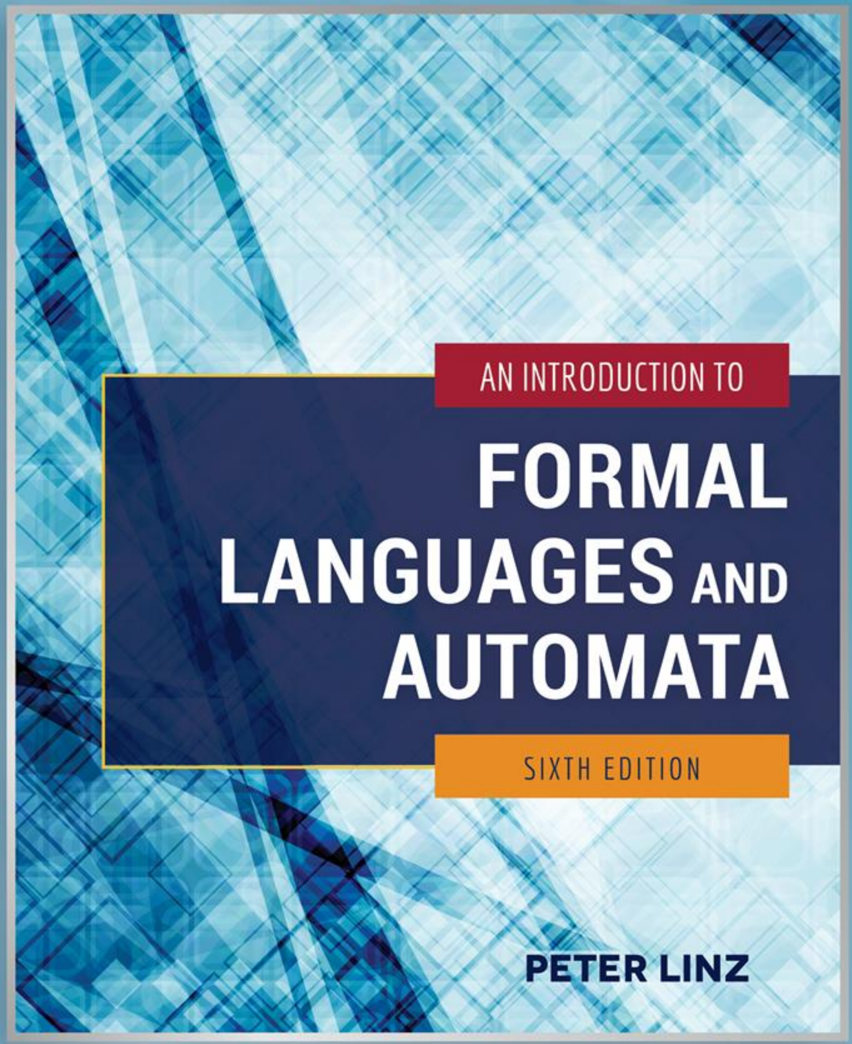
Chapter 6

SIMPLIFICATION OF CONTEXT-FREE GRAMMARS AND NORMAL FORMS



Learning Objectives

At the conclusion of the chapter, the student will be able to:

- Simplify a context-free grammar by removing useless productions
- Simplify a context-free grammar by removing λ-productions
- Simplify a context-free grammar by removing unit-productions
- Determine whether or not a context-free grammar is in Chomsky normal form
- Transform a context-free grammar into an equivalent grammar in Chomsky normal form
- Determine whether or not a context-free grammar is in Greibach normal form
- Transform a context-free grammar into an equivalent grammar in Greibach normal form

Methods for Transforming Grammars

- The definition of a context-free grammar imposes no restrictions on the right side of a production
- In some cases, it is convenient to restrict the form of the right side of all productions
- Simplifying a grammar involves eliminating certain types of productions while producing an equivalent grammar, but does not necessarily result in a reduction of the total number of productions
- For simplicity, we focus on languages that do not include the empty string

