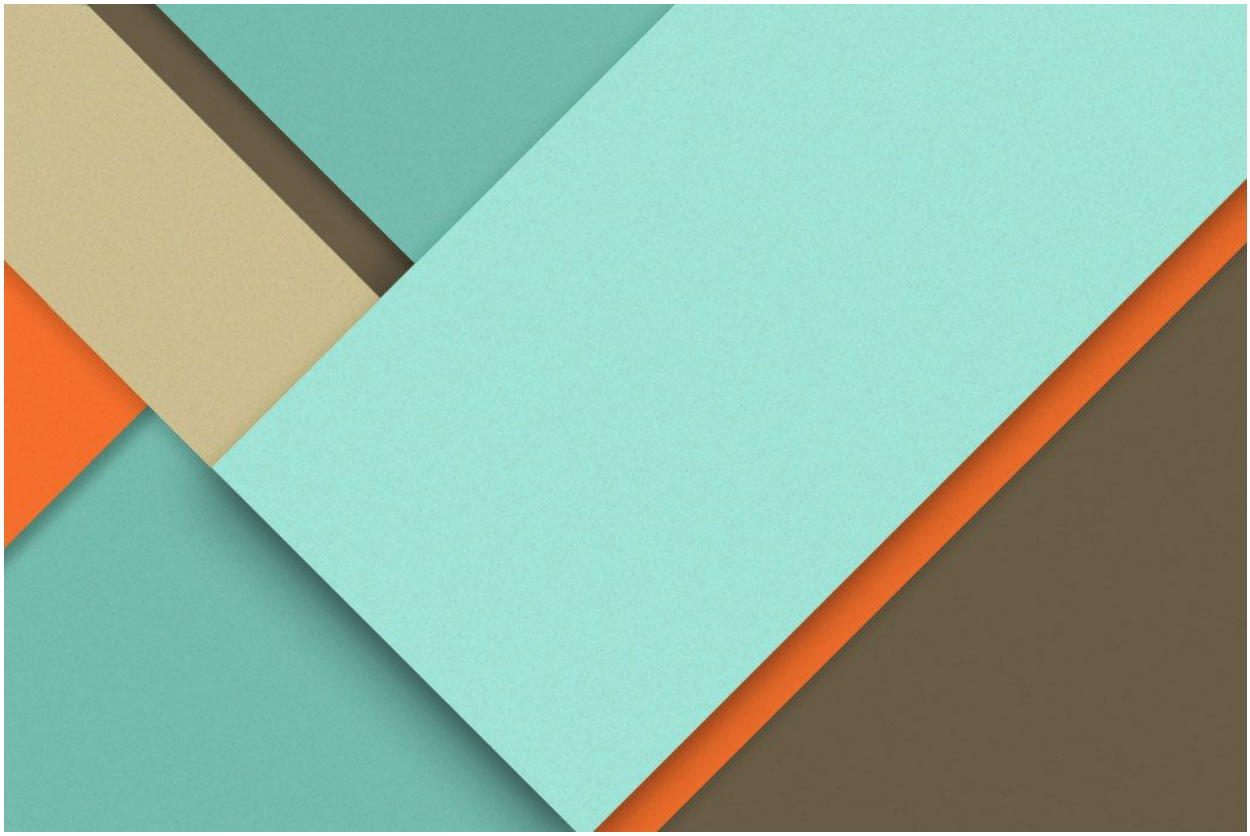


# E-NFA TO DFA CONVERTER

Design and Analysis of Algorithms (UE17CS251)


Team:

Ruben John Mampilli (PES1201700105)

Dhruv Vohra (PES1201700281)

## Overview

Finite state machine (FSM) or finite automata is a mathematical model of computation. It is an abstract machine that can be in any of the finite states at any given time. The FSM can change from one state to another based of external inputs. The change is called transition. An FSM is defined by its lists of states, its initial state and the possible transitions on the input set. A finite state machine is of two types - Deterministic Finite State Machines (DFSM or DFA) and Non-Deterministic Finite State Machine (NFSM or N-DFA).

The difference between NFA and DFA is that for a DFA, there is a mandatory transition defined from every state for every symbol in the given alphabet for the machine. However for an NFA, it is not necessary that every state will have a transition for every symbol in the alphabet. Epsilon NFA (E-NFA) is another type of NFA with epsilon transition as well. If an epsilon transition is present between 2 states A and B it means the control flow can move from state A to state B without having to encounter a specific input.

## Why convert from NFA to DFA?

In application both NFA and DFA implementation are used in the industry for various computing models. NFA is usually application specific and has minimum states and transitions targeted to its set of input strings depending upon its application whereas DFA is a more defined model with transitions for every symbol that could possible encountered in the input string even though for a specific application some transitions will never we used.

