Lecture #23: Chomsky Hierarchy IV – Normal Forms and Parsing

Lecture Presentation

Algorithms for Right-Regular Grammars

Consider a right-regular grammar $G = (V, \Sigma, \Pi, S)$ where $V = \{S, T, U, W\}$. $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, and Π includes the following rules.

$$
\begin{array}{lll}
S \to \mathtt{a}S & T \to \mathtt{a}U & W \to \mathtt{b}W \\
S \to \mathtt{b}S & U \to \mathtt{b}W & W \to \lambda \\
S \to \mathtt{a}T & W \to \mathtt{a}W &
\end{array}
$$

Suppose we wish to design an algorithm that solves a ***parsing*** problem for this grammar:

- The algorithms's input is a string $\omega \in \Sigma^\star$.
- If $\omega \in L(G)$ then the algorithm's output should be a parse tree, corresponding to a derivation of ω from S.
- If $\omega \notin L(G)$ then the algorithm's output should be `null`.

Organization of Algorithm

Algorithm

Proving Correctness of the Algorithm

Chomsky Normal Form

Next consider a context-free grammar $G = (V, \Sigma, \Pi, S)$ such that $V = \{S, T\}$, $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, and Π includes the following rules.

$$\begin{array}{ll} S \to \mathtt{a}S\mathtt{a} & \quad T \to \mathtt{b}T \\ S \to T & \quad T \to \lambda. \end{array}$$

Suppose we wish to produce a context-free grammar $\widehat{G} = (\widehat{V}, \Sigma, \widehat{\Pi}, \widehat{S})$ in ***Chomsky normal form*** such that $L(\widehat{G}) = L(G)$.

Introducing a New Start Variable

Eliminating Long Rules

Eliminating λ-Rules

Eliminating Unit Rules

Rules with Input Symbols

The Grammar Obtained

Algorithms for Grammars in Chomsky Normal Form

Consider the context-free grammar $\widehat{G} = (\widehat{V}, \Sigma, \widehat{\Pi}, \widehat{S})$ in Chomsky normal form that has now been obtained. Suppose we wish to design an algorithm that solves a ***parsing*** problem for this grammar:

- The algorithms's input is a string $\omega \in \Sigma^\star$.
- If $\omega \in L(\widehat{G})$ then the algorithm's output should be a parse tree, corresponding to a derivation of ω from $\widehat{S}$.
- If $\omega \notin L(G)$ then the algorithm's output should be `null`.

Organization of Algorithm

Algorithm

Proving Correctness of the Algorithm

Recovering a Parse Tree for the Original Grammar

Suppose, now, that $\omega \in L(\widehat{G}) = L(G)$, so that an application of the previous algorithm would produce a parse tree, showing a derivation of ω from $\widehat{S}$, for the grammar $\widehat{G}$.

Question: How could this be used to a produce a derivation of ω from the original start variable S, in the original context-free grammar G?