IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 10, Issue.1, January 2022 E-ISSN: 2347-2693

Non-Deterministic Finite Automata to Deterministic Finite Automata Conversion by Subset Construction Method using Python

Kuldeep Vayadande[1*], Krisha Patel[2], Nikita Punde[3], Shreyash Patil[4], Srushti Nikam[5], Sudhanshu Pathrabe[6]

[1,2,3,4,5,6]Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Pune, India

**Corresponding Author: kuldeep.vayadande1@vit.edu*

DOI: https://doi.org/10.26438/ijcse/v10i1.15 | Available online at: www.ijcseonline.org

Received: 14/Jan/2022, Accepted: 24/Jan/2022, Published: 31/Jan/2022

Abstract— The theory of finite automata, formal languages and complexity are the modern branches in the field of computer theory and their mathematical models play a very important role in the practical world. This makes it really important, in theory as well as practice to translate a Non-deterministic Finite Automata (NFA) into a Deterministic Finite Automata (DFA). Many different approaches are developed for doing so. In this paper, an optimized algorithm for transition from NFA to DFA is given. This method is implemented using the programming language called python3. Along with this we have also used the pandas library of python for constructing the transition tables for NFA as well as DFA. A website for the same is also created to make it more user friendly by using the streamlit library of python.

Keywords—DFA, Finite Automata, NFA, NFA to DFA, Streamlit Web Application.

I. Introduction

Automata theory [10] is considered the most important theory because it allows us to understand how machines solve problems. Modern applications of almost all models of finite automata theory go far beyond the usual compiler techniques or hardware verification. Deterministic Finite Automata (DFA) has many real-life applications such as content-aware applications, NIDS, load balancing, protocol identification, etc. [5]. In Automata Theory, if a language is recognized by a Non-deterministic Finite Automata (NFA), then there exists a Deterministic Finite Automata (DFA) that also recognizes it. Also, direct conversion of many regular languages or expressions (RE) to their corresponding DFA is a tedious task. So the conversion of regular expression to its respective NFA and then the translation of that NFA to a DFA that simulates that NFA is an easier way to do so [8]. Hence, it is preferred in theory as well as in practice to use this method.

Various efforts are still going on to develop different methods for NFA to DFA conversion. Till now, the most popular one from them is the subset construction method. In this paper, a simpler and efficient algorithm to straightly translate an NFA into a DFA is described. The implementation of this algorithm is done using the python3 programming language along with the functions of two of its popular libraries - pandas and streamlit. The pandas library is a open source library which is fast, powerful, flexible to use for data analysis and manipulation. We have used this library to construct the data frame for the NFA and DFA transition functions. A website is also created as a front-end part for the project. The website is created using the streamlit library which is a open-source web app framework for creating interactive websites written completely in python.

The primary goal of this research is to provide an efficient and easy to understand approach for the conversion of an NFA to a DFA. This method can be used to directly find the minimized DFA for the given NFA. Simple or we can say small NFA's can be easily converted to DFA's manually. But, it becomes a hectic task to translate large and complex NFA's to DFA's by a human. Whereas a machine can do this efficiently in just a number of seconds. Many researchers or scientists can improvise their work and even save time while converting such complex NFA's to their corresponding DFA's. Students can also use this interface for studying automata theory and to solve the problems related to NFA to DFA conversion.

Finite Automata (FA) has types, having their own importance and applications. In NFA, when a specific input is given to any state, it goes to multiple states. It has zero, one or more number of moves on a every input. Whereas, in DFA, the machine goes to single state for any given input. This makes it necessary to convert NFA to DFA in many cases. For this to happen we require an efficient and fast algorithm that can do the task for us and speed up our work [6][9].

The following is the order in which the paper is presented. First, we go over the existing NFA to DFA conversion literature and research and the existing computer approaches for different finite automata conversions. Following that, we go over our case in further depth and describe our role as researchers in the NFA to DFA conversion study. The paper's methodology section describes the steps of project implementation, the algorithm for NFA to DFA conversion and the challenges and questions that arose throughout the process. We wrap up by synthesizing our findings from the project and discussing about the practical consequences.

II. Related Work

[1] studied about the mechanism of the NFA to DFA conversion algorithm, its complexities. A great number of invalid or repeated traversals in the popular SUBSET algorithm were studied especially. An improved MF-SUBSET algorithm was proposed in this paper to solve the problem of the present one. The results of the simulation performed indicated that the MF-SUBSET algorithm can decide the searching strategy by increasing the number of status and traversal path flags, which could avoid the invalid or repeated traversal operation and also improve the conversion efficiency effectively.

