23MT2014

# THEORY OF COMPUTATION

Topic:

# CHOMSKY NORMAL FORM- PART-2

Session - 1

# AIM OF THE SESSION

Aim: To understand and apply Chomsky Normal Form (CNF) in the context of context-free languages.

# INSTRUCTIONAL OBJECTIVES

This Session is designed to:

1. Convert a given context-free grammar into Chomsky Normal Form (CNF) using appropriate transformation rules.
2. Create a grammar in CNF for a given language.
3. Evaluate the correctness of a grammar in CNF.

# LEARNING OUTCOMES

At the end of this session, you should be able to:

1. Define Chomsky Normal Form (CNF) and explain its significance in the study of context-free languages.
2. Identify the necessary steps and transformation rules to convert a given context-free grammar into CNF.
3. Apply the transformation rules to convert a context-free grammar into CNF.

# In general:

From any context-free grammar
(which doesn't produce $\lambda$ )
not in Chomsky Normal Form

we can obtain:
An equivalent grammar
in Chomsky Normal Form

# The Procedure

First remove:

Nullable variables

Unit productions

Then, for every symbol $a$:

Add production $T_a \to a$

In productions: replace $a$ with $T_a$

New variable: $T_a$

Replace any production $A \to C_1 C_2 \cdots C_n$

with
$$A \to C_1 V_1$$
$$V_1 \to C_2 V_2$$
$$\cdots$$
$$V_{n-2} \to C_{n-1} C_n$$

New intermediate variables: $V_1, V_2, \ldots, V_{n-2}$

**Theorem:** For any context-free grammar (which doesn't produce $\lambda$ ) there is an equivalent grammar in Chomsky Normal Form

# Observations

- Chomsky normal forms are good for parsing and proving theorems
- It is very easy to find the Chomsky normal form for any context-free grammar

# MCQ

Question 1: Which of the following statements is true about the conversion of a context-free grammar to Chomsky Normal Form? A) The resulting grammar will always have fewer non-terminals. B) The resulting grammar will always have fewer production rules. C) The resulting grammar will always be unambiguous. D) The resulting grammar will always have a unique start symbol.

Answer: B) The resulting grammar will always have fewer production rules.

Question 2: Which of the following is not a step in the conversion of a context-free grammar to Chomsky Normal Form? A) Eliminating epsilon productions. B) Removing unit productions. C) Factoring out common prefixes. D) Eliminating left recursion.

Answer: D) Eliminating left recursion.

Question 3: In Chomsky Normal Form, what is the purpose of introducing new non-terminals? A) To reduce the number of production rules. B) To eliminate ambiguity in the grammar. C) To ensure every non-terminal has at least one production rule. D) To simplify the parsing process.

Answer: C) To ensure every non-terminal has at least one production rule.

Question 4: Which of the following is a characteristic of a grammar in Chomsky Normal Form? A) All non-terminals must be reachable from the start symbol. B) The start symbol must be a terminal. C) Every production rule must have a single non-terminal on the right-hand side. D) Epsilon productions are allowed.

Answer: C) Every production rule must have a single non-terminal on the right-hand side.

# Terminal questions

1. How does the Greibach Normal Form simplify the parsing process in compiler design? Provide a detailed explanation of the steps involved in converting a context-free grammar to GNF and discuss the benefits of using GNF in compiler implementations.

2. Describe the role of the Greibach Normal Form in natural language processing (NLP) tasks. Explain how converting context-free grammars to GNF can aid in syntactic analysis, dependency parsing, and other NLP applications. Provide examples to illustrate your points.

3. Discuss the relationship between the Greibach Normal Form and other normal forms, such as the Chomsky Normal Form (CNF) and the Backus-Naur Form (BNF). Compare and contrast these normal forms in terms of their properties, restrictions, and applications.

4. Explore the practical implications of converting a context-free grammar to GNF in terms of efficiency and complexity. Analyze the impact of GNF on the parsing time and space complexity, and discuss any trade-offs or limitations associated with using GNF.

5. Provide a case study of a real-world application where the Greibach Normal Form was used to enhance the performance of a parsing algorithm. Explain the problem domain, the context-free grammar representation, and how GNF was applied to improve the efficiency or accuracy of the parsing process.

6. Investigate the impact of GNF on error detection and error recovery in context-free grammars. Explain how the structural properties of GNF can help identify syntax errors and suggest possible error recovery strategies when parsing inputs that do not conform to the grammar rules.

7. Discuss the computational complexity of converting a context-free grammar to GNF. Analyze the time and space complexity of the conversion algorithm and explore any existing optimizations or algorithms that can expedite the process.

8. Examine the relationship between GNF and the concept of ambiguity in context-free grammars. Explain how GNF can help resolve ambiguity and discuss any limitations or challenges in achieving unambiguous grammars using GNF.

9. Explore the role of GNF in grammar-based machine translation systems. Discuss how GNF can be utilized to represent and parse source and target language grammars, enabling the translation process. Provide examples and discuss any challenges or considerations in this application domain.

10. Investigate the use of GNF in automated code generation and program synthesis. Discuss how GNF can assist in generating syntactically correct code or programs from high-level specifications, and analyze the advantages and limitations of using GNF in this context.

THANK YOU

Team – TOC