Chap 6 Simplification of Context-Free Grammars and Normal Forms

6.1 Methods for Transforming Grammars

Let L be any context-free language, and let $G = (V, T, S, P)$ be a context-free grammar for $L - \{\lambda\}$. Then the grammar we obtain by adding to V the new variable S_0, making S_0 the start variable, and adding to P the productions

$$S_0 \to S|\lambda$$

generates L. Therefore, any nontrivial conclusion we can make for $L - \{\lambda\}$ will almost certainly transfer to L. Also, given any context-free grammar G, there is a method for obtaining $\widehat{G}$ such that $L\left(\widehat{G}\right) = L(G) - \{\lambda\}$

[1] ## A Useful Substitution Rule

Theorem 6.1 Let $G = (V, T, S, P)$ be a context-free grammar. Suppose that P contains a production of the form

$$A \to x_1 B x_2.$$

Assume that A and B are different variables and that

$$B \to y_1 | y_2 | \cdots | y_n$$

is the set of all productions in P that have B as the left side. Let $\widehat{G} = \left(V, T, S, \widehat{P}\right)$ be the grammar in which $\widehat{P}$ is constructed by deleting

$$A \to x_1 B x_2 \tag{6.1}$$

from P, and adding to it

$$A \to x_1 y_1 x_2 \,|x_1 y_2 x_2|\cdots|x_1 y_n x_2.$$

Then

$$L\left(\widehat{G}\right) = L\left(G\right).$$

Example 6.1 Consider $G = (\{A, B\}, \{a, b, c\}, A, P)$ with productions

$$\begin{aligned} A &\to a\,|aaA|\,abBc, \\ B &\to abbA|b. \end{aligned}$$

Using the suggested substitution for the variable B, we get the grammar $\widehat{G}$ with productions

$$\begin{aligned} A &\to a\,|aaA|\,ababbAc|abbc, \\ B &\to abbA|b. \end{aligned}$$

The new grammar $\widehat{G}$ is equivalent to G. The string $aaabbc$ has the derivation

$$A \Rightarrow aaA \Rightarrow aaabBc \Rightarrow aaabbc$$

in G, and the corresponding derivation

$$A \Rightarrow aaA \Rightarrow aaabbc$$

in $\widehat{G}$.

Notice that, in this case, the variable B and its associated productions are still in the grammar even though they can no longer play a part in any derivation. We will next show how such unnecessary productions can be removed from a grammar. ■

[2] ## Removing Useless Productions

One invariably wants to remove productions from a grammar that can never take part in any derivation. For example, in the grammar whose entire production set is

$$S \to aSb \,|\, \lambda \,|\, A,$$
$$A \to aA,$$

the production $S \to A$ clearly plays no role, as A cannot be transformed into a terminal string. While A can occur in a string derived from S, this can never lead to a sentence. Removing this production leaves the language unaffected and is a simplification by any definition.

Definition 6.1

Let $G = (V, T, S, P)$ be a context-free grammar. A variable $A \in V$ is said to be **useful** if and only if there is at least one $w \in L(G)$ such that

$$S \overset{*}{\Rightarrow} xAy \overset{*}{\Rightarrow} w, \tag{6.2}$$

with x, y in $(V \cup T)^*$. In words, a variable is useful if and only if it occurs in at least one derivation. A variable that is not useful is called **useless**. A production is useless if it involves any useless variable.

Example 6.2 A variable may be useless because there is no way of getting a terminal string from it. The case just mentioned is of this kind. Another reason a variable may be useless is shown in the next grammar. In a grammar with start symbol S and productions

$$S \rightarrow A,$$
$$A \rightarrow aA|\lambda,$$
$$B \rightarrow bA,$$

the variable B is useless and so is the production $B \rightarrow bA$. Although B can derive a terminal string, there is no way we can achieve $S \stackrel{*}{\Rightarrow} xBy$.

Example 6.3 Eliminate useless symbols and productions from $G = (V, T, S, P)$, where $V = \{S, A, B, C\}$ and $T = \{a, b\}$, with P consisting of

$$\begin{aligned} S &\rightarrow aS \,|A|\, C, \\ A &\rightarrow a, \\ B &\rightarrow aa, \\ C &\rightarrow aCb. \end{aligned}$$

First, we identify the set of variables that can lead to a terminal string. Because $A \rightarrow a$ and $B \rightarrow aa$, the variables A and B belong to this set. So does S, because $S \Rightarrow A \Rightarrow a$. However, this argument cannot be made for C, thus identifying it as useless. Removing C and its corresponding productions, we are led to the grammar G_1 with variables $V_1 = \{S, A, B\}$, terminals $T = \{a\}$, and productions