[4] illustrates various types of automata and its accepted languages. Construction of the exact language of binary strings representing the DFA is done. Based on the proposed approach, a systematic and convenient way to design a DFA from constructed NFA's is provided.

For NFA to DFA conversion [5] used Gallier's algorithm for conversion. It doesn't detect whether the input is a DFA or an NFA. It doesn't matter whether the input is an NFA or a DFA, the output is always a DFA. Additionally, the Gallier algorithm finds reachable state from a given state means that it automatically gets rid of all the unreachable states.

[7] made use of the C++ programming language for the conversion of NFA to DFA. Conversion was difficult for NFA containing null as a transition function. It had to create separate classes for that. If it is not then the normal program for conversion can be designed. It also finds reachable state from a given state means that it automatically gets rid of all the unreachable states.

III. Methodology

Methodology of the project involves the active participation and coordination of all group members under proper guidance, resulting in the best possible outcomes. It then includes the study of all the accessible domains and finalization of project topic, relating it with study of assorted factors like socials, capability of the group, knowledge about the project domain and application and most importantly the scope of achieving the specified work and products. Then it came to the study of literature and reviews of similar projects that were accomplished before. This was all included within the pre-development stage of the methodology.

The event stage included acquiring knowledge for the project implementation like the tools to be used and the algorithms to be utilized for completion of the project. Later on, the flowchart of the whole project was formed and then the work was distributed equally among the members. After everyone completed their assigned tasks, we added a final touch to our project.

To implement the entire project we have used the python programming language and its libraries. The algorithm for conversion of NFA to DFA is written in python. We have used the pandas library of python to form a data frame for the transition table of NFA and DFA. The streamlit framework is used for developing the complete front-end for the project. We have created a website as our front-end.

a. Proposed System

The NFA to DFA conversion project helps to convert NFA to its equivalent DFA in a few seconds along with an efficient and user friendly front-end that speeds up the process with ease. Such systems can be used in various fields in theory as well as in practice.

Many algorithms are implemented for conversion of NFA to DFA. But they are mostly in theoretical form and very few are implemented in the practical world. The implemented ones are efficient enough to function but also much more complex to understand than we think. There are some projects that provide a front-end but it is also complex to use. Professionals might be able to use it but it may be a problem for the beginners. Not many of them are implemented using python entirely.

Our project also functions in a similar way like others but the algorithm we used for doing so is quick and easy to understand even for beginners. We have also added a functionality which is to check whether the entered NFA is already a DFA or not. Also, our complete project, including both back-end and front-end is designed using the python programming language only.

b. Flowchart/Algorithm

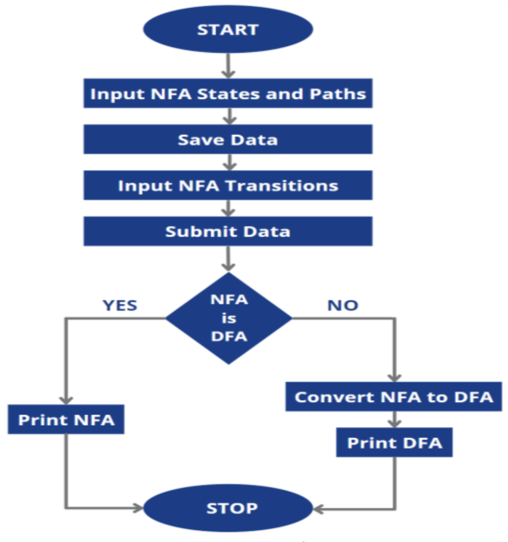


Figure 1. Flowchart of the Program

Algorithm

Overall algorithm of the project is as follows:

Step 1:- Take NFA Input
The user has to select the number of states and alphabets of the NFA. Then he has to enter their names and they will be stored in different lists. After that when he selects the save option, he will have to enter the transitions and the final state for the NFA. Once he is done with entering all the data correctly he can press the submit button. This NFA is stored in a dictionary.

Step 2:- Check if NFA is already a DFA or not
After the NFA is entered, we check if it is already a DFA or not using the nested for loops and if else conditional statement. If it is not a DFA then we move to Step 3, or else we skip that step and directly print the entered NFA as it is already a DFA.