On converting an NFA to a DFA (or an E-NFA to NFA then to a DFA), understanding the machine becomes easier and its scope becomes clear. Once there is a clear understanding of the machine, improvisions can be made to make it efficient and it can also be converted back to an NFA or E-NFA if the application requires.

## Goals

1. Taking input for an E-NFA Number of states, Number of Symbols and Transition. (taken in form of a file input in this projects' implementation).
2. Convert the E-NFA to an NFA followed by converting it to a DFA using the algorithm given below and writing the final results into a file.

# Implementation

## Algorithm

```
ALGORITHM nfa_to_dfa( NFA[0...n_states][0...n_symbols], n_states, n_symbols, DFA[][] )
//Input: Algorithms takes in the E-NFA transition matrix, number of states and symbols in it.
//Output: The Algorithm copies DFA transitions for the equivalent E-NFA in a matrix.
//GLOBAL VARIABLES: STATES ← 256, SYMBOLS ← 20
Char statename[STATES][STATES], nextstate[STATES], nextstate_E[STATES];
i  ←0;
n ← 1;
Char *start_state ← NFA[0][n_sym];        //pointer to the start state
Copy(statename[0] ↔ start_state)
Copy(DFA[0] ↔ statename[0])               //start state of DFA
for i ← 0 to n do:
        for j ← 0 to n_symbols do:
                gen_next_state( nextstate, statename[i], NFA, j );
                gen_next_state( nextstate_E, statename, NFA, n_symbols );
                Dfa[i][j] = state_index(nextstate_E, statename, n);
        end
end

ALGORITHM gen_next_state(*nextstate, *currentstate, *NFA[STATES][SYMBOLS], symbol )
//Input: Takes pointer to the nextstates, currentstate, NFA matrix and current symbol.
//Output: Generates the next state in equivalent DFA based on the arguments.
Char temp[STATES];
Temp[0] = '\0';
```

for i ← 0 to length(currentstate) do:

string_merge(temp, NFA[currentstate[i]-'0'][symbol]);

Copy(nextstate ↔ temp);

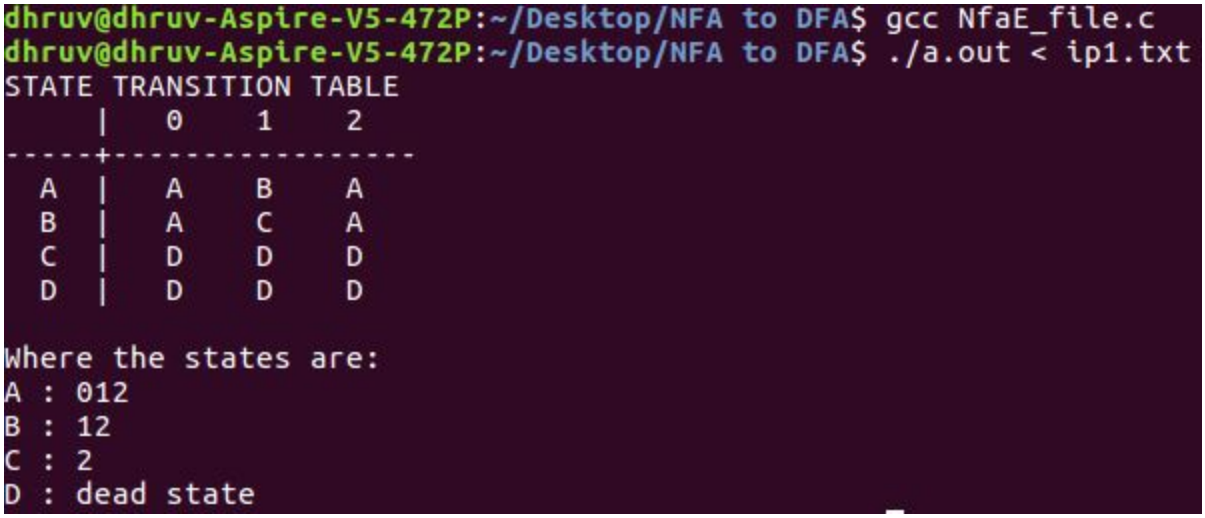
# Test Results

## I. Test Result 1

INPUT FILE

```
3
3
null
1
2
012
0
2
01
1
null
null
null
2
```

OUTPUT FILE



## II. Test Result 2

INPUT FILE

```
11
2
null
null
01247
null
null
124
3
null
2
null
null
123467
null
5
4
null
null
124567
null
null
12467
8
null
7
null
9
8
null
null
9
```

OUTPUT FILE

```
dhruv@dhruv-Aspire-V5-472P:~/Desktop/NFA to DFA$ ./a.out < ip3.txt
STATE TRANSITION TABLE
     |   0    1
-----+------------
  A  |   B    C
  B  |   B    D
  C  |   B    C
  D  |   B    C

Where the states are:
A : 01247
B : 1234678
C : 124567
D : 1245679
```