① Remove C ∵ C cannot lead to a terminal string!

$$S \to aS|A,$$
$$A \to a,$$
$$B \to aa.$$

② Remove B ∵ B cannot be reached from S

Next we want to eliminate the variables that cannot be reached from the start variable. For this, we can draw a **dependency graph** for the variables. Dependency graphs are a way of visualizing complex relationships and are found in many applications. For context-free grammars, a dependency graph has its vertices labeled with variables, with an edge between vertices C and D if and only if there is a production of the form

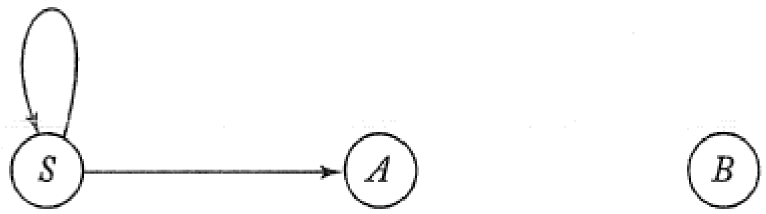
$$C \to xDy.$$

A dependency graph for V_1 is shown in Figure 6.1. A variable is useful only if there is a path from the vertex labeled S to the vertex labeled with that variable. In our case, Figure 6.1 shows that B is useless. Removing it and the affected productions and terminals, we are led to the final answer $\widehat{G} = \left(\widehat{V}, \widehat{T}, S, \widehat{P}\right)$ with $\widehat{V} = \{S, A\}, \widehat{T} = \{a\}$, and productions

$$\begin{aligned} S &\to aS|A, \\ A &\to a. \end{aligned}$$

The formalization of this process leads to a general construction and the corresponding theorem.

Figure 6.1



Theorem 6.2 Let $G = (V, T, S, P)$ be a context-free grammar. Then there exists an equivalent grammar $\widehat{G} = \left(\widehat{V}, \widehat{T}, S, \widehat{P}\right)$ that does not contain any useless variables or productions.

[3]

Removing λ-Productions

One kind of production that is sometimes undesirable is one in which the right side is the empty string.

Definition 6.2

Any production of a context-free grammar of the form

$$A \to \lambda$$

is called a **λ-production**. Any variable A for which the derivation

$$A \stackrel{*}{\Rightarrow} \lambda \tag{6.3}$$

is possible is called **nullable**.

Example 6.4 Consider the grammar

$$S \rightarrow aS_1b,$$
$$S_1 \rightarrow aS_1b|\lambda,$$

with start variable S. This grammar generates the λ-free language $\{a^n b^n : n \geq 1\}$. The λ-production $S_1 \rightarrow \lambda$ can be removed after adding new productions obtained by substituting λ for S_1 where it occurs on the right. Doing this we get the grammar

$$S \rightarrow aS_1b|ab,$$
$$S_1 \rightarrow aS_1b|ab.$$

We can easily show that this new grammar generates the same language as the original one.

In more general situations, substitutions for λ-productions can be made in a similar, although more complicated, manner.

Theorem 6.3 Let G be any context-free grammar with λ not in $L(G)$. Then there exists an equivalent grammar $\widehat{G}$ having no λ-productions.

Proof: We first find the set V_N of all nullable variables of G, using the following steps.

1. For all productions $A \rightarrow \lambda$, put A into V_N.

2. Repeat the following step until no further variables are added to V_N.

   For all productions

$$B \rightarrow A_1 A_2 \cdots A_n,$$

   where $A_1, A_2, ..., A_n$ are in V_N, put B into V_N.

Once the set V_N has been found, we are ready to construct $\widehat{P}$. To do so, we look at all productions in P of the form

$$A \to x_1 x_2 \cdots x_m, m \geq 1,$$

where each $x_i \in V \cup T$. For each such production of P, we put into $\widehat{P}$ that production as well as all those generated by replacing nullable variables with λ in all possible combinations. For example, if x_i and x_j are both nullable, there will be one production in $\widehat{P}$ with x_i replaced with λ, one in which x_j is replaced with λ, and one in which both x_i and x_j are replaced with λ. There is one exception: If all x_i are nullable, the production $A \to \lambda$ is not put into $\widehat{P}$.

The argument that this grammar $\widehat{G}$ is equivalent to G is straightforward and will be left to the reader. ■

Example 6.5 Find a context-free grammar without λ-productions equivalent to the grammar defined by

$$\begin{aligned} S &\to ABaC, \\ A &\to BC, \\ B &\to b|\lambda, \\ C &\to D|\lambda, \\ D &\to d. \end{aligned}$$

From the first step of the construction in Theorem 6.3, we find that the nullable variables are A, B, C. Then, following the second step of the construction, we get

$$
\begin{aligned}
S &\rightarrow ABaC \,|BaC|\, AaC \,|ABa|\, aC \,|Aa|\, Ba|a,\\
A &\rightarrow B\,|C|\,BC,\\
B &\rightarrow b,\\
C &\rightarrow D,\\
D &\rightarrow d.
\end{aligned}
$$

■

[4] Removing Unit-Productions

As we have seen in Theorem 5.2, productions in which both sides are a single variable are at times undesirable.

