COMP3438 Assignment 2

JAHJA Darwin, 16094501d

---

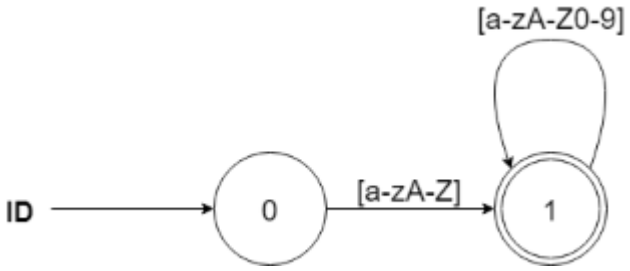
1. Regex

```
KEYWORD   -> var|begin|end
COMMA     -> ,
SEMICOLON -> ;
ASSIGN    -> =
PERIOD    -> \.
NUM       -> [0-9]*(\.([0-9]+))?
PLUS      -> +
MINUS     -> -
MUL       -> *
DIV       -> /
LBRACE    -> (
RBRACE    -> )
ID        -> [a-zA-Z][a-zA-Z0-9]*
```

2. Finite Automata (FA)

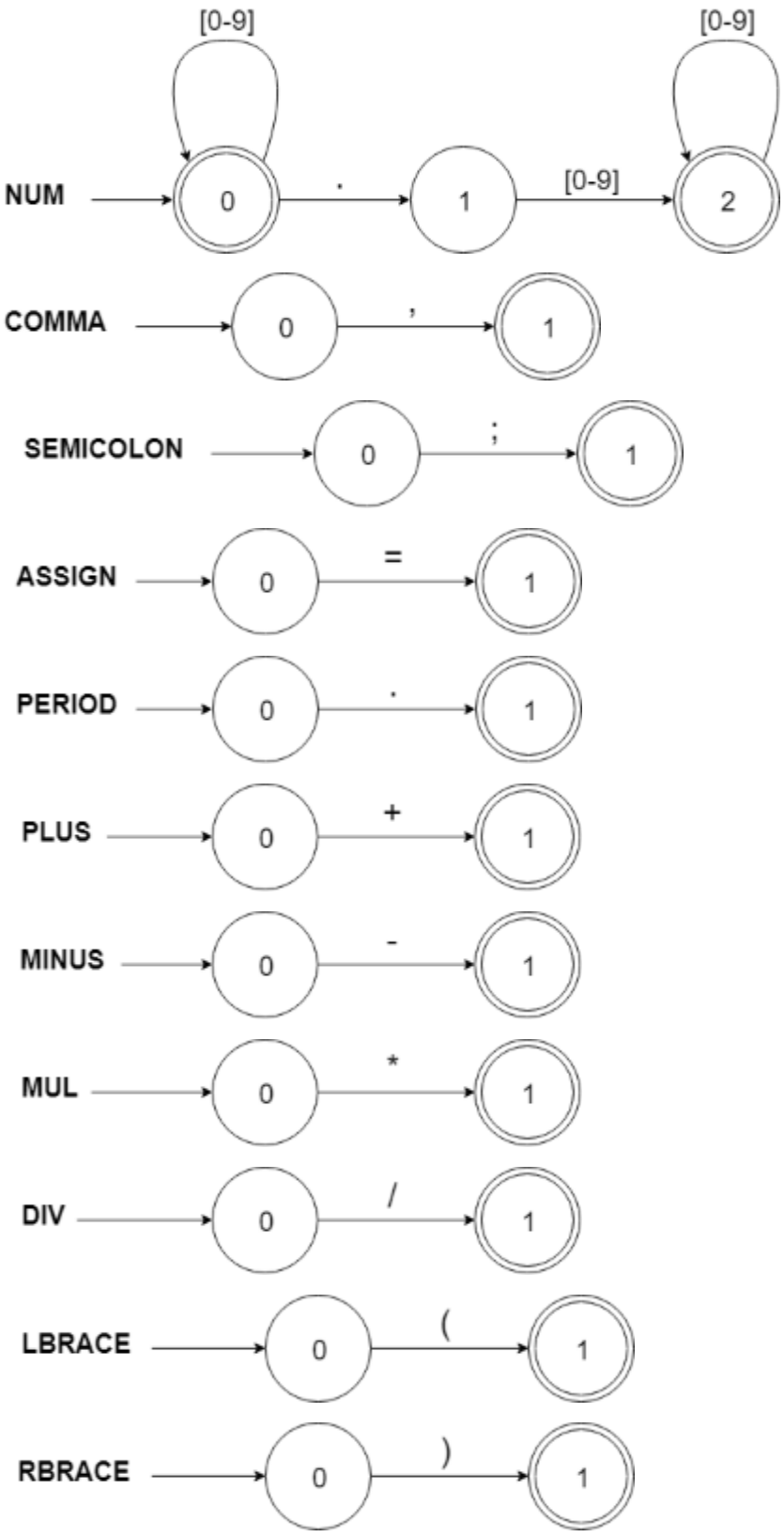
The following constructed minimized-FAs are used to identify each token type:

KEYWORD & ID



A keyword is identifies by the above FA as an ID. Then, after an ID is obtained, the program will check a keyword table to see whether it is an ID or keyword.

In this way, the number of states can be reduced.


[0-9]
[0-9]
NUM
0
.
1
[0-9]
2
COMMA
0
,
1
SEMICOLON
0
;
1
ASSIGN
0
=
1
PERIOD
0
.
1
PLUS
0
+
1
MINUS
0
-
1
MUL
0
*
1
DIV
0
/
1
LBRACE
0
(
1
RBRACE
0
)
1

3. Program

Functions & Explanation

`main()`: The main function of the lexical analyzer

1. Load the file that needs to be analyzed
2. Run lexical analysis

`lexAnalyze()`: Read characters one by one and identify token category.

List of Category

1. KEYWORD, ID:

   `checkKeyId()` will be run if the character is alphabet.

2. NUM:

   `checkKeyId()` will be run if the character is digit.

3. ASSIGN, ADD, MINUS, MUL, DIV:

   Further check in `checkOperator()`.

4. SEMICOLON, COMMA, PERIOD, LBRACE, RBRACE:

   Further check in `checkSeparator()`.

`checkKeyId()`: Process the value and further check whether it is KEYWORD or ID

1. KEYWORD is identified by a keyword table.
2. Otherwise, it will be labeled as ID.

`checkNum()`: Process the value of NUM

`checkOperator()`: Further identify the operator type using switch statement

`checkSeparator()`: Further identify the separator type using switch statement

4. Procedures

Run the following commands to compile and run the program:

```
gcc -o lex_analyzer lex_analyzer.c
./lex_analyzer testme.txt
```

To test the program with different test file, simply change the 1st argument. Two pre-made test files, `testsample.txt` and `testme.txt`, can be used for testing.

```
# lex_analyzer your-file-name
./lex_analyzer testsample.txt
```