CS224 – Tokenizer

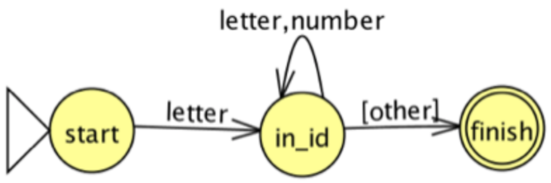
Purpose: The first step in a compiler is to chunk the characters in the program into a list of tokens. Each token is of a particular type like an identifier, number, string, or symbol.

Knowledge: This activity will help you become familiar with the following content knowledge:

- How to build a state diagram which describes a token
- How to mechanically construct code from a state diagram to recognize a token

Activity: With your group perform the following tasks and answer the questions. You will be reporting your answers back to the class in 45 minutes.

1. A state diagram describes a token. The following state diagram describes an `identifier` which must start with a letter and contains only letters and numbers. After a character is detected that is not a letter or a number, then that character is not consumed or contained in the token (as indicated by the square brackets) and we have the entire token.



Draw a single state diagram, with four different terminal states for each of the four tokens `<`, `<=`, `>=`, and `==`

2. The state diagram can be mechanically turned into code. The following incomplete code recognizes the `identifier` token from the state diagram above. This code checks which state we are in and then moves to a new state depending upon the input character.

   ```
   state = start;
   input = getNextChar()
   while (state != finish and state!=error) {
          switch (state) {
          case start:
                if (isalpha(input)) {
                       state = in_id;}
                break;
          case in_id:
          <COMPLETE CODE HERE>
          }
          input = getNextChar()
   }
   if (state == finish) return ID;
   else return ERROR;
   ```

   Complete this code.

3. Give a state diagram and then code for the tokens `*`, `*/`, `*//`, and `*/#`. Your code should return the type of token found.

4. Should we have state diagrams for reserved words like `if`, `while`, etc? What are the alternatives?

5. Give a state diagram (no code) for the Jack tokens Symbol, Identifier, Int_Const, String_Const. You can look up in the text what these tokens look like.

6. Add to the state diagram additional states for handling line comments // and block comments /* ... */. Make sure the state diagram still works for the symbol /
   (It looks like you are ready for the tokenizer project now!!)