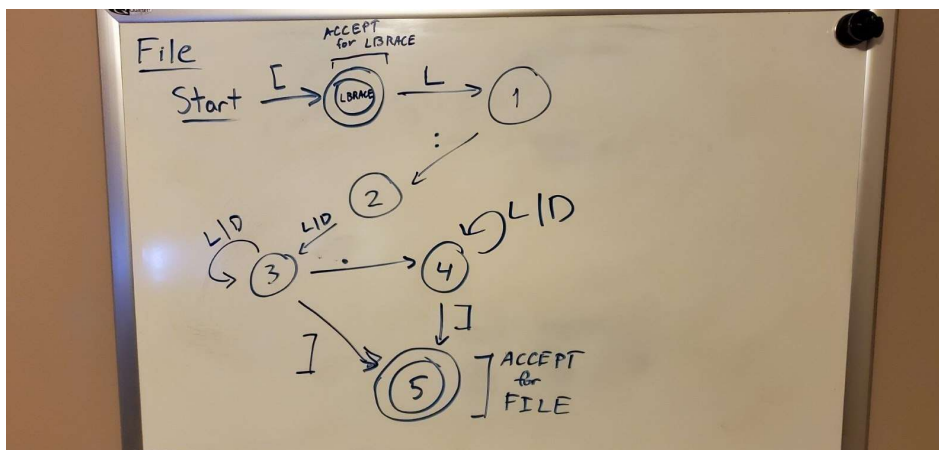
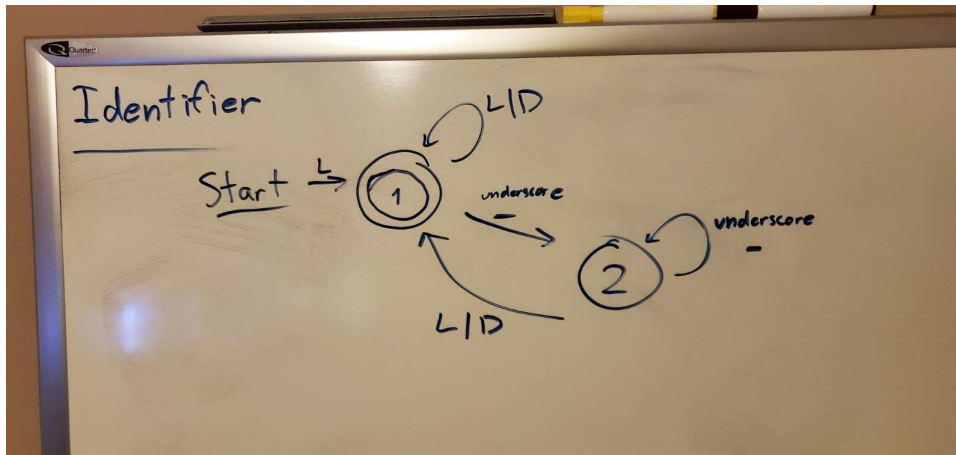
