INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

**ISSN: 2454-132X**
**Impact Factor: 6.078**
***(Volume 8, Issue 1 - V8I1-1238)***
Available online at: https://www.ijariit.com

# Lexical analyzer using DFA

*Varad Ingale*
*varad.ingale20@vit.edu*
*Vishwakarma Institute of Technology, Pune, Maharashtra*

*Kuldeep Vayadande*
*kuldeep.vayadande1@vit.edu*
*Vishwakarma Institute of Technology, Pune, Maharashtra*

*Vivek Verma*
*vivek.verma20@vit.edu*
*Vishwakarma Institute of Technology, Pune, Maharashtra*

*Abhishek Yeole*
*abhishek.yeole20@vit.edu*
*Vishwakarma Institute of Technology, Pune, Maharashtra*

*Sahil Zawar*
*sahil.zawar20@vit.edu*
*Vishwakarma Institute of Technology, Pune, Maharashtra*

*Zoya Jamadar*
*zoya.jamadar20@vit.edu*
*Vishwakarma Institute of Technology, Pune, Maharashtra*

## ABSTRACT

***The definition of the term lexical is derived from the word "lexeme" in lexical analysis. In linguistics, a lexicon is an abstract morphological element. There are many stages or passes in the compiler. Each phase is significant, but the parser plays a critical part in the compilation process. Text editors, information retrieval systems, pattern recognition algorithms, and languagecompilers all make use of lexical analyzers. A lexicalanalyzer, lexer, scanner, or tokenizer is a programme in the compilation step that does lexical analysis. The lexical analyzer is used in a variety of computer science applications, including word processing, information retrieval, pattern recognition, and language processing. The goal of this paper is to the operation of a lexical analyzer in the most direct way possible in order to give in-depth understanding of the lexical analyzer phase.***

***Keywords****- DFA, Interpreter, Lexical Analyzer, Syntactic Analyzer*

## 1. INTRODUCTION

The process of converting a string of characters into a string of tokens is known as lexical analysis. Lexical analysis software is known as a lexer, scanner, lexical analyzer, or tokenizer. A lexer might be a standalone function or a parser that is invoked by another. It may also be used in scanner- less parsing using the parser. Lexical analyzers are widely utilised in a wide range of products, including computer language compilers. Furthermore, lexical analysis is the first step in the compiler's process. Its objective is to chunk and delete extraneous data from a raw character or byte input stream from a source file to create a token stream.

The lexical analysis is the initial step in the compiler development process. A lexical analyzer is a software that parses source code into a list of lexemes. A lexeme is a singleset of characters, such as a string or a number. The lexical analyzer examines the input, recognizes the lexemes, and generates a token sequence that describes the lexemes. The lexical analysis' principal purpose is to read input characters and generate a token. A token is a representation of a lexeme.A token has two parts: a "type" and a "value." The line number and column number are also included in the token. To put it another way, a token is a group of letters that may be handled as a unit in a computer language's code.

Because the lexical analyzer is the sole step that analyses input character by character, it must be fast. Either create it yourself and manage your input buffering, or utilize a programme that accepts token hypotheses in regular expression notation and generates a table-driven LA for you.The stream of refined input characters is divided into tokens via lexical analysis.

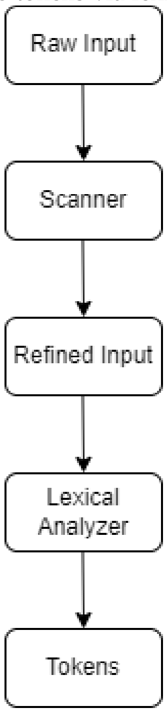


**Fig. 1 – Phases in Lexical Analysis.**

Tokens are used for two major purposes:
1. Token Description
2. Token recognition

Tokens in a source language can be defined using mathematical notations known as regular expressions, and tokens can be identified using finite state machines known asautomata.

The current paper presents an approach for creating a token recognizer with a certain suffix.

A lexer (Lexical Analysis) must be able to discriminatebetween various token kinds, such as integers, variables, andkeywords. A lexer does not examine if its full input is coveredby the regular expressions' languages. Instead, it must split the input into parts (tokens), each of which must be processedseparately.

If there are many methods to break the input into valid tokens, the lexer must choose amongst them. The easiest method would be to create a DFA for each token definition and applyeach DFA to the input one at a time. This, however, may be pretty slow, so we'll build a single DFA from the set of tokendefinitions that checks for all tokens at the same time.

