(1)

1. Type = { INT, CHAR }

2. integer = { ... -1, 0, 1 ... }

3. Literal String = { "Hello world", ... }

4 An identifier of variables and functions:
{ i, j, k, ab_12, ... }

5. Keywords for special statement: { IF, ELSE, WHILE, RETURN }

6 Arithmatic operators: { +, -, *, / }

7 Assignment operator: { = }

8 Comparison operator: { <, >, ==, !=, <=, >= }

9 A terminating symbol of statements: { ; }

10 A pair of symbols for defining area/scope of variables and function: { {, } }

11 A pair of indicating a function: { (,) }

12 A symbol for sperating input arguments in functions: { , }

13 Whitespace: \t \n " "

14 Comments: { "are not allowed" }

① TOKEN

TYPE int | char

ALPABET_S a|b|c|d|... |z

DIGIT 0|1|2|3|4|5|6|7|8|9

ALPABET_M A|B|C|D ... |Z

KEYWORLD if| else| return | while

OPERATOR +|-|*|/

ASSIGNMENT =

COMPARISON <|>|==| !=|<= |>=

SEMI ;

LBRACE {

RBRACE }

LPAREN (

RPAREN)

COMMA ,

WHITESPACE ~~\t~~ |~~\n~~ | " "

② Regular expression

Integer `(-)*[1-9][0-9]* | 0)`

Literal_String `" ([A-Z]*/[a-z]*/[" "]*)* "`

Identifier

`[a-z]+[0-9,a-z]*`

Arithmetic operator

`+ | - | * | /`

Comparison operator

`< | > | == | != | <= | >=`

Lparen (

Rparen)

Lbrace {

Rbrace }

Semi ;

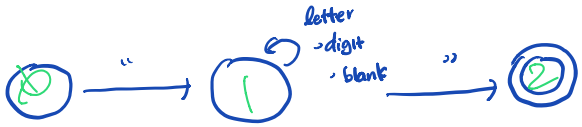
Whit space `\t | \n | ' '`

Keyword `int | char | INT | CHAR`

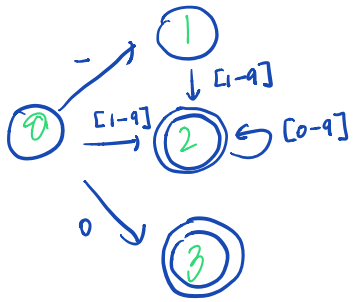
DFA

(2)

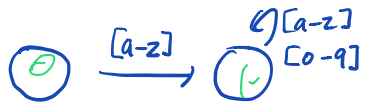
Literal String



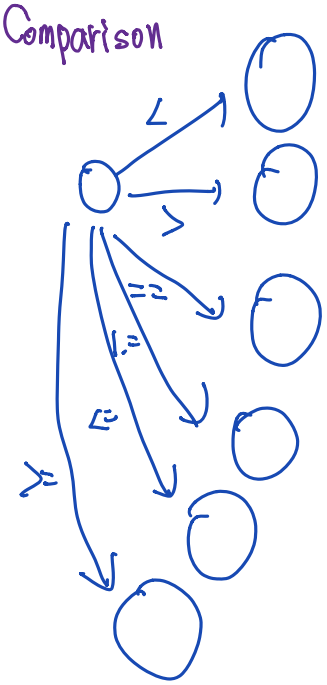
Integer



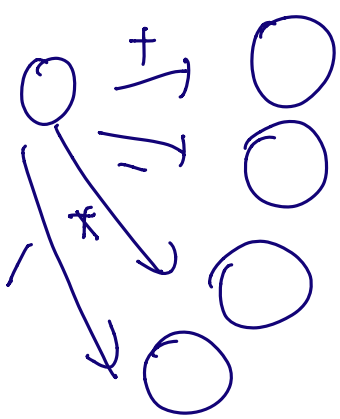
Identifier



Comparison



Arithmetic operation



key-word