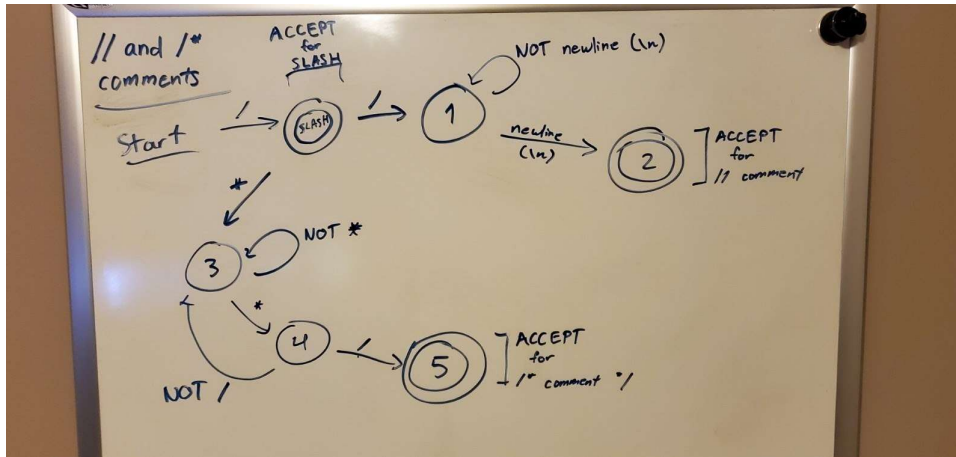
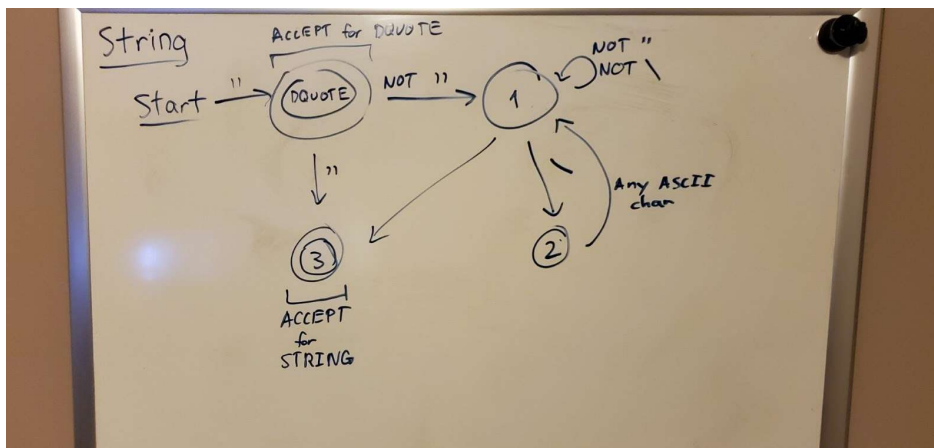
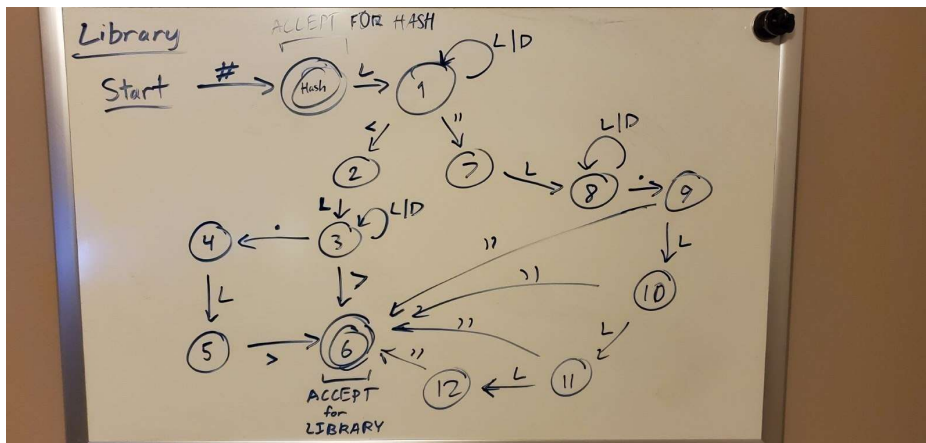
