1. Variables/Non-terminals (V): These are symbols that represent categories or groups of words in the language. They're typically denoted by capital letters like A, B, C, etc.
2. Terminals (T): Terminals are the actual words or symbols in the language. They're the building blocks of sentences. Terminals are usually represented by lowercase letters or symbols like a, b, c, x, y, z, etc.
3. Productions (P): Productions are rules that dictate how variables and terminals can be combined to form phrases or sentences. Each production is in the form $\alpha \rightarrow \beta$, where α and β are strings made up of a combination of variables and terminals. α must contain at least one variable. These rules are sometimes called rewriting rules.
4. Starting Symbol (S): This is a special variable or non-terminal that indicates where the construction of sentences or phrases begins.

When writing a grammar, it's important to ensure that variables and terminals are distinct, meaning no terminal can be a non-terminal and vice versa. In other words, there should be no overlap between the sets V and T.

Related Posts:

1. What are the types of passes in compiler ?
2. Discuss the role of compiler writing tools. Describe various compiler writing tools.
3. What do you mean by regular expression ? Write the formal recursive definition of a regular expression.
4. How does finite automata useful for lexical analysis ?
5. Explain the implementation of lexical analyzer.
6. Write short notes on lexical analyzer generator.
7. Explain the automatic generation of lexical analyzer.
8. Explain the term token, lexeme and pattern.

9. What are the various LEX actions that are used in LEX programming ?
10. Explain formal grammar and its application to syntax analyzer.
11. Define parse tree. What are the conditions for constructing a parse tree from a CFG ?
12. Describe the capabilities of CFG.
13. What is parser ? Write the role of parser. What are the most popular parsing techniques ? OR Explain about basic parsing techniques. What is top-down parsing ? Explain in detail.
14. What are the common conflicts that can be encountered in shift-reduce parser ?
15. Differentiate between top-down and bottom-up parser.Under which conditions predictive parsing can be constructed for a grammar ?
16. Differentiate between recursive descent parsing and predictive parsing.
17. What is the difference between S-attributed and L-attributed definitions ?
18. What is intermediate code generation and discuss benefits of intermediate code ?
19. Define parse tree. Why parse tree construction is only possible for CFG ?
20. Discuss symbol table with its capabilities ?
21. What are the symbol table requirements ? What are the demerits in the uniform structure of symbol table ?