DHA SUFFA UNIVERSITY

Language Specification Document

Compiler Construction

CS- 3102

Group Members:

Muhammad Hamza Ali Khan | cs182044 (Group Leader)

Arham Shams | cs182024

Usman Yousuf | cs182037

Razor:

The language razor is based upon the c++ language syntax, it is stationarily typed and is capable of supporting manual memory management. It is capable of compiling with the help of external libraries.

Lexical Analyzer:

The lexical analysis is truly independent of the syntax parsing and the semantic analysis. The lexical analyzer splits the source text up into tokens. The grammar is designed to be suitable for high-speed scanning and to make it easy to write a correct scanner for it. It has a minimum of special case rules. An identifier is a gathering of letters, digits, and highlights that starts with a letter. The case is colossal.

Whitespace (spaces, tabs, newlines, returns, and structure takes care of) or then again, comments may appear among tokens and are dismissed. A remark begins with/* and closes with */. Remarks might be settled.

An entire number predictable is a plan of at any rate one decimal digit (i.e., 0123456789). There are no bad entire number constants; negative numbers may be gotten by invalidating an entire number reliable using the unary - administrator.

A string predictable is a gathering of at any rate zero printable characters, spaces, or flight progressions enveloped by twofold proclamations. Every escape game plan starts with a sideways accentuation line \ and represents some course of action of characters.

Rules for identifier:

1. Names can contain letters, digits and underscores.
   For example: var_1
2. Names must begin with a letter.
3. Names are case sensitive, which means that if one person write, var_1 and Var_1 then there's the difference between the two.
4. Names cannot contain whitespaces or special characters. For example, if a person uses their variable with spaces and special it will not be accepted as identifier.

5. Keyword cannot be used as identifiers. As mentioned in the last point, the characteristics will become that of a keyword if we use spaces and special characters.

Keywords:

1. “if”
2. “case”
3. “switch”
4. “else”
5. “while”
6. “then”
7. “fi”
8. “do”

Datatypes:

1. Int
2. char
3. string
4. float

Separators:

1. “(“
2. “)”
3. “[“
4. “]”

Operators:

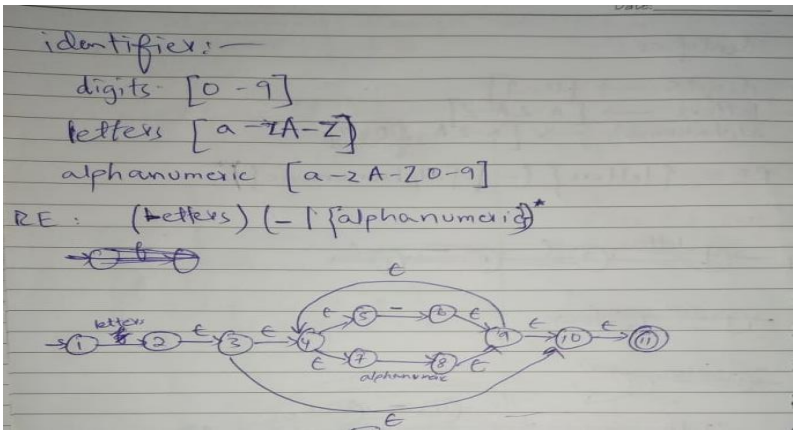
The Binary operators are + - */:= <> < > <= >= and | Parenthesis bundle enunciations in the standard way. A principle short sign dishonors a number explanation. The parallel operators+, - , *, and/require entire number operands moreover, return an entire number result. The double operators>, <, >=, and <= take a gander at their operands, which may be either both number or both string and produce the number 1 if the connection holds and 0 regardless. String the assessment is done using a run of the mill ASCII lexicographic solicitation. The double operators= and <> can break down any two operands of the same (non-pointless) type and return either number 0 or 1. Numbers are the comparable if they

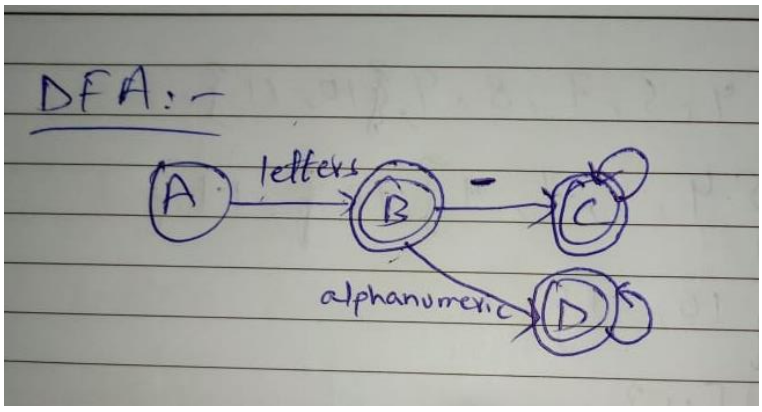
have a comparative worth. Strings are something similar if they contain comparative characters.