Step 3:- Convert NFA to DFA
This is the most crucial step of the algorithm. Here, the NFA entered by the user is converted to DFA as follows:

- The entered NFA is in the form of nested dictionary, where the keys represent the states of NFA, while their values represent another dictionary that stores the transitions.
- The initial state of NFA is appended in the list for DFA states and the paths are stored in the DFA path list.
- The first key-value pair of NFA dictionary is appended into the DFA dictionary using the for loop making it the first transition of the DFA.
- Using while loop and nested for loops, we will find the transitions of each state through each path and append it as the transition of the combined state in the DFA dictionary and also add that new combined state in the new states list.
- The initial state of DFA will be the same as that of NFA while the final state(s) of DFA will be all the states containing the final state/states of NFA.

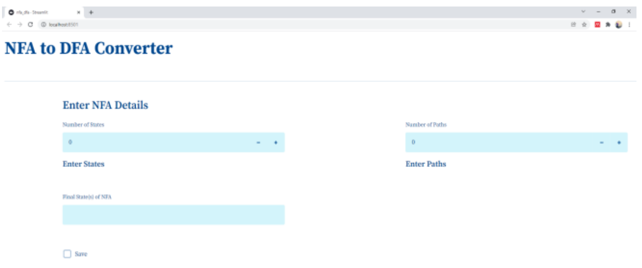
Step 4: Display the NFA Transition Table
If the user presses the submit button, the transition table of the entered NFA is displayed. This table is created by converting the NFA dictionary into a data frame using the dataframe() and transpose() functions of the pandas library in python.

Step 5:- Display the DFA
After successful conversion of NFA to DFA, the DFA transition table is displayed along with the initial state and final states of the DFA. This table is also created in the same manner as that of NFA transition table.

IV. Results and Discussion

This NFA to DFA Conversion website consists of the following two pages:

- Home Page.
- About us Page.


NFA to DFA Converter

Enter NFA Details

Number of States: 0

Number of Paths: 0

Enter States

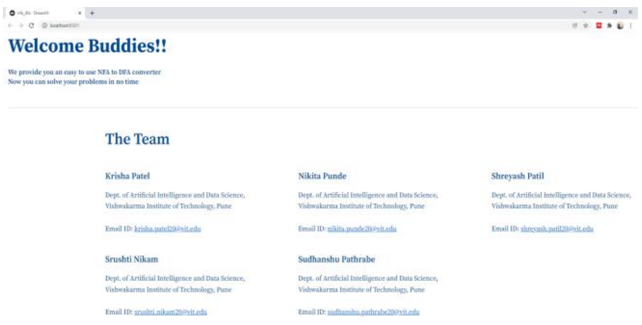
Enter Paths

Final State(s) of NFA

☐ Save

Figure 2. Home Page

Figure 2 shows us the Home page. The whole process of NFA to DFA conversion starting from taking user input to displaying the results, everything is performed on this page.


Welcome Buddies!!

We provide you an easy to use NFA to DFA converter
Now you can solve your problems in no time

The Team

Krisha Patel
Dept. of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Pune
Email ID: krisha.patel20@vit.edu

Nikita Punde
Dept. of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Pune
Email ID: nikita.punde20@vit.edu

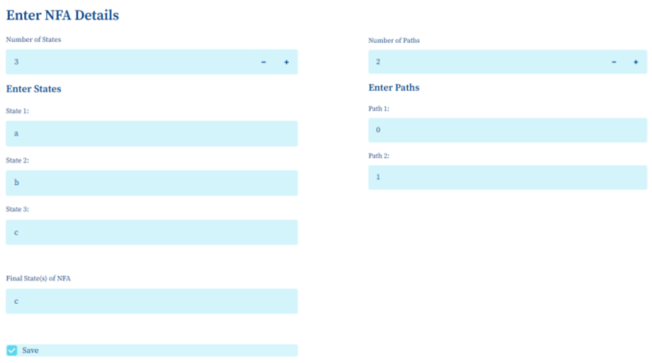
Shreyash Patil
Dept. of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Pune
Email ID: shreyash.patil20@vit.edu

Srushti Nikam
Dept. of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Pune
Email ID: srushti.nikam20@vit.edu

Sudhanshu Pathrabe
Dept. of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Pune
Email ID: sudhanshu.pathrabe20@vit.edu

Figure 3. About Us Page

Figure 3 is of the About Us page. This page contains the details and contact information about the developers of the website which is our group members.


