CSE 2002

THEORY OF COMPUTATION AND COMPILER DESIGN

Submitted to:-

Prof. Sathya Rajasekaran K

TEAM MEMBERS:-

Ch. Sai Adithya Reddy-15BCE1055

T. Anish-15BCE1145

P. Sai Abhishek-15BCE1276

M.Sai Lokesh – 15BCE1153

PROJECT TITLE: -

Handwriting recognition using DFA

ABSTRACT: -

In this project we will design and implement a NFA that can detect hand written digits. Image extraction will be done using Matlab.

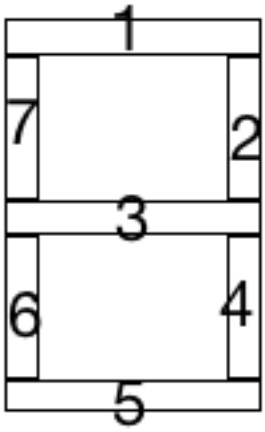
The implementation of NFA will be done in python.

Introduction:

The domain we are working is numbers. Initially we represent the all the numbers using binary. We will draw a write transition states for the numbers and draw a NFA using it. Then we will get the hand written image and extract a single digit out of it. We will then process it using Matlab and get a binary representation using it. This binary numbers is given as an input to DFA program written in python which gives the right number as output.

Representation of numbers using binary:

1. 0101000
2. 1110110
3. 1111100
4. 0111001
5. 1011101
6. 1011111
7. 1101000
8. 1111111
9. 1111101
0. 1101111



Transition states:

S is the start state:

tf[('s', '0')] = 'a';

tf[('s', '1')] = 'd';

tf[('a', '1')] = 'aa';

tf[('aa', '0')] = 'b';

tf[('aa', '1')] = 'c';

tf[('d', '0')] = 'e';

tf[('d', '1')] = 'h';

tf[('e', '1')] = 'ee';

tf[('ee', '1')] = 'eee';

tf[('eee', '1')] = 'eeee';

tf[('eeee', '0')] = 'f';

tf[('eeee', '1')] = 'g';

tf[('d', '1')] = 'h';

tf[('h', '0')] = 'i';

tf[('i', '1')] = 'ii';

tf[('ii', '0')] = 'j';

tf[('ii', '1')] = 'k';

tf[('h', '1')] = 'l';

tf[('l', '0')] = 'm';

tf[('l', '1')] = 'n';

tf[('n', '1')] = 'nn';

tf[('nn', '0')] = 'o';

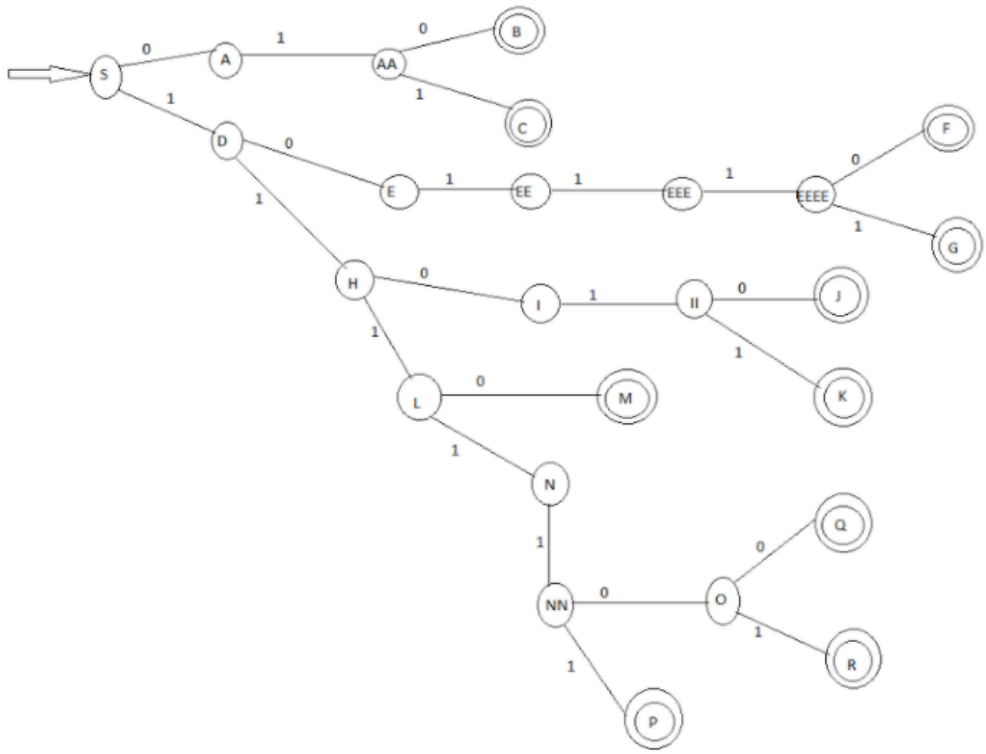
tf[('o', '0')] = 'q';

tf[('o', '1')] = 'r';

tf[('nn', '1')] = 'p';

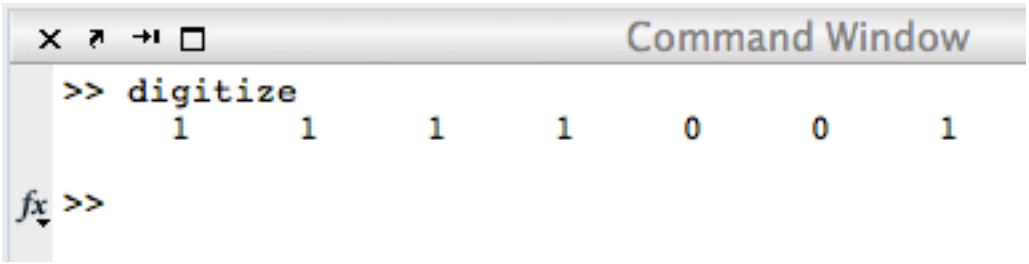
Note: tf represents transition function.

NFA:



Example input:

This image is given as an input to matlab code which generates an array of binary digits as an output.



This array of digits is given as an input to NFA and the given digit is recognized by the DFA.

Efficiency:

Total number of steps to identify a digit normally = 7 to 70 steps.

Number of steps to identify using NFA = 3 to 7 steps.

On an average we have improved the efficiency by 5 to 6 times using NFA.

Conclusion:-

By converting images into limited binary numbers, we were able to recognize handwritings using NFA. We have not only identified the digits but also have done them efficiently.

Scope of project:

References:

1. http://pythonfiddle.com/dfa-simple-implementation/
2. http://lrn.noip.info/other/books/neural/Neocognitron/1991_handwritten_alphanumeric.pdf
3. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.259.3553&rep=rep1&type=pdf
4.