A Useful Substitution Rule

- Theorem 6.1 states that, If A and B are distinct variables, a production of the form A → uBv can be replaced by a set of productions in which B is substituted by all strings B derives in one step.
- Consider the grammar

  V = { A, B }, T = { a, b, c }, and productions

  $A \rightarrow a \mid aaA \mid abBc$

  $B \rightarrow abbA \mid b$

- We can replace $A \rightarrow abBc$ with two productions that replace B (in red), obtaining an equivalent grammar with productions

  $A \rightarrow a \mid aaA \mid ababbAc \mid abbc$

  $B \rightarrow abbA \mid b$

Useless Productions

- A variable is *useful* if it occurs in the derivation of at least one string in the language
- Otherwise, the variable and any productions in which it appears is considered *useless*
- A variable is useless if:
  - No terminal strings can be derived from the variable
  - The variable symbol cannot be reached from S
- In the grammar below, B can never be reached from the start symbol S and is therefore considered useless

  $S \to A$

  $A \to aA \mid \lambda$

  $B \to bA$

Removing Useless Productions

It is always possible to remove useless productions from a context-free grammar:

1. Let V_1 be the set of useful variables, initialized to empty
2. Add a variable A to V_1 if there is a production of the form

   $A \rightarrow$ terminal symbols or variables in V_1

   (Repeat until nothing else can be added to V_1)
3. Eliminate any productions containing variables not in V_1
4. Use a dependency graph to identify and eliminate variables that are unreachable from S

Application of the Procedure for Removing Useless Productions

- Consider the grammar from example 6.3:
  $S \rightarrow aS \mid A \mid C$
  $A \rightarrow a$
  $B \rightarrow aa$
  $C \rightarrow aCb$
- In step 2, variables A, B, and S are added to V_1
- Since C is useless, it is eliminated in step 3, resulting in the grammar with productions
  $S \rightarrow aS \mid A$
  $A \rightarrow a$
  $B \rightarrow aa$
- In step 4, B is identified as unreachable from S, resulting in the grammar with productions
  $S \rightarrow aS \mid A$
  $A \rightarrow a$

λ-Productions

- A production with λ on the right side is called a *λ-production*
- A variable A is called *nullable* if there is a sequence of derivations through which A produces λ
- If a grammar generates a language not containing λ, any λ-productions can be removed
- In the grammar below, S_1 is nullable

  $S \rightarrow aS_1b$

  $S_1 \rightarrow aS_1b \mid \lambda$
- Since the language is λ-free, we have the equivalent grammar

  $S \rightarrow aS_1b \mid ab$

  $S_1 \rightarrow aS_1b \mid ab$

Removing λ-Productions

It is possible to remove λ-productions from a context-free grammar that does not generate λ:

1. Let V_N be the set of nullable variables, initialized to empty
2. Add a variable A to V_N if there is a production having one of the forms:
   - $A \rightarrow \lambda$
   - $A \rightarrow$ variables already in V_N

   (Repeat until nothing else can be added to V_N)
3. Eliminate λ-productions
4. Add productions in which nullable symbols are replaced by λ in all possible combinations

Application of the Procedure for Removing λ-Productions

- Consider the grammar from example 6.5:
  $S \to ABaC$
  $A \to BC$
  $B \to b \mid \lambda$
  $C \to D \mid \lambda$
  $D \to d$
- In step 2, variables B, C, and A (in that order) are added to V_N
- In step 3, λ-productions are eliminated
- In step 4, productions are added by replacing nullable symbols with in λ all possible combinations, resulting in
  $S \to ABaC \mid BaC \mid AaC \mid Aba \mid aC \mid Aa \mid Ba \mid a$
  $A \to B \mid C \mid BC$
  $B \to b$
  $C \to D$
  $D \to d$

