversely, let $M = (Q, \Sigma, \delta, q_0, F)$ be a NFA recognizing $L$. Consider $G = (V, \Sigma, R, S)$ where

- $V = Q$
- $S = q_0$
- $q_1 \rightarrow aq_2 \in R$ iff $q_2 \in \delta(q_1, a)$ and $q \rightarrow \epsilon \in R$ iff $q \in F$.

We can show, for any $q, q' \in Q$ and $w \in \Sigma^*$, $q \xrightarrow{w,M} q'$ iff $q \Rightarrow_G wq'$. Thus, $L(M) = L(G)$. \hfill \Box

1.2 Context-free Languages

**Type 2 Grammars**

The rules in a type 2 grammar are of the form

$$A \rightarrow \beta$$

where $A \in V$ and $\beta \in (\Sigma \cup V)^*$. We say $\alpha A \beta \Rightarrow_G \alpha \gamma \beta$ iff $A \rightarrow \gamma \in R$. $L(G) = \{w \in \Sigma^* \mid S \Rightarrow_G w\}$

By definition, Type 2 grammars describe exactly the class of context-free languages.
1.3 Beyond Context-Free Languages

1.3.1 Type 0 Grammars

Type 0 Grammars

The rules in a type 0 grammar are of the form

$$\alpha \rightarrow \beta$$

where $$\alpha, \beta \in (\Sigma \cup V)^*$$.

We say $$\gamma_1 \alpha \gamma_2 \Rightarrow_G \gamma_1 \beta \gamma_2$$ iff $$\alpha \rightarrow \beta \in R$$. 

$$L(G) = \{w \in \Sigma^* | S \Rightarrow_G^* w\}$$

Example of Type 0 Grammar

Example 2. Consider the grammar $$G$$ with $$\Sigma = \{a\}$$ with

$$S \rightarrow \$Ca\# \mid a \mid \epsilon \quad Ca \rightarrow aaC \quad \$D \rightarrow \$C$$

$$C\# \rightarrow D\# \mid E \quad aD \rightarrow Da \quad aE \rightarrow Ea$$

$$\$E \rightarrow \epsilon$$

The following are derivations in this grammar

$$S \Rightarrow \$Ca\# \Rightarrow \$aaC\# \Rightarrow \$aaE \Rightarrow \$aEa \Rightarrow \$Eaa \Rightarrow aa$$

$$S \Rightarrow \$Ca\# \Rightarrow \$aaC\# \Rightarrow \$aaD\# \Rightarrow \$Da\# \Rightarrow \$Da\# \Rightarrow \$Caa\#$$

$$\Rightarrow \$aaCa\# \Rightarrow \$aaaaC\# \Rightarrow \$aaaaE \Rightarrow \$aaaEa \Rightarrow \$aaEaa$$

$$\Rightarrow \$aEaaa \Rightarrow \$Eaaaa \Rightarrow aaaa$$

$$L(G) = \{a^i \mid i \text{ is a power of 2}\}$$

Expressive Power of Type 0 Grammars

Recall that any decision problem can be thought of as a formal language $$L$$, where $$x \in L$$ iff the answer on input $$x$$ is “yes”.

Proposition 3. A decision problem $$L$$ can be “solved on computers” iff $$L$$ can be described by a Type 0 grammar.

Proof. Need to develop some theory, that we will see in the next few weeks.
1.3.2 Type 1 Grammars

Type 1 Grammars

The rules in a type 1 grammar are of the form

\[ \alpha \rightarrow \beta \]

where \( \alpha, \beta \in (\Sigma \cup V)^* \) and \( |\alpha| \leq |\beta| \).

We say \( \gamma_1 \alpha \gamma_2 \Rightarrow G \gamma_1 \beta \gamma_2 \) iff \( \alpha \rightarrow \beta \in R \). \( L(G) = \{ w \in \Sigma^* \mid S \Rightarrow G w \} \)

Normal Form for Type 1 Grammars

We can define a normal form for Type 1 grammars where all rules are of the form

\[ \alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2 \]

Thus, the rules in Type 1, can be seen as rules of a CFG where a variable \( A \) is replaced by a string \( \beta \) in one step, with the only difference being that rule can be applied only in the context \( \alpha_1 \square \alpha_2 \).

Thus, languages described by Type 1 grammars are called context-sensitive languages.

1.3.3 Hierarchy

Chomsky Hierarchy

**Theorem 4.** Type 0, Type 1, Type 2, and Type 3 grammars define a strict hierarchy of formal languages.

*Proof.* Clearly a Type 3 grammar is a special Type 2 grammar, a Type 2 grammar is a special Type 1 grammar, and a Type 1 grammar is special Type 0 grammar.

Moreover, there is a language that has a Type 2 grammar but no Type 3 grammar (\( L = \{ 0^n1^n \mid n \geq 0 \} \)), a language that has a Type 1 grammar but no Type 2 grammar (\( L = \{ a^n b^n c^n \mid n \geq 0 \} \)), and a language with a Type 0 grammar but no Type 1 grammar.

Overview of Languages