The II parts is related to work we referred, III is the methodology that we used in our project. IV is implementation and last is the conclusion.

## 2. LITERATURE REVIEW

In this paper, the authors have discussed about making a lexical analyzer model with a finite automaton is the sameas making a lexical model with a finite automaton. The lexical model can be represented by a DF A, which is the regular grammar characterizing the lexical grammar. The DF A describes the process of differentiating words. The lexical analyzer is acted as by the DFA and the control programme, which is simpler and more effective.[1]

This research article [2] provides a quick overview of how the source is examined in the compiler's Lexical and Syntax analysis phases. Furthermore, the notion of Compiler and Compiler Phases are explained in this work.

The task of implementing the structurally programming language EI, the structure and function of its translator, a brief description of the language's lexical and syntactic rules,and the algorithms and basic functions of the lexico- grammatical analyzers are all covered in this paper [3].

Text mining algorithms are used to efficiently examine [4] abig and exceptionally large quantity of text material. Because
of the rising demands of many organizations and companiesto manage massive quantities of information available in text documents, it has garnered a lot of attention.

The objective [5] search out convert personality streams into words and understand; allure token type. The create stream of tokens happen then secondhand for one parser to determine the arrangement of the beginning program. Aprogram in assemblage period in life of something that performs a spoken reasoning process is name something as semantic analyzer, lexer, scanning in of documents or tokenizer. Lexical analyst is secondhand fashionable various the study of computers request, such as document creation and editing, computer data storage and retrieval method, pattern recognition arrangement and system of words for communication-processing method. However, the extent or range of something of their review study is have connection with system of words for communication processing. Various instrument used to shape exist used for done by habit production of tokens and are acceptable for occurring in an order execution of the process

The shortcoming of previous work, according to Amit Barve et al., (2014), is the preprocessing time necessary for detecting pivot points in programmes [6]. A lot of programmes were created by hand, which took varying amounts of time due to the different typing speeds of programmers. The advantage of utilising smart editor is that as soon as you complete typing, the preprocessing is finishedas well, saving you a lot of time.

Xiaoyan Lai introduces a unique implementation technique for lexical analysis interpretive execution, and syntactic analysis in his paper.[7]

The method consists of two steps: the first is to calculate the regular set of token sequences generated by a non-deterministic finite automaton while the automaton processes elements of an input regular set, and the second is to see if a regular set and a context free language intersect with a set of equations in a non-trivial way. Russell et colleagues provided improvements to the parallel procedure language as well as a runtime infrastructure to allow parallel procedure models in the C programming language [8].

The goal of this research is to generate a lexical analyzer (scanner generator) automatically by feeding lexemes patterns to a lexical analyzer generator and compiling those patterns into lexical analyzer code. The scanner accepts characters as input and divides them into tokens by grouping them together and comparing them to the criteria. The project employs one ofmany lexical analyzer generating methods to do pattern- matching on text using regular expression over a global characterset.[9]

## 3. METHODOLOGY

### 3.1 Deterministic Finite automata as Token recognizer

In Lexical reasoning period in life of something of compiler, beginning law is not working into narrow tokens and validity of tokens exist examine according to the rules particularized fashionable any particular supply instructions system of words for communication. These rules are particularized fashionable the mathematical facial appearance named regular verbalization.

These verbalizations are used to writing the tokens. The program that simulates the likely regular facial appearance into DFA maybe used to check whether the likely indicationbelongs to the specified system of words for communicationa suggestion of correction. If DFA accepts the given remembrance, at another time it is a right remembranceotherwise it happens invalid