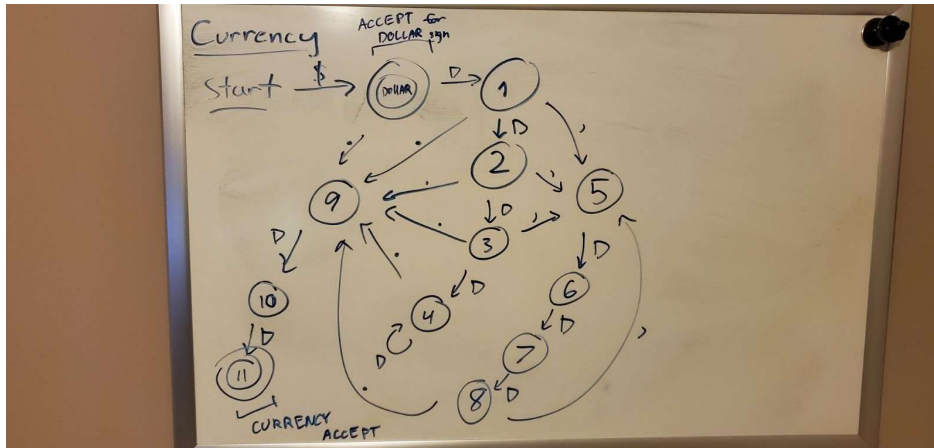
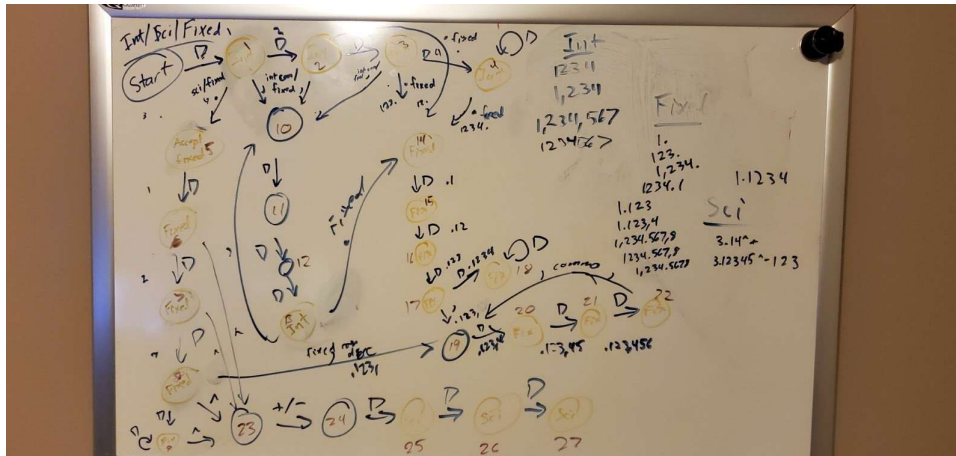


Finite-State Automata Images







## NOTE

The Full DFA is provided separate from this document.

## Screenshots of Scanner Log

Figure 1 – Scan of `big_test.txt` [from Professor Giacalone's scanner repository, `scanner/docs/`. No whitespace mode.

```
scan (1) x
"C:\Users\jchom\Documents\CSULB\Graduate\Fall 2021\CECS 444\Compiler\venv\Scripts\python.exe" "C:/Users/jchom/Documents/CSULB/Graduate/Fall 2021/CECS 444/Compiler/Project/big_test.txt"
Enter file path: C:\Users\jchom\Documents\CSULB\Graduate\Fall 2021\CECS 444\Compiler\Project\big_test.txt
Enter 1 to include white-space. Otherwise, no white-space scan
TOKEN: comment - //First are the single operators
TOKEN: grave - `
TOKEN: lbracket - <
TOKEN: rbracket - >
TOKEN: lbrace - {
TOKEN: rbrace - }
TOKEN: lsquig - {
TOKEN: rsquig - }
TOKEN: at - @
TOKEN: ampersand - &
TOKEN: hash - #
TOKEN: bang - !
TOKEN: tilde - ~
TOKEN: quote - '
TOKEN: dollar - $
TOKEN: colon - :
TOKEN: semicolon - ;
TOKEN: period - .
TOKEN: comma - ,
TOKEN: plus - +
TOKEN: minus - -
TOKEN: slash - /
TOKEN: star - *
TOKEN: equal - =
TOKEN: carat - ^
TOKEN: lparen - (
TOKEN: rparen - )
TOKEN: comment - /* Next are the compound operators */
TOKEN: ddash - --
TOKEN: dplus - ++
TOKEN: plusminus - +-
TOKEN: minusplus - -+
TOKEN: greaterreq - >=
TOKEN: lesseq - <=
TOKEN: noteq - !=
TOKEN: spaceshp - <>
TOKEN: eqeq - ==
TOKEN: coloneq - :=
TOKEN: dless - <<
TOKEN: slasheq - /=
TOKEN: dbang - !!
TOKEN: comment - //IDs
RESERVED WORD: this - IDENTIFIER
RESERVED WORD: this - IDENTIFIER
TOKEN: identifier - will
TOKEN: identifier - had
RESERVED WORD: this - IDENTIFIER
TOKEN: identifier - all
TOKEN: identifier - appear
TOKEN: identifier - appear
TOKEN: identifier - twice
TOKEN: identifier - appear
TOKEN: identifier - a
TOKEN: identifier - a
TOKEN: identifier - b
TOKEN: identifier - hal
TOKEN: identifier - a
TOKEN: identifier - a
TOKEN: identifier - a
RESERVED WORD: this - IDENTIFIER
TOKEN: identifier - cool
TOKEN: identifier - had
TOKEN: identifier - good
TOKEN: identifier - times
TOKEN: identifier - were
TOKEN: identifier - had
TOKEN: identifier - all
TOKEN: identifier - around
TOKEN: comment - { *Integers * }
TOKEN: rsquig - 1
TOKEN: integer - 1234
TOKEN: integer - 1,234
TOKEN: integer - 1,234,567
TOKEN: integer - -123
```

```

TOKEN:    integer - -1,234
TOKEN:    comment - //Currency
TOKEN:    currency - $555.55
TOKEN:    currency - $1234.56
TOKEN:    currency - $1,234.56
TOKEN:    comment - /* Library Tokens */
TOKEN:    library - #include"math.h"
TOKEN:    semicolon - ;
TOKEN:    library - #include"string.cpp"
TOKEN:    semicolon - ;
TOKEN:    library - #include<iostream.h>
TOKEN:    semicolon - ;
TOKEN:    comment - { * Fixed Point Numbers *}
TOKEN:    fixed - 3.141
TOKEN:    fixed - 777.
TOKEN:    fixed - 3.14159
TOKEN:    fixed - 1,234.567
TOKEN:    fixed - 1,234.5678
TOKEN:    fixed - 1,234.567,8
TOKEN:    comment - // Scientific Notation
TOKEN:    scinotation - 3.3^+1
TOKEN:    scinotation - 3.14159^-1
TOKEN:    scinotation - 3.14159^+123
TOKEN:    comment - // Device/File Tokens (begins with a reserved word)
RESERVED WORD: read - IDENTIFIER
TOKEN:    file - [A:file.txt]
TOKEN:    semicolon - ;
RESERVED WORD: write - IDENTIFIER
TOKEN:    file - [A:file9.txt]
TOKEN:    semicolon - ;
TOKEN:    comment - /* String Literals */
TOKEN:    string - "hey!!'"
TOKEN:    string - "this should' not * break"
TOKEN:    comment - /* This will
be a *
multi-line //
comment * with
stuff * that would
break it if it /
were coded differently
*/
TOKEN:    comment - { *This is the end comment*}

Process finished with exit code 0