1. “=” : Equals to
2. “!=”: not Equals to
3. “<” : Less than
4. “<=” : Less than or equal to
5. “>” : Greater Than
6. “>=”: Greater than or equals to
7. “+” : Addition
8. “-“ : Subtraction
9. “*” : Multiplication.
10. “/” : Division
11. “(“ : Left Parenthesis
12. “)” : Right Parenthesis
13. “:=”: Assignment
14. “;” : Terminator

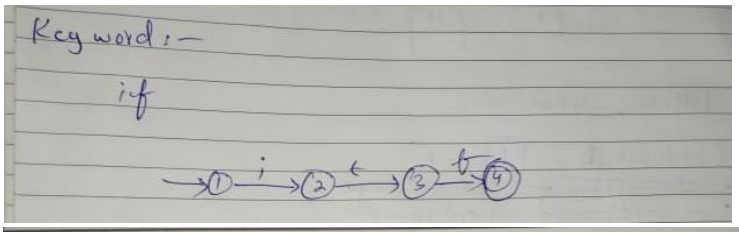
DFA of Some Terminologies:

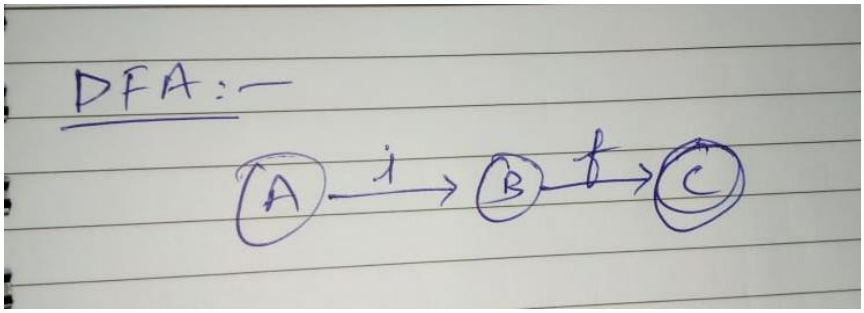
- Identifier



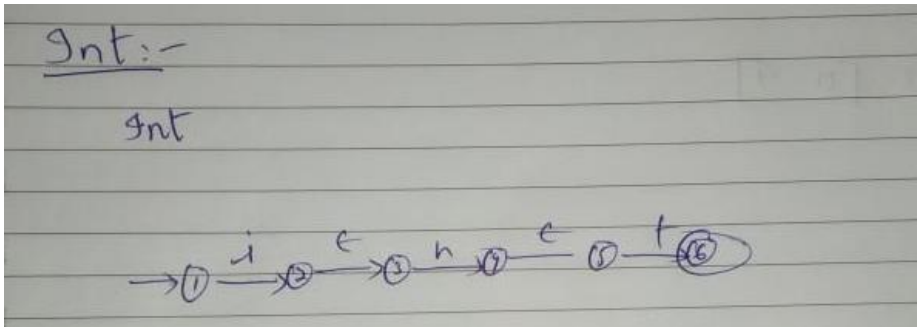


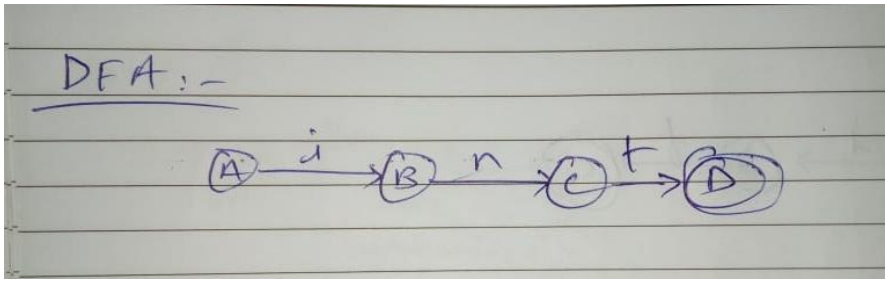
- Keyword: If





- Int:





- Digits: