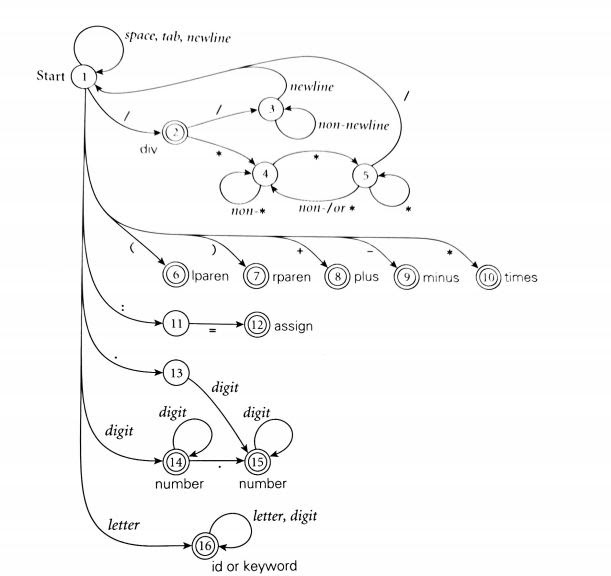
The DFA That is the base for the scan function



The data structures to be used in the scanner:

Arrays to store the inputs into then checking it to determine what everything needs to be outputted also used as a Queue to hold the input from the scanner to be later determined and It should provide an ease of eliminating commits

Pseudocode:

:

Function Name:Main

   Data

Given

File name:the file that is used to scan

Unknown

Input:the data in the file

FileLength:how many chars in the file

Output:the output

Intermediate

FileLength: how long the array that holds the file data is

SelectedString:string that is being processed

Selection: what switch statement to be selected

  Plan

Int,FileLength = 0,selection=0

String Output

while(nextchar != null)

FileLength++

Nextchar ++

if(FileLength==0)

Error

FileData[FileLength]

Initialize scanner

FileData[] = scanner(Filename)

while(!done)

SelectedString = “”

while(nextchar == null)

Nextchar ++

while(nextchar != null)

SelectedString + nextchar

Nextchar ++

If (SelectedString ==“Read” or SelectedString==“read”)

Selection =1

Else if(SelectedString==string)

Selection =2

Else if(SelectedString==number)

Selection =3

Else if(SelectedString==”+”)

Selection =4

Else if(SelectedString==”-”)

Selection =5

Else if(SelectedString==”\*”)

Selection =6

Else if(SelectedString==”/”)

Selection =7

Else if(SelectedString==”=”)

Selection =8

Else if(SelectedString==”(”)

Selection =9

Else if(SelectedString==”)”)

Selection =10

Else

error

Switch(switch)

Case 1:

Output +” read”

Break

Case 2:

Output +” id”

Break

Case 3:

Output +” number”

Break

Case 4:

Output +” addition”

Break

Case 5:

Output +” subtraction”

Break

Case 6:

Output +” multiplication”

Break

Case 7:

Output +” division”

Break

Case 8:

Output + “equal”

Break

Case 9:

Output + “LeftParentheses”

Break

Case 10:

Output + “RightParentheses”

Break

Defalt:

Error

Print output

Function Name:Scanner

  input

File name

   Data

Given

Unknown

FileLength:number of characters in the file

InputHold:the chars in the file

Intermediate

inCommit: check to see if inside a commit

counter : used to count through the array

Nextchar :what the char is that is currently selected

  Plan

Bool inCommit = False

Int counter = 0

Nextchar = first char from File name

FileLength = 0

while(nextchar != null)

FileLength++

Nextchar++

if(FileLength==0)

Error

make array InputHold length FileLength

Nextchat = first char from File name

while(counter <= FileLength)

InputHold[counter] = Nextchar

Nextchar++

counter++

//at this point there is an array with every char in the file put in it in order

//remove the commits TIME

while(counter <= FileLength)

if(InputHold[counter] == ‘/’ && InputHold[counter+1] == ‘\*’)

inCommit=True

if(InputHold[counter] == ‘\*’ && InputHold[counter+1] == ‘\’ && inCommit)

inCommit = False

InputHold[counter] = ‘ ’

InputHold[counter+1] = ‘ ’

Counter+2

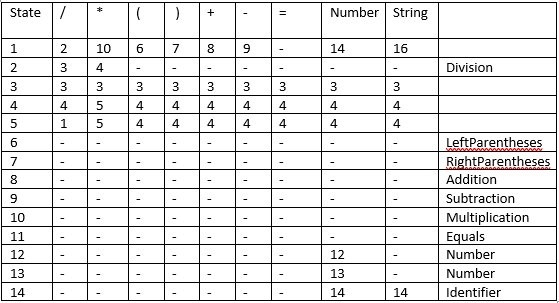
if(inCommit)

InputHold[counter] = ‘ ’

if(inCommit)

error

Return InputHold[]



Test cases:

Read for /\*jtgfjjkdjk omop3904t0gmn03 9g0jg0p\*/

23 - 65

54 +  read id2 /  32 \* 8

This is a test case to conform that everything is working

Another test case with nothing in it to check that the error checking works

  Fuifwh /\* ujefi02nfkslnfn029j f2o02nmf

Tests case to check that the error checking for end of commits works

3325m20(df

Test case to see if check that errors happen for bad input

Contributors:

Jonathan Turner: did pseudocode,test case selection, data structure explanation

 Jose Trevino: did the table, pseudo code and document outline