```

Figure 2: Scan of hardtestcase.txt [from Beachboard] White-space mode.

```
scan (1)
"C:\Users\jchom\Documents\CSULB\Graduate\Fall 2021\CECS 444\Compiler\venv\Scripts\python.exe" "C:/Users/jchom/Documents/CSULB/Graduate/Fall 2021/CECS 444/Compiler/Project/hardtestcase.txt"
Enter file path: C:\Users\jchom\Documents\CSULB\Graduate\Fall 2021\CECS 444\Compiler\Project\hardtestcase.txt
Enter 1 to include white-space. Otherwise, no white-space scan 1
TOKEN: library - #include<math.h>
TOKEN: whitespace -
TOKEN: library - #include<string>
TOKEN: whitespace -
TOKEN: library - #include<iostream.h>
TOKEN: whitespace -
TOKEN: library - #include"mylib.h"
TOKEN: whitespace -
TOKEN: comment - //include library functions
TOKEN: comment - //using namespace std;
RESERVED WORD: program - IDENTIFIER
TOKEN: whitespace -
TOKEN: identifier - test1
TOKEN: semicolon - ;
TOKEN: whitespace -
RESERVED WORD: var - IDENTIFIER
TOKEN: whitespace -
TOKEN: identifier - arm
TOKEN: comma - ,
TOKEN: identifier - leg
TOKEN: colon - :
TOKEN: identifier - currency
TOKEN: semicolon - ;
TOKEN: whitespace -
TOKEN: comment - //currency
TOKEN: whitespace -
TOKEN: identifier - a
TOKEN: comma - ,
TOKEN: identifier - b
TOKEN: comma - ,
TOKEN: identifier - ab
TOKEN: colon - :
RESERVED WORD: string - IDENTIFIER
TOKEN: semicolon - ;
TOKEN: whitespace -
TOKEN: comment - //string variables
TOKEN: whitespace -
TOKEN: identifier - c
TOKEN: comma - ,
TOKEN: identifier - d
TOKEN: comma - ,
TOKEN: identifier - cd
TOKEN: colon - :
RESERVED WORD: int - IDENTIFIER
TOKEN: semicolon - ;
TOKEN: whitespace -
TOKEN: comment - //integer variables
TOKEN: whitespace -
TOKEN: identifier - e
TOKEN: comma - ,
TOKEN: identifier - goo
TOKEN: colon - :
TOKEN: identifier - real
TOKEN: semicolon - ;
TOKEN: whitespace -
TOKEN: comment - //float type
TOKEN: whitespace -
RESERVED WORD: start - IDENTIFIER
TOKEN: whitespace -
TOKEN: comment - //start of the main code
TOKEN: whitespace -
TOKEN: identifier - a
TOKEN: coloneq - :=
TOKEN: string - "hello"
TOKEN: semicolon - ;
TOKEN: whitespace -
TOKEN: identifier - b
TOKEN: coloneq - :=
TOKEN: string - " world"
TOKEN: semicolon - ;
TOKEN: whitespace -
TOKEN: identifier - ab
TOKEN: coloneq - :=
TOKEN: identifier - a
```

```

TOKEN:      plus - +
TOKEN: identifier - b
TOKEN: semicolon - ;
TOKEN: whitespace -
TOKEN: comment - /*merge strings*/
TOKEN: whitespace -
TOKEN: identifier - c
TOKEN: coloneq - :=
TOKEN: integer - 543
TOKEN: semicolon - ;
TOKEN: whitespace -
TOKEN: identifier - d
TOKEN: coloneq - :=
TOKEN: integer - 12,345
TOKEN: semicolon - ;
TOKEN: whitespace -
TOKEN: identifier - leg
TOKEN: equal - =
TOKEN: currency - $12,123.05
TOKEN: semicolon - ;
TOKEN: whitespace -
TOKEN: identifier - cd
TOKEN: coloneq - :=
TOKEN: identifier - c
TOKEN: plus - +
TOKEN: identifier - d
TOKEN: semicolon - ;
TOKEN: whitespace -
TOKEN: comment - /*add unsigned integers*/
TOKEN: whitespace -
TOKEN: identifier - e
TOKEN: coloneq - :=
TOKEN: fixed - 444,444.13
TOKEN: semicolon - ;
TOKEN: whitespace -
TOKEN: identifier - goo
TOKEN: coloneq - :=
TOKEN: identifier - c
TOKEN: plus - +
TOKEN: identifier - e
TOKEN: semicolon - ;
TOKEN: whitespace -
TOKEN: comment - /*add signed integers and reals */
TOKEN: whitespace -
TOKEN: identifier - scinot
TOKEN: coloneq - :=
TOKEN: scinotation - 3.14^+2
TOKEN: whitespace -
RESERVED WORD: read_from_device - IDENTIFIER
TOKEN: file - [A:file.test]
TOKEN: semicolon - ;
TOKEN: whitespace -
RESERVED WORD: write_to_device - IDENTIFIER
TOKEN: file - [A:test]
TOKEN: semicolon - ;
TOKEN: whitespace -
RESERVED WORD: return - IDENTIFIER
TOKEN: whitespace -
TOKEN: rsquig - 0
TOKEN: semicolon - ;
TOKEN: whitespace -
RESERVED WORD: stop - IDENTIFIER
TOKEN: whitespace -
TOKEN: comment - //stop the program
TOKEN: whitespace -
TOKEN: comment - {* something
more something
even more..
*}

```

Process finished with exit code 0

Note: The output of the scanner prints to the console as shown above as well as creates a text log in the same directory as the scanner. Both are provided in this submission.