Definition 6.3

Any production of a context-free grammar of the form

$$A \rightarrow B,$$

where $A, B \in V$, is called a **unit-production**.

To remove unit-productions, we use the substitution rule discussed in Theorem 6.1. As the construction in the next theorem shows, this can be done if we proceed with some care.

Theorem 6.4

Let $G = (V, T, S, P)$ be any context-free grammar without λ-productions. Then there exists a context-free grammar $\widehat{G} = \left(\widehat{V}, \widehat{T}, S, \widehat{P}\right)$ that does not have any unit-productions and that is equivalent to G.

Proof: Obviously, any unit-production of the form $A \rightarrow A$ can be removed from the grammar without effect, and we need only consider $A \rightarrow B$, where A and B are different variables. At first sight, it may seem that we can use Theorem 6.1 directly with $x_1 = x_2 = \lambda$ to replace

$$A \rightarrow B$$

with

$$A \rightarrow y_1 | y_2 | \cdots | y_n.$$

But this will not always work; in the special case

$$A \to B,$$
$$B \to A,$$

the unit-productions are not removed. To get around this, we first find, for each A, all variables B such that

$$A \stackrel{*}{\Rightarrow} B. \tag{6.4}$$

We can do this by drawing a dependency graph with an edge (C, D) whenever the grammar has a unit-production $C \to D$; then (6.4) holds whenever there is a walk between A and B. The new grammar $\widehat{G}$ is generated by first putting into $\widehat{P}$ all non-unit productions of P. Next, for all A and B satisfying (6.4), we add to $\widehat{P}$

$$A \to y_1 |y_2| \cdots |y_n,$$

where $B \to y_1 |y_2| \cdots y_n$ is the set of all rules in $\widehat{P}$ with B on the left. Note that since $B \to y_1 |y_2| \cdots |y_n$ is taken from $\widehat{P}$, none of the y_i can be a single variable, so that no unit-productions are created by the last step.

To show that the resulting grammar is equivalent to the original one, we can follow the same line of reasoning as in Theorem 6.1. ■

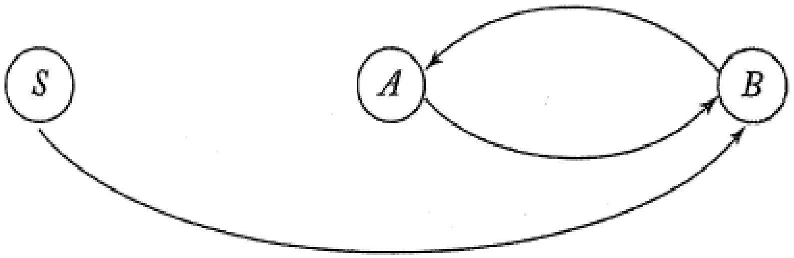
Example 6.6 Remove all unit-productions from

$$S \rightarrow Aa|B,$$
$$B \rightarrow A|bb,$$
$$A \rightarrow a\,|bc|\,B.$$

The dependency graph for the unit-productions is given in Figure 6.3; we see from it that $S \stackrel{*}{\Rightarrow} A, S \stackrel{*}{\Rightarrow} B, B \stackrel{*}{\Rightarrow} A$, and $A \stackrel{*}{\Rightarrow} B$. Hence, we add to the original non-unit productions

$$S \rightarrow Aa,$$
$$A \rightarrow a|bc,$$
$$B \rightarrow bb,$$

Figure 6.3



the new rules

$$
\begin{aligned}
S &\rightarrow a\,|bc|\,bb, \\
A &\rightarrow bb, \\
B &\rightarrow a|bc,
\end{aligned}
$$

to obtain the equivalent grammar

$$
\begin{aligned}
S &\rightarrow a\,|bc|\,bb|Aa, \\
A &\rightarrow a\,|bb|\,bc, \\
B &\rightarrow a\,|bb|\,bc.
\end{aligned}
$$

Note that the removal of the unit-productions has made B and the associated productions useless. ■

Theorem 6.5 Let L be a context-free language that does not contain λ. Then there exists a context-free grammar that generates L and that does not have any useless productions, λ-productions, or unit-productions.

1. Remove λ-productions.
2. Remove unit-productions.
3. Remove useless productions.