Unit-Productions

- A production of the form A $\rightarrow$ B (where A and B are variables) is called a *unit-production*
- Unit-productions add unneeded complexity to a grammar and can usually be removed by simple substitution
- Theorem 6.4 states that any context-free grammar without λ-productions has an equivalent grammar without unit-productions
- The procedure for eliminating unit-productions assumes that all λ-productions have been previously removed

Removing Unit-Productions

1. Draw a dependency graph with an edge from A to B corresponding to every A $\rightarrow$ B production in the grammar
2. Construct a new grammar that includes all the productions from the original grammar, except for the unit-productions
3. Whenever there is a path from A to B in the dependency graph, replace B using the substitution rule from Theorem 6.1, but using only the productions in the new grammar

Application of the Procedure for Removing Unit-Productions

- Consider the grammar from example 6.6:

  $S \rightarrow Aa \mid B$

  $A \rightarrow a \mid bc \mid B$

  $B \rightarrow A \mid bb$

  The dependency graph contains paths from S to A, S to B, B to A, and A to B

- After removing unit-productions and adding the new productions (in red), the resulting grammar is

  $S \rightarrow Aa \mid a \mid bc \mid bb$

  $A \rightarrow a \mid bc \mid bb$

  $B \rightarrow a \mid bc \mid bb$

Simplification of Grammars

- Theorem 6.5 states that, for any context-free language that does not include λ, there is a context-free grammar without useless, λ-, or unit-productions
- Since the removal of one type of production may introduce productions of another type, undesirable productions should be removed in the following order:
    1. Remove λ-productions
    2. Remove unit-productions
    3. Remove useless productions

Chomsky Normal Form

- In Chomsky normal form, the number of symbols on the right side of a production is strictly limited.
- A context-free grammar is in *Chomsky normal form* if all of its productions are in one of the forms below (A, B, C are variables; a is a terminal symbol)
  - $A \to BC$
  - $A \to a$
- The grammar below is in Chomsky normal form

$$S \to AS \mid a$$
$$A \to SA \mid b$$

Transforming a Grammar into Chomsky Normal Form

For any context-free grammar that does not generate λ, it is possible to find an equivalent grammar in Chomsky normal form:

1. Copy any productions of the form $A \to a$
2. For other productions containing a terminal symbol x on the right side, replace x with a variable X and add the production $X \to x$
3. Introduce additional variables to reduce the lengths of the right sides of productions as necessary, replacing long productions with productions of the form $W \to YZ$ (W, Y, Z are variables)

Application of the Procedure for Removing Unit-Productions

- Consider the grammar from example 6.8, which is clearly not in Chomsky normal form

  $S \rightarrow ABa$

  $A \rightarrow aab$

  $B \rightarrow Ac$

- After replacing terminal symbols with new variables and adding new productions (in red), the resulting grammar is

  $S \rightarrow AC$

  $C \rightarrow BX$

  $A \rightarrow XD$

  $D \rightarrow XY$

  $B \rightarrow AZ$

  $X \rightarrow a$

  $Y \rightarrow b$

  $Z \rightarrow c$

Greibach Normal Form

- In Greibach normal form, there are restrictions on the positions of terminal and variable symbols
- A context-free grammar is in *Greibach Normal Form* if, in all of its productions, the right side consists of single terminal followed by any number of variables
- The grammar below is in Greibach normal form

$$S \rightarrow aAB \mid bBB \mid bB$$
$$A \rightarrow aA \mid bB \mid b$$
$$B \rightarrow b$$

Transforming a Grammar into Greibach Normal Form

- For any context-free grammar that does not generate λ, it is possible to find an equivalent grammar in Greibach normal form
- Consider the grammar from example 6.10, which is clearly not in Greibach normal form

  $S \rightarrow abSb \mid aa$

- After replacing terminal symbols with new variables and adding new productions (in red), the resulting grammar is

  $S \rightarrow aBSB \mid aA$

  $A \rightarrow a$

  $B \rightarrow b$