Tiny language

TeamID 7

Amr Hany:	2023170410
Jana Waleed:	2023170166
Mariam Hassan:	2023170582
Ahmed Atta:	2023170052
Mariam Sherif:	2023170589
Farah Younis:	2023170426

Token Classes

1-Reserved Keywords

Lexeme	Token Class
int	Int
float	Float
read	Read
write	Write
repeat	Repeat
until	Until
if	If
elseif	ElseIf
else	Else
then	Then
endl	Endl
string	String
return	Return

2-Logical Operators

Lexeme	**Token**
&&	AndOp
||	OrOp

3-Brackets & Symbols

Lexeme	**Token**
;	Semicolon
,	Comma
(	LParanthesis
)	Rparanthesis
{	LCurl
}	RCurl

4-Comparison & Assignment Operators

Lexeme	Token Class
=	EqualOp
:=	AssignmentOp
<	LessThanOp
>	GreaterThanOp
<>	NotEqualOp

5-Arithmetic Operators

Lexeme	Token Class
+	PlusOp
-	MinusOp
*	MultiplyOp
/	DivideOp

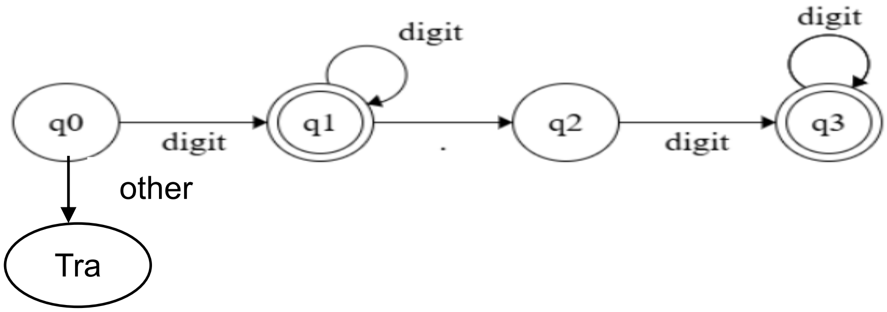
6-String, comment, number, identifier

category	Token class	Description
“String”	StringElem	Any text between two double Quotes
Identifier	Identifier	Start with letter followed by letters or digits
/*comment*/	Comment	Any text between /* and */
Number	Number	Integer, float

DFA

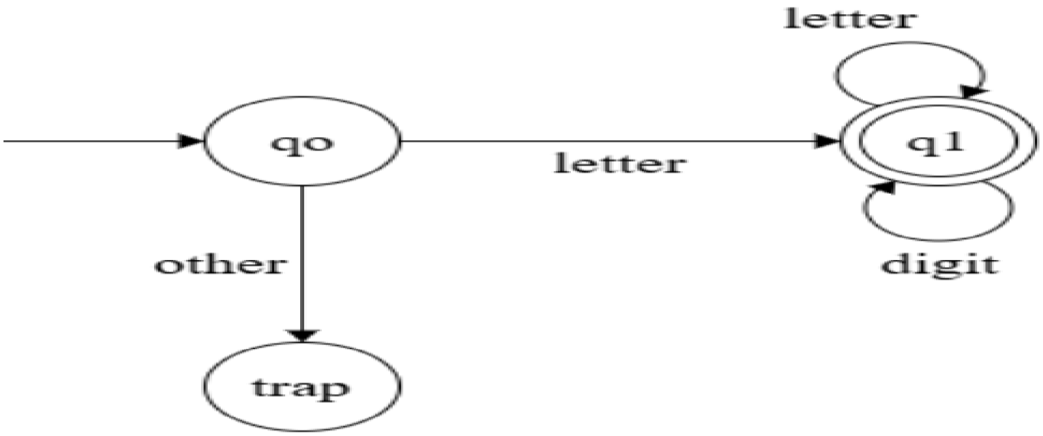
1-Number

Regex: digit+(/.digit+)?



