
2. The main message of this lecture:

Not all context free grammars are really needed for generating context free languages. In fact, some proper subclasses of CFG with rather simple productions suffices to generate all of CFL’s. Such simplified though still general CFG’s are called normal forms.

**Definition 19.1.** Chomsky Normal Form (CNF) is a CFG whose productions are of the form $A \rightarrow BC$ or $A \rightarrow a$, where $A, B, C$ are nonterminal letters, $a$ is a terminal letter.

**Example 19.2.** Consider CFG

$$S \rightarrow aSb, \quad S \rightarrow \epsilon$$

that generates the language $L = \{a^n b^n \mid n \geq 0\}$. Let us try to find its Chomsky Normal Form, if any. First of all, we have to exclude $\epsilon$ since no CNF can generate the null string (why?). So, we will try to find a CNF that generates the language $\{a^n b^n \mid n > 0\}$.

1. Getting rid of $\epsilon$ is easy.

$$S \rightarrow aSb, \quad S \rightarrow ab$$

Is this method scalable? It surely is! Whenever a CFG has productions $A \rightarrow \alpha B \gamma$ and $B \rightarrow \epsilon$ add $A \rightarrow \alpha \gamma$ and drop $B \rightarrow \epsilon$. In particular, if there are no $A \rightarrow \alpha B \gamma$’s then just drop $B \rightarrow \epsilon$, we are not going to use it anyway!

2. Replacing terminal letters in all nontrivial right-hand sides of productions by fresh nonterminal ones.

$$S \rightarrow ASB, \quad S \rightarrow AB, \quad A \rightarrow a, \quad B \rightarrow b$$

3. Shortening right-hand sides of productions (new nonterminal letters are needed!).

$$S \rightarrow AC, \quad C \rightarrow SB, \quad S \rightarrow AB, \quad A \rightarrow a, \quad B \rightarrow b$$

Done!

**Theorem 19.3.** For any CFG $G$ there is a CNF $G'$ such that $L(G') = L(G) - \{\epsilon\}$.

**Proof.** Get rid of $\epsilon$. Replace terminal letters in all nontrivial right-hand sides of productions by fresh nonterminal ones. Shorten right-hand sides of productions by using new nonterminal letters.
Example 19.4. \( \text{PAREN} - \{ \varepsilon \} \).

\[
S \rightarrow [S], \ S \rightarrow SS, \ S \rightarrow \varepsilon \\
S \rightarrow [S], \ S \rightarrow SS, \ S \rightarrow [ \\
S \rightarrow LSR, \ S \rightarrow SS, \ S \rightarrow LR, \ L \rightarrow [ \ , \ R \rightarrow ] \\
S \rightarrow AR, \ A \rightarrow LS, \ S \rightarrow SS, \ S \rightarrow LR, \ L \rightarrow [ \ , \ R \rightarrow ]
\]

Example 19.5. Palindromes (\( \neq \varepsilon \)).

\[
S \rightarrow aSa | bSb | a | b | \varepsilon \\
S \rightarrow aSa | bSb | aa | bb | a | b \\
S \rightarrow ASA | BSB | AA | BB | A | B, \ A \rightarrow a, \ B \rightarrow b \\
S \rightarrow AC | BD | AA | BB | A | B, \ C \rightarrow SA, \ D \rightarrow SB, \ A \rightarrow a, \ B \rightarrow b
\]

Definition 19.6. Greibach Normal Form (GNF) is a CFG whose productions are of the form \( A \rightarrow aB_1B_2 \ldots B_n \), where \( a \) is terminal and \( B_i \)'s are nonterminal letters, \( n \geq 0 \). In particular, when \( n = 0 \) the production becomes \( A \rightarrow a \).

Theorem 19.7. For any CFG \( G \) there is a GNF \( G' \) such that \( L(G') = L(G) - \{ \varepsilon \} \).

Proof. Skip it.

Example 19.8. The Greibach normal form

\[
S \rightarrow aB | bA, \ A \rightarrow aS | bAA | a, \ B \rightarrow bS | aBB | b
\]
generates exactly the set of nonnull strings over \( \{ a, b \} \) with equally many \( a \)'s and \( b \)'s.

Claim: all generated strings have equal number of \( a \)'s and \( b \)'s. Proof: by a straightforward induction on the length of derivation we establish that 'a generated string (possibly with non-terminals) has equal number of \( a \)'s and \( b \)'s, case non sensitive.

Claim: all nonnull strings with equal number of \( a \)'s and \( b \)'s are generated by this grammar. The proof of this claim will be left as a homework problem. Meanwhile consider some examples of derivations.

\[
S \rightarrow aB \rightarrow aaBB \rightarrow aaBBB \rightarrow aaabBB \rightarrow aaabbb \\
S \rightarrow aB \rightarrow aaBB \rightarrow aabSB \rightarrow aabaBB \rightarrow aababB \rightarrow aababb \\
S \rightarrow aB \rightarrow aaBB \rightarrow aabB \rightarrow aabbS \rightarrow aabbbA \rightarrow aabbbba
\]

Homework problems. Page 307 of Kozen’s book: \( \S \) 1 (remaining part), \( \S \) 4 (CNF only).