Assignment 2 COL226 Submission Date: 29/02/2020

I wrote a **lexer.mll** file :

Format:

```
{ header }
let ident = regexp …
[refill { refill-handler }]
rule entrypoint [arg1… argn] =
  parse regexp { action }
      | …
      | regexp { action }
and entrypoint [arg1… argn] =
  parse …
and …
{ trailer }
```

Header section : In this section i declared the token names
Float -- Float numbers
Index -- Natuaral numbers
Lparan -- '('
Rparan -- ')'
Lbrac -- '['
Rbrac -- ']'
Comma -- ','
Colon -- ':'
Indice I -- **[** *i* **,** *j* **]**
Range R -- (I : I)
Uop - COUNT ,ROWCOUNT ,COLCOUNT ,SUM,ROWSUM ,COLSUM ,AVG ,ROWAVG ,COLAVG,MIN , ROWMIN ,COLMIN ,MAX ,ROWMAX ,COLMAX
Bop – ADD, SUBT, MULT, DIV
Assign -- ':='
Terminator -- ';'
Cons – Floating and natural
EOF – End Of File

and *two exceptions Eof and Ill.*

Eof- If we have any uncategaries string or regular expression then this exception simple terminates the execution of executable file.

Ill- If we have illegal natural num or floating num or index or range then this simple terminates the execution with a error warning.

Declaration : In this section I declared the regular expressions for
Natural Numbers(n) :- Doesn't accept zeros before first non-zero interger.
Floating Numbers(f) :- With optional sign, no redundant initial zeroes before the decimal point or unnecessary trailing zeroes after the decimal point.
Indices(i) :- Defines indices as defined in problem statement.
FalseIndices(x) :- Indices in which we give float number to i or j then it is a illegal index.
Range(r) :- Defines ranges as defined in problem statement.
FalseRanges(y) :- Ranges in which we give illegal index as input then it is a illegal range.
ConstantSet(c) :- Flaot or Natural
UnaryOperations(uop) :- Defines names of unary operations.

BinaryOperations(bop) :- Defines names of Binary operations.

At last by ruling the token by the parse I omitted the unnecessary blanks and tabs, assigned the appropriate sequence of character(s) and regular expressions as defined above to their respectetive token names declared in header section.

Trailer Section : In this I wrote the executing funtion main which reads the sequence of characters and prints “I saw a token!” for each successful token read and if an illegal expression is detected terminates the program with a warning.

Lokesh Acharya
2018CS10352