

What do you mean by regular expression ? Write the formal recursive definition of a regular expression.

1.Regular expression is a formula in a special language that is used for specifying simple classes of strings.

2.A string is a sequence of symbols; for the purpose of most text-based search techniques, a string is any sequence of alphanumeric characters (letters, numbers, spaces, tabs, and punctuation).

Formal recursive definition of regular expression :

Formally, a regular expression is an algebraic notation for characterizing a set of strings.

1.Any terminals, i.e., the symbols belong to S are regular expression. Null string (ϵ) and null set (\emptyset) are also regular expression.

2.If P and Q are two regular expressions then the union of the two regular expressions, denoted by $P + Q$ is also a regular expression.

3.If P and Q are two regular expressions then their concatenation denoted by PQ is also a regular expression.

4.If P is a regular expression then the iteration (repetition or closure) denoted by P^* is also a regular expression.

5.If P is a regular expression then P , is a regular expression.

6.The expressions got by repeated application of the rules from (1) to (5) over Σ are also regular expression.

What do you mean by regular expression ? Write the formal recursive definition of a regular expression.

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. How does finite automata useful for lexical analysis ?
4. Explain the implementation of lexical analyzer.
5. Write short notes on lexical analyzer generator.
6. Explain the automatic generation of lexical analyzer.
7. Explain the term token, lexeme and pattern.
8. What are the various LEX actions that are used in LEX programming ?
9. Describe grammar.
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 ?