

### Homework 3

For **Homework 3 Part 1**, you are to create a DFA that recognizes and categorizes the strings described by the following 5 regular expressions.

- Category 1: abc
- Category 2: a\*b+c\*
- Category 3: (a|c)\*b(a|c)\*
- Category 4: (abc)+
- Category 5: (a|b)+(b|c)+

Date Due: 7 September 2017 at the beginning of lecture.

To Turn in: A printed copy of your DFA.

The program at the end of this document will recognize strings belonging to the first 2 categories. You can modify the table in the recognize function in this program to verify the accuracy of your DFA.

For **Homework 3 Part 2**, you are to test the 6 C++ programs using the 3 input files in the folder Homework3-2 in the course pickup folder.

Date Due: 12 September 2017 at the beginning of lecture.

To Turn in: A printed copy of your completed table and the answer to the question in step 6.

1. Copy the folder Homework3-2 from ~tiawatts/cs460pickup. The files are:  
H3a.cpp H3b.cpp H3c.cpp H3d.cpp H3e.cpp H3f.cpp  
input.em input.np input.vi
2. Compile each of the 6 C++ programs.
3. Create a table for your observed results:

	(no input)	input.em	input.np	input.vi
H3a.cpp				
H3b.cpp				
H3c.cpp				
H3d.cpp				
H3e.cpp				
H3f.cpp				

4. Run each of the programs without providing the input file name as an argument.
5. Run each of the programs using each of the 3 input files as input. The name of the input file should be entered as a command line argument when executing the program.
6. Why do the results vary?

```

// Program: H3.cpp
// Author: Dr. Watts
// Description: This program uses a table driven function to recognize and
//              categorize strings

#include <iostream>
#include <fstream>

using namespace std;

#define ERROR 6

int recognize (string s);

int main (int argc, char * argv [])
{
    ifstream input (argv[1]);
    string one;
    while (input >> one)
    {
        int c = recognize (one);
        if (c < ERROR)
            cout << "Category " << c;
        else
            cout << "Error";
        cout << ": " << one << endl;
    }
    input.close();
    return 0;
}

int recognize (string s)
{
    static int table [][][5] = {{0, 1, 5, 7, 7}, // line 0
                                {7, 4, 2, 7, 7}, // line 1
                                {-7, 7, 7, 3, 7}, // line 2
                                {-1, 7, 7, 6, 7}, // line 3
                                {7, 4, 5, 7, 7}, // line 4
                                {-2, 7, 5, 6, 7}, // line 5
                                {-2, 7, 7, 6, 7}, // line 6
                                {-ERROR, 7, 7, 7, 7}}; // line 7

    s += " ";
    int i = 0;
    int state = 0;
    while (state >= 0)
    {
        int col = 0;
        if (s[i] >= 'a' && s[i] <= 'c')
            col = s[i] - 'a' + 1;
        else if (!isspace (s[i]))
            col = 4;
        state = table[state][col];
        i++;
    }
    return -state;
}

```