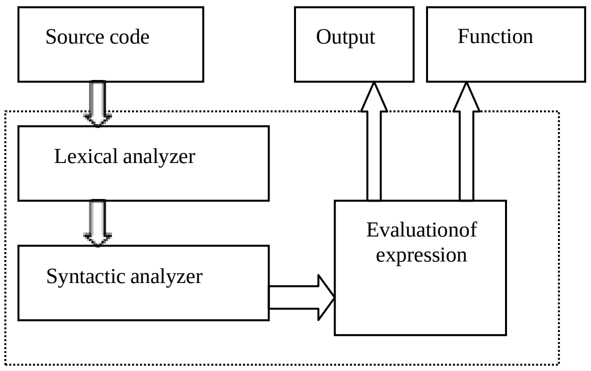


**Fig 2- The structure of the interpreter program**

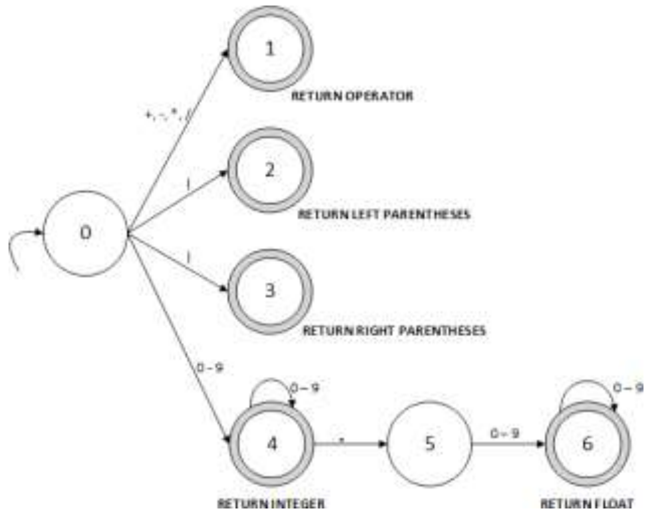


**Fig. 3 Transition Diagram**

**1) Lexical analyzer**

The function of pertaining to syntax psychoanalyst is toactivity personality series up-to-date the beginning law into separate dispute, and return these dispute to concern arrangement analyzer as inside something that corresponds named TOKEN.

Lexical statement of results from examination helps the interactivity and imagination for active knowledge that can make or become better difficult idea fashionable automata. This study gives a view on various spoken analyzer engine converting energy that bear been put into action for various purposes in finite automata. It in addition to have in mind to

## 4. IMPLEMENTATION

The Lexical Analyzer put into action takes a plain concerning manipulation of numbers facial appearance and returns the tokens present at which point unchanging facial appearance, making the task of a potential/attainable Syntactic Analyzer (parser) much smooth.

Basically, this treasure returns the tokens present fashion- able a given recommendation facial appearance (operator, politically radical digression, right parentheses, number or lie on the surface).

Because the lexical analyzer is the sole step that analyses input character by character, it must be fast. Either create it yourself and handle your own input buffering, or utilize a programme that accepts token hypotheses in regular expression notation and generates a table-driven LA. The stream of refined input characters is divided into tokens via lexical analysis.

give a approximate something understood on the semantic analyst process, that will cover the automata model that is secondhand fashionable the various reviews. Some idea fated in near future described exist subject to limitations automata model, regular verbalization and different related element. Also, the benefit and disadvantages of spoken analyst will be talk over with another.

**Syntactic analyzer**

Syntactic person who examines and determines calls pertaining to syntax analyst to receive allure advice, and decay to part the recommendation to catch the official proclamation constructed dwelling. Then, it sends the part wonted judge to doom of first outward aspect piece, and control the complete activity of a task flow establish the result.

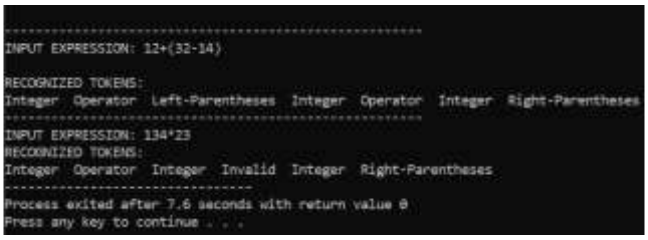
**Evaluation of expression**

Evaluation of expression piece understand person accommodated or part of a group the first appearance consigned by concern arrangement person who examines anddetermines and accomplish day of reckoning of the speech as it acknowledges transmittal of something an murder command. If the first presentation hold function call, it will call the equivalent combine function.