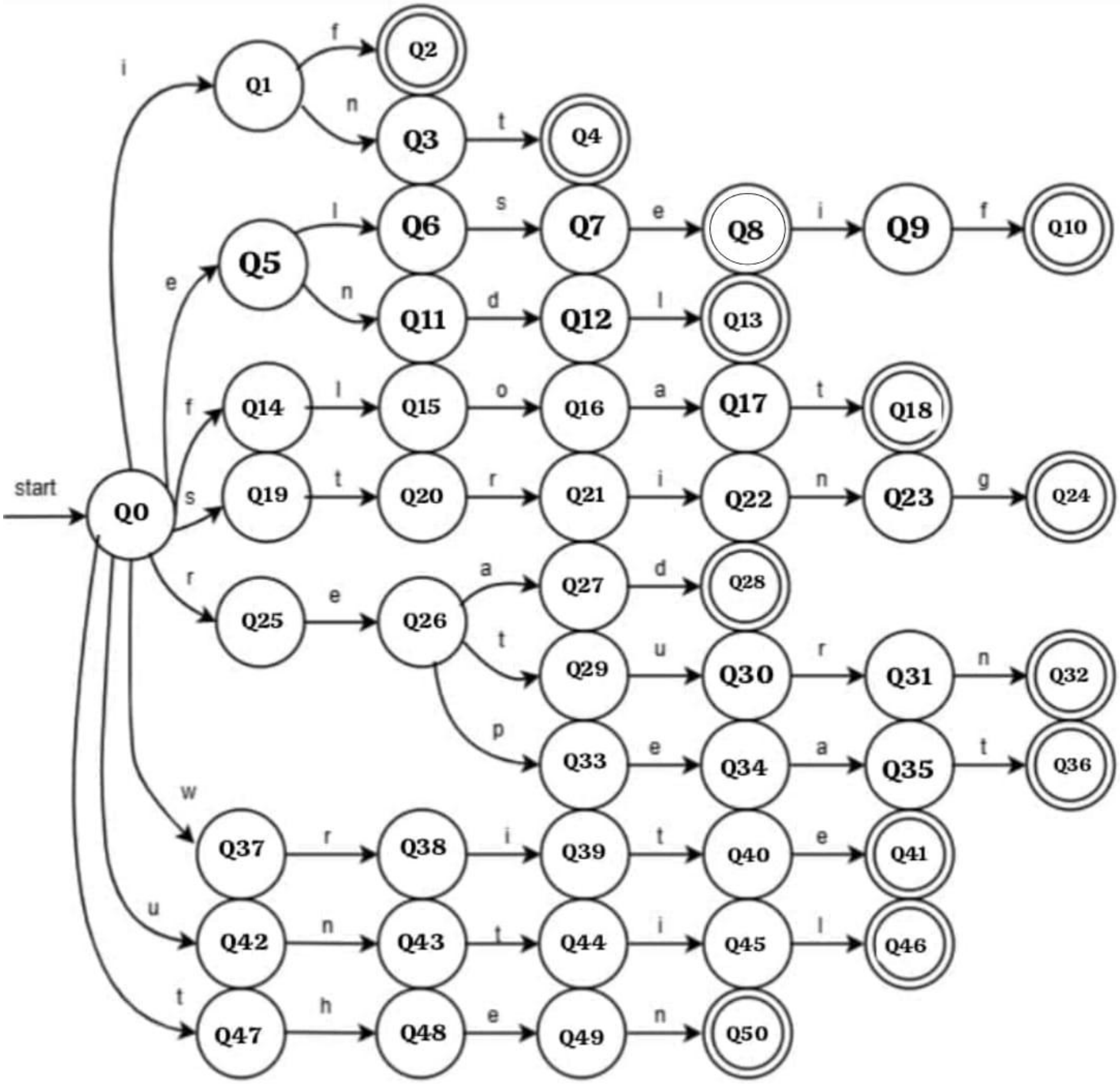
2-Identifier

Regex: letter(letter|digit)*



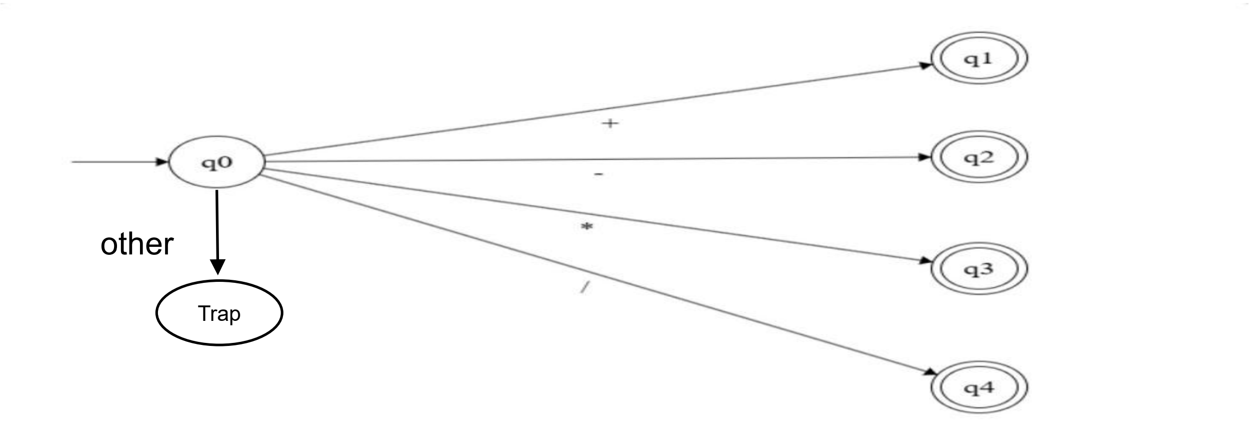
3-Reserved Keywords

Regex: Reserved_Keywords: /b int | float | string | read | write | repeat | until | if | elseif | else | then | return | endl /b



4-Arithmetic operators

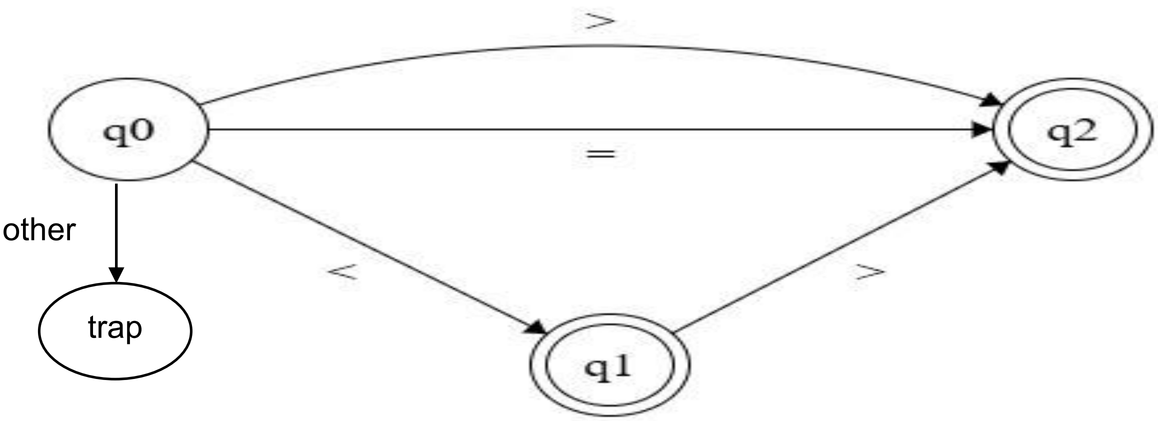
Regex: (+|-| *|/)



5-ConditionOperators

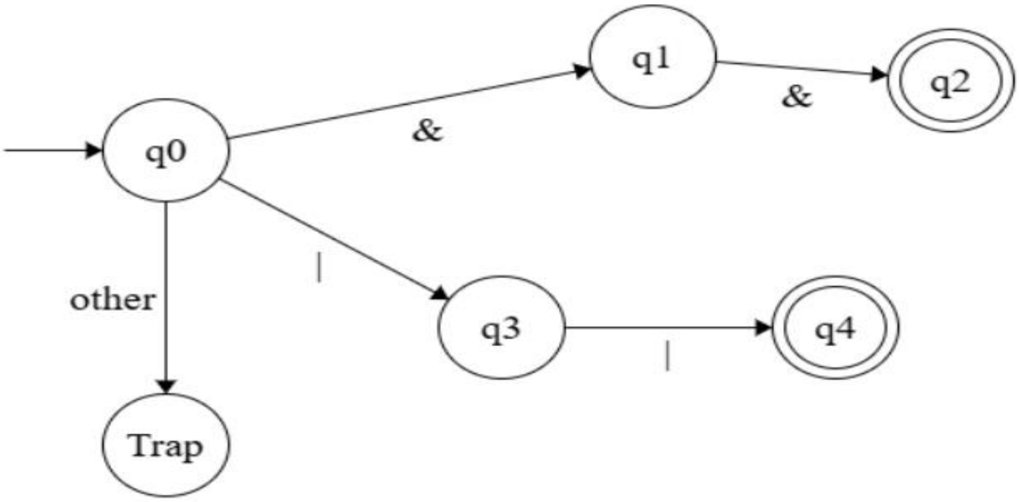
Regex: (< | > | = | <>)

