**Assignment 1**

CS323

Due (10/1 Sunday, Midnight (11:59pm), soft-copy)

(10/2 Monday, 5 pm, hard-copy)

The first assignment is to write a lexer(lexical analyzer).
**You can build your entire lexer using a FSM, Or build using at least FSMs for identifier, integer and real (the rest can be written ad-hoc)**

**BUT YOU HAVE TO CONSTRUCT A FSM for this assignment otherwise, there will be a deduction of 2 points!**

**Note: In your documentation (design section), YOU MUST write the REs for Identifiers, Real and Integer, and also show the NFSM using Thompson.**

**The Lexer**

A major component of your assignment will be to write a procedure (Function) – lexer () - that returns a token when it is needed. Your lexer() should return a record, one field for the token and another field the actual "value" of the token (lexeme), i.e. the instance of a token.

**Your main program should test the lexer i.e., your program should read a file containing the source code of Rat17F to generate tokens and write out the results to a file.**
Make sure that you print both, the tokens and lexemes.
Basically, your main program should work as follows

```
while not finished (i.e. not end of the source file) do
    call the lexer for a token
    print the token and lexeme
endwhile
```

**Do at least 3 test cases and make sure that you turn in proper documentation using the documentation template.**

**A simple Example**

**Source code**:

```
while (fahr#x  < upper)   a := 23.00
```

**Output:**

| **token** | **lexeme** |
|---|---|
| keyword | while |
| separator | ( |
| identifier | fahr#x |
| operator | < |
| identifier | upper |
| separator | ) |
| identifier | a |
| operator | := |
| real | 23.00 |