1 Chomsky Normal Form

Normal Forms for Grammars

It is typically easier to work with a context free language if given a CFG in a normal form.

Normal Forms

A grammar is in a normal form if its production rules have a special structure:

- **Chomsky Normal Form**: Productions are of the form $A \rightarrow BC$ or $A \rightarrow a$, where $A, B, C$ are variables and $a$ is a terminal symbol.

- **Greibach Normal Form**: Productions are of the form $A \rightarrow a\alpha$, where $\alpha \in V^*$ and $A \in V$.

If $\epsilon$ is in the language, we allow the rule $S \rightarrow \epsilon$. We will require that $S$ does not appear on the right hand side of any rules.

We will restrict our discussion to Chomsky Normal Form.

Main Result

**Proposition 1.** For any non-empty context-free language $L$, there is a grammar $G$, such that $L(G) = L$ and each rule in $G$ is of the form

1. $A \rightarrow a$ where $a \in \Sigma$, or
2. $A \rightarrow BC$ where neither $B$ nor $C$ is the start symbol, or
3. $S \rightarrow \epsilon$ where $S$ is the start symbol (iff $\epsilon \in L$)

Furthermore, $G$ has no useless symbols.

Outline of Normalization

Given $G = (V, \Sigma, S, P)$, convert to CNF

- Let $G' = (V', \Sigma, S, P')$ be the grammar obtained after eliminating $\epsilon$-productions, unit productions, and useless symbols from $G$.

- If $A \rightarrow x$ is a rule of $G'$, where $|x| = 0$, then $A$ must be $S$ (because $G'$ has no other $\epsilon$-productions). If $A \rightarrow x$ is a rule of $G'$, where $|x| = 1$, then $x \in \Sigma$ (because $G'$ has no unit productions). In either case $A \rightarrow x$ is in a valid form.

- All remaining productions are of form $A \rightarrow X_1X_2\cdots X_n$ where $X_i \in V' \cup \Sigma$, $n \geq 2$ (and $S$ does not occur in the RHS). We will put these rules in the right form by applying the following two transformations:
  1. Make the RHS consist only of variables
  2. Make the RHS be of length 2.
Make the RHS consist only of variables

Let \( A \rightarrow X_1 X_2 \cdots X_n \), with \( X_i \) being either a variable or a terminal. We want rules where all the \( X_i \) are variables.

Example 2. Consider \( A \rightarrow BbCdefG \). How do you remove the terminals?

For each \( a, b, c \ldots \in \Sigma \) add variables \( X_a, X_b, X_c, \ldots \) with productions \( X_a \rightarrow a, X_b \rightarrow b, \ldots \). Then replace the production \( A \rightarrow BbCdefG \) by \( A \rightarrow BX_aCX_bX_dX_fG \).

For every \( a \in \Sigma \)

1. Add a new variable \( X_a \)
2. In every rule, if \( a \) occurs in the RHS, replace it by \( X_a \)
3. Add a new rule \( X_a \rightarrow a \)

Make the RHS be of length 2

- Now all productions are of the form \( A \rightarrow a \) or \( A \rightarrow B_1 B_2 \cdots B_n \), where \( n \geq 2 \) and each \( B_i \) is a variable.
- How do you eliminate rules of the form \( A \rightarrow B_1 B_2 \cdots B_n \) where \( n > 2 \)?
- Replace the rule by the following set of rules

\[
\begin{align*}
A & \rightarrow B_1 B_{(2,n)} \\
B_{(2,n)} & \rightarrow B_2 B_{(3,n)} \\
B_{(3,n)} & \rightarrow B_3 B_{(4,n)} \\
& \vdots \\
B_{(n-1,n)} & \rightarrow B_{n-1} B_n
\end{align*}
\]

where \( B_{(i,n)} \) are “new” variables.

An Example

Example 3. Convert: \( S \rightarrow aA|bB|b, A \rightarrow Baa|ba, B \rightarrow bAA|ab, \) into Chomsky Normal Form.

1. Eliminate \( \epsilon \)-productions, unit productions, and useless symbols. This grammar is already in the right form.
2. Remove terminals from the RHS of long rules. New grammar is: \( X_a \rightarrow a, X_b \rightarrow b, S \rightarrow X_a A|X_b B|b, A \rightarrow BX_a X_a|X_b X_a, \) and \( B \rightarrow X_b A A X_b|X_a X_b \)
3. Reduce the RHS of rules to be of length at most two. New grammar replaces \( A \rightarrow BX_a X_a \) by rules \( A \rightarrow BX_{aa}, X_{aa} \rightarrow X_a X_a, \) and \( B \rightarrow X_b A A X_b \) by rules \( B \rightarrow X_b X_{A A b}, X_{A A b} \rightarrow A X_{A b}, X_{A b} \rightarrow A X_b \)