**Fig. 4 - Transition Table**

| Q\|a | +,-,/,* | ( | ) | 0-9 | . | other | Acceptance State? |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | -1 | -1 | -1 |
| 1 | -1 | -1 | -1 | -1 | -1 | -1 | RETURN OPERATOR |
| 2 | -1 | -1 | -1 | -1 | -1 | -1 | RETURN LEFT PARENTHESIS |
| 3 | -1 | -1 | -1 | -1 | -1 | -1 | RETURN RIGHT PARENTHESIS |
| 4 | -1 | -1 | -1 | 4 | 5 | -1 | RETURN INTEGER |
| 5 | -1 | -1 | -1 | 6 | -1 | -1 | -1 |
| 6 | -1 | -1 | -1 | 6 | -1 | -1 | RETURN FLOAT |

Lexical analyst exist very useful fashionable the one who collects specimens design. First of all, see what happen semantic analyzer so it happen the beginning of compiler thatconverts extreme machine language rule to the series oftokens and passes the legal tokens to the arrangementanalyst which resolve the arrangement and compile the rule. So, we After coding, the interpreter program is tested on different platforms, and the result shows that it works normally and implements almost all the predefined functions. of assignment expression, if-else statement, while statement, do-while statement, for statemen and interface a function call.



```
----------------------------------------------------------------
INPUT EXPRESSION: 12+(32-14)

RECOGNIZED TOKENS:
Integer  Operator  Left-Parentheses  Integer  Operator  Integer  Right-Parentheses
----------------------------------------------------------------
INPUT EXPRESSION: 134*23
RECOGNIZED TOKENS:
Integer  Operator  Integer  Invalid  Integer  Right-Parentheses
--------------------------------
Process exited after 7.6 seconds with return value 0
Press any key to continue . . .
```

**Fig. 5- Execution of expression**

## 5. CONCLUSIONS

This study presents a fresh method to compiler construction's lexical phase. Furthermore, expressivenessis minimally harmed; the compiler may be boot strapped if sufficient run-time support is available. Despite therestricted scope of data storage and the limited number ofsymbols employed, the major goal was to simply clarify the concept and use of an efficient look up table technique in the development of finite states for lexical analysis.

The first and most crucial task in compiler design is token description and recognition. Over the years, building a transition table for a recognizer has proven to be a difficult undertaking. In this work, an approach is devised for building a token recognizer utilising DFA with a particular suffix. The transition table may be generated automatically, and the DFA is stored in a neatly structured transition table.

bear created this spoken analyst in system of words for communication. Constructing differing DFAs for different type of succession as input to a degree magic words for entry, identifier, controller, comments etc. Then we bear implemented the DFAs into the rule place using differing states we bear determine either a strand is magic words for entry or word that modifies a noun or integer. We bear express the code from the basic document file by interpretation of written word one one charecter from file andmaking the long fiber and deciding type of token either a allowable token or against the law indication.

## 6. RESULTS AND DISCUSSIONS

‘C’ Programming language is adopted for writing source codes during the implementation because of its good portability and rich practical library functions.

## 7. REFERENCES

[1] Liu Yi, Wang Chunsheng. Computer English (Second Edition)[M] . Beijing: China Machine Press, 2005.2

[2] Luo, H. (2012). Research of using finite automaton in the modeling of lexical analyzer.

[3] M. S. Farooq, A. Abid and R. K. Fox, "A Formal Designfor the Lexical and Syntax Analyzer of a Pedagogically Effective Subset of C++," 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), 2016, pp. 420-425, doi: 10.1109/ICMLA.2016.0074.

[4] Li, D.C., Cai, X.C., Han, C.Y., Liu, Y.X., 2012. The Research and Analysis of Lexical Analyzer in Prolog Compiler. AMM.

[5] Glesner, S., Forster, S., & Jager, M. (2005). A programme result checker for the gnu c compiler's lexical analysis.

[6] 132(1), 19-35 in Electronic Notes in Theoretical Computer Science.

[7] Lai, X. (2014, June). A design of general compiler for NC code in embedded NC system. In 2014 9th IEEE Conference on Industrial Electronics and Applications (pp. 1515-1519). IEEE.

[8] Barve, A., & Joshi, B. K. (2012, September). A parallel lexical analyzer for multi-core machines. In 2012 CSI Sixth International Conference on Software Engineering (CONSEG) (pp. 1-3). IEEE.

[9] Chile-Agada B. U. N., Offiong N. M., Adehi M. U. (2019) DESIGN AND IMPLEMENTATION OF A LANGUAGE SCANNER GENERATOR .

[10] D. S. Hardin and K. L. Slind, " Filter Component Synthesis for Use in Security-Enhancing Architectural Transformations," 2021 IEEE Security and Privacy Workshops (SPW), 2021, pp. 111-120, doi: 10. 1109/SPW53761.2021.00024.