

CS 410 – Automata Theory and Formal Languages

Project 1 Report

NFA to DFA converter

By: Esrah Zahid (S020289)

Table of Contents

1. Introduction.....	3
2. Tools	3
3. System Architecture	3
4. What I Need to Know	4
5. Steps To Take.....	4

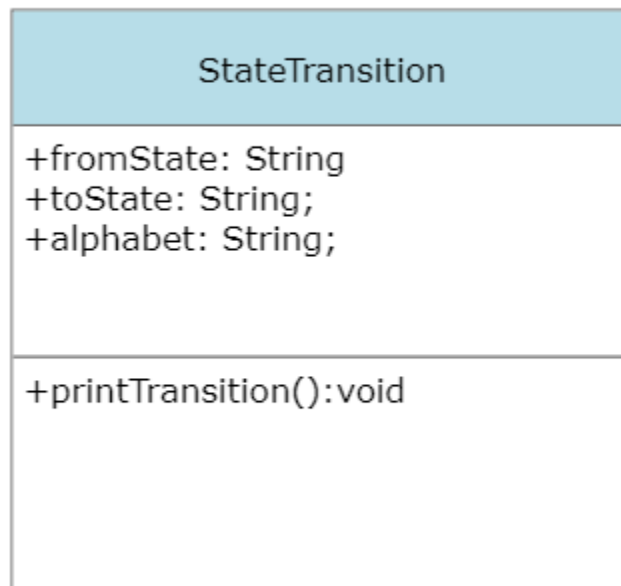
1. Introduction

This document describes the overview design for the implementation of a program code that converts NFA (Nondeterministic finite automata) to DFA (Deterministic finite automata). The program will first read a text document as input and then convert the read NFA into a DFA. The program will be able to convert any type of NFA given as input to a DFA. The resulting DFA will be printed to the console in the same format as the NFA.

2. Tools

I will be implementing this program using the Java programming language.
IntelliJ IDEA 2022.2.3 will be used to write the program code.

3. System Architecture



4. What I Need to Know

Before writing the program code, I need to know how to convert an NFA to a DFA algorithmically:

Let, $M = (Q, \Sigma, \delta, q_0, F)$ is an NFA which accepts the language $L(M)$. There should be equivalent DFA denoted by $M' = (Q', \Sigma', q_0', \delta', F')$ such that $L(M) = L(M')$.

Step 1: Initially $Q' = \varnothing$

Step 2: Add q_0 of NFA to Q' . Then find the transitions from this start state.

Step 3: In Q' , find the possible set of states for each input symbol. If this set of states is not in Q' , then add it to Q' .

Step 4: In DFA, the final state will be all the states which contain F (final states of NFA)

5. Steps To Take

Step 1) Reading the input file and initialize variables:

Let Q_DFA be a new set of states of the DFA. Q_DFA is null in the starting.

String $Q_DFA = \text{null}$;

Let T_DFA be a new transition table of the DFA.

Step 2) Generate a state transition table using 2-D array in java:

String[][] $T_NFA = \text{new String}[\text{states}][\text{alphabets}]$;

Step 3) Add start state of the NFA to Q_DFA .

Add transitions of the start state to the transition table T_DFA .

If start state makes transition to multiple states for some input alphabet, then treat those multiple states as a single state in the DFA.

Step 3) Convert the generated NFA state table to DFA:

If (any new state is present in the transition table T_DFA) {

Add the new state in Q_DFA

Add transitions of that state in the transition table T_DFA }

}

Keep repeating the loop until no new state is present in the transition table T_DFA .

Finally, the transition table T_DFA obtained is the complete transition table of the required DFA.

Step 4) Output the results to the console in the same format as the input