**Chapter 4: Lexical and Syntax Analysis**

**1. Differentiate between top-down and bottom-up parsing.** **What is left factoring?**

**Ans:**

|  |  |
| --- | --- |
| **Top-Down Parsing** | **Bottom-Up Parsing** |
| In Top-Down Parsing, the parse tree is built from the root downward to the leaves. | In Bottom-Up Parsing, the parse tree  is built from the leaves upward to the root. |
| This parsing technique uses Left Most Derivation. | This parsing technique uses Right Most Derivation. |
| A top-down parser can be easily structured and formed. | It is difficult to produce a bottom-up parser. |

**Left factoring:** Left factoring is a useful grammar transformation used in parsing. It removes the common left factor that appears in two productions of the same non-terminal.

**2. What is Left Recursion? Define lexeme and token.**

**Ans:**

**Left Recursion:** A production of grammar is said to have left recursion if the leftmost variable of its RHS is same as variable of its LHS.

**Lexeme:** A lexeme is the lowest level syntactic unit of a language (e.g., \*, sum, begin)

**Token:** A token is a category of lexemes (e.g., identifier)

**3. Explain the three reasons why lexical analysis is separated from syntax analysis.**

**Ans:**

**There are three reasons why lexical analysis is separated from syntax analysis:**

* **Simplicity** - less complex approaches can be used for lexical analysis; separating them simplifies the parser
* **Efficiency** - separation allows optimization of the lexical analyzer
* **Portability** - parts of the lexical analyzer may not be portable, but the parser always is portable.

**4. Describe three advantages of LR parsers.**

**Ans:**

* They will work for nearly all grammars that describe programming languages.
* They can detect syntax errors as soon as it is possible.
* They work on a larger class of grammars than other bottom-up algorithms, but are as efficient as any other bottom-up parser.

|  |  |
| --- | --- |
| **5.** |  |
| **Ans:** |  |
| **6.** |  |
| **Ans:** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **7.** | |  | | |
| **Ans:** | | |  | |
| **8.** |  | | |
| **Ans:** | | |  |
|  | | | |

**9. Describe the Advantages of Using BNF to Describe Syntax**

**Ans:**

**Advantages of Using BNF to Describe Syntax:**

* Provides a clear and concise syntax description
* The parser can be based directly on the BNF
* Parsers based on BNF are easy to maintain

**10. Describe briefly the three approaches to building a lexical analyzer.**

**Ans:**

**Three approaches to building a lexical analyzer:**

* Firstly, write a formal description of the tokens. Then use a software tool that constructs a table-driven lexical analyzer from the description.
* Secondly, design a state diagram that describes the tokens. Then write a program that implements the state diagram.
* Finally design a state diagram that describes the tokens. Then hand-construct a table-driven implementation of the state diagram.