Name: _____ Date: _____

Note: The purpose of the following questions is:

• *Enhance learning*	• *Summarized points*	• *Analyze abstract ideas*

Class: 09 Simplifications of Context-Free Grammars & Normal Forms

The definition of a context-free grammar imposes no restriction whatsoever on the right side of a production. However, complete freedom is not necessary and, in fact, is a detriment in some arguments. In Theorem 5.2 (Linz 5e, p138), we see the convenience of certain restrictions on grammatical forms; eliminating rules of the form $A \rightarrow \lambda$ and $A \rightarrow B$ make the argument easier. In many instances, it is desirable to place even more stringent restrictions on the grammar. Because of this we need to look at methods for transforming an arbitrary context-free grammar into an equivalent one that satisfies certain restrictions on its form.

1. Define *normal form*? And what is the use of it?
2. Define the term *simplification* and how it is used here?
3. For the following grammar, substitute $B \rightarrow b$ and $B \rightarrow aA$ to find the equivalent grammar:

$$S \rightarrow aB$$
$$A \rightarrow aaA$$
$$A \rightarrow abBc$$
$$B \rightarrow aA$$
$$B \rightarrow b$$

4. Eliminate the λ-production from the grammar:

$$S \rightarrow aMb$$
$$M \rightarrow aMb$$
$$M \rightarrow \lambda$$

5. Remove all unit-productions from the grammar

$$S \rightarrow aA$$
$$A \rightarrow a$$
$$A \rightarrow B$$
$$B \rightarrow A$$
$$B \rightarrow bb$$

6. Remove useless productions from the grammar

$$S \rightarrow aS \mid A \mid C$$
$$A \rightarrow a$$
$$B \rightarrow aa$$
$$C \rightarrow aCb$$

Normal Forms for Context-free Grammars:

Chomsky Normal Form: One kind of normal form we can look for is one in which the number of symbols on

the right of a production is strictly limited. In particular, we can ask that the string on the right of a production consist of no more than two symbols. One instance of this is the Chomsky normal form.

7. Define Chomsky Normal Form.
8. Give example for a grammar in *Chomsky* normal form and another is *Not* Chomsky normal form.
9. Convert the grammar with productions to Chomsky normal form

$$S \to ABa$$
$$A \to aab$$
$$B \to Ac$$

10. What is the procedure to convert context-free grammar *not* in Chomsky Normal Form to an equivalent grammar in Chomsky Normal Form.
11. What are your observations about Chomsky normal forms?

Greibach Normal Form This is another useful grammatical form. Here we put restrictions not on the length of the right side of a production, but on the position in which terminals and variables can appear. Arguments justifying Greibach normal form are a little complicated and not very transparent. Similarly, constructing grammar in Greibach normal form equivalent to a given context-free grammar is tedious. We therefore deal with this matter very briefly. Nevertheless, Greibach normal form has many theoretical and practical consequences.

12. What is Greinbach Normal Form?
13. Give a grammar example in Greinbach normal form and another *Not* in Greinbach normal form?
14. Convert the grammar into Greinbach normal form

    S → abSb

    S → aa
15. What are your observations about Greinbach normal form?

The CYK Parser (*Algorithm*)

The algorithm works only if the grammar is in Chomsky normal form and succeeds by breaking one problem into a sequence of smaller ones.

Use the CYK algorithm to determine whether the string ***w = aabbb*** is in the language generated by the grammar

$$S \to AB$$
$$A \to BB$$
$$A \to a$$
$$B \to AB$$
$$B \to b$$

16. Use the approach employed in the previous exercise to show how the CYK membership algorithm can be used into a parsing method.
17. What is the time complexity of the CYK membership algorithm?
18. What are your observations about the CYK algorithm?

The CYK algorithm, as described here, determines membership for any language generated by a grammar in Chomsky normal form. With some additions to keep track of how elements of V_{ij} are derived, it can be converted into parsing method.

Note: The material in this section is a good example of dynamic programming, and students can benefit from studying it.