DFA:



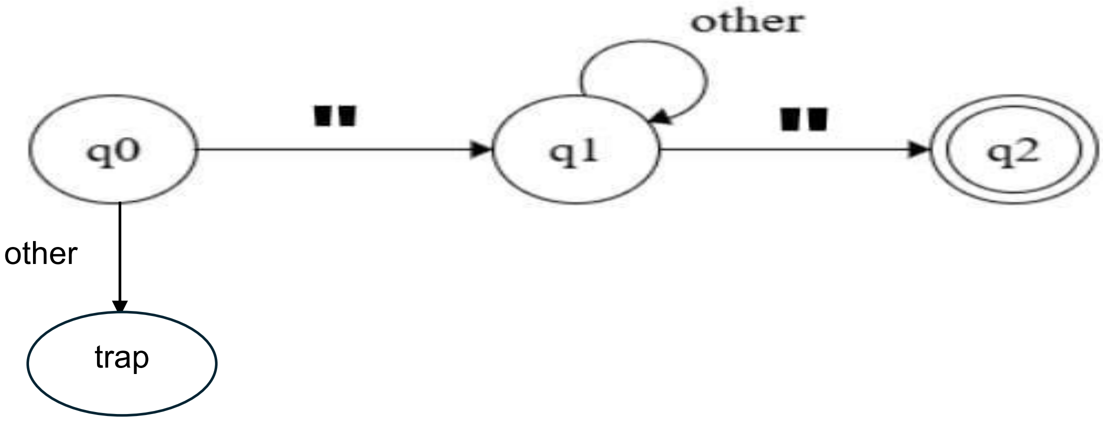
6-**Boolean_Operator**

Regex: (&& | ||)



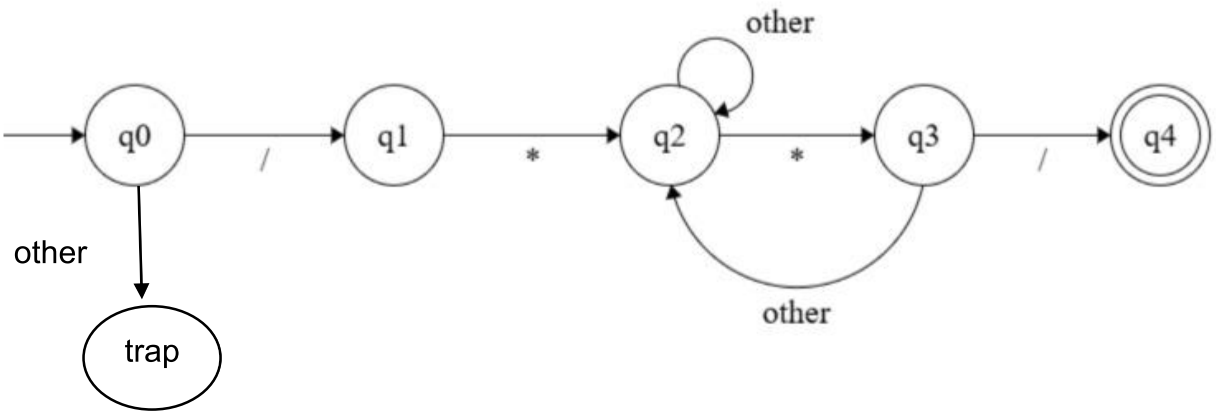
7-**String**

Regex: “ .* “



8-Comment

Regex: / * .* * /


other
q0
/
q1
*
q2
*
q3
/
q4
other
trap
other