Enter NFA Details

Number of States: 3

Number of Paths: 2

Enter States

State 1: a

State 2: b

State 3: c

Enter Paths

Path 1: 0

Path 2: 1

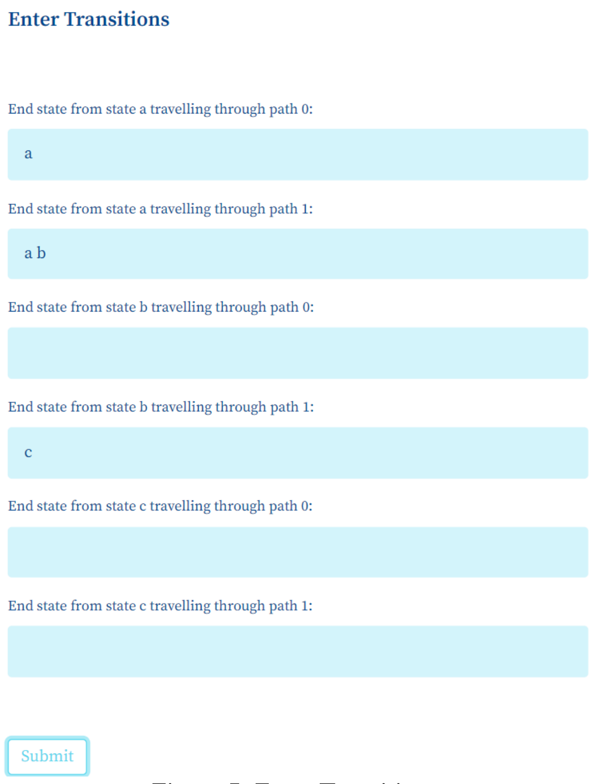
Final State(s) of NFA: c

✓ Save

Figure 4. Enter States and Paths

Figure 4 represents the state of home page where we can increase or decrease the number of states and paths. According to the chosen number the text boxes to enter the names of states and paths are formed. A text box for entering the final state(s) of NFA is also there.

After entering all the information we can select the save option to save the data and generate the text boxes to enter the transitions of NFA.



Enter Transitions

End state from state a travelling through path 0:

a

End state from state a travelling through path 1:

a b

End state from state b travelling through path 0:

End state from state b travelling through path 1:

c

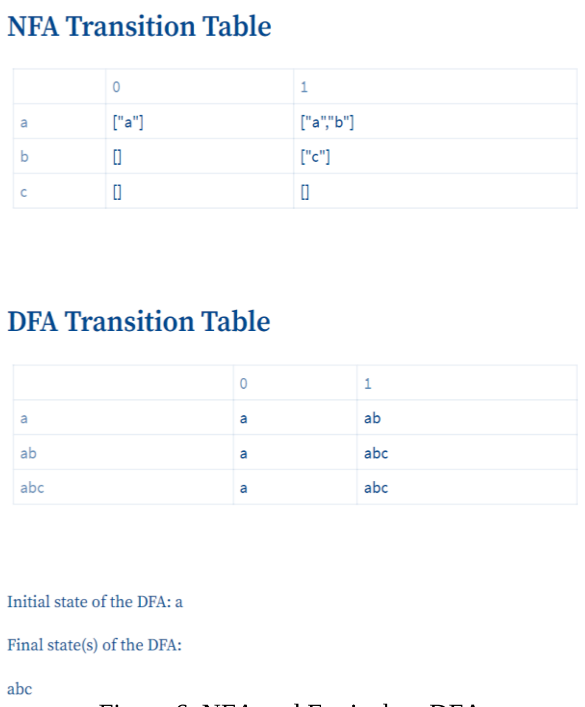
End state from state c travelling through path 0:

End state from state c travelling through path 1:

Figure 5. Enter Transitions

Figure 5 shows that state of the home page after selecting the save option. Here, we can see the text boxes to enter the transitions for the NFA.

If the transition is null, then leave it blank and if there are more than one states for transition enter the states leaving a single white space between them in the same text box. After entering all the transitions, click on the submit button.



NFA Transition Table

	0	1
a	["a"]	["a","b"]
b	[]	["c"]
c	[]	[]

DFA Transition Table

	0	1
a	a	ab
ab	a	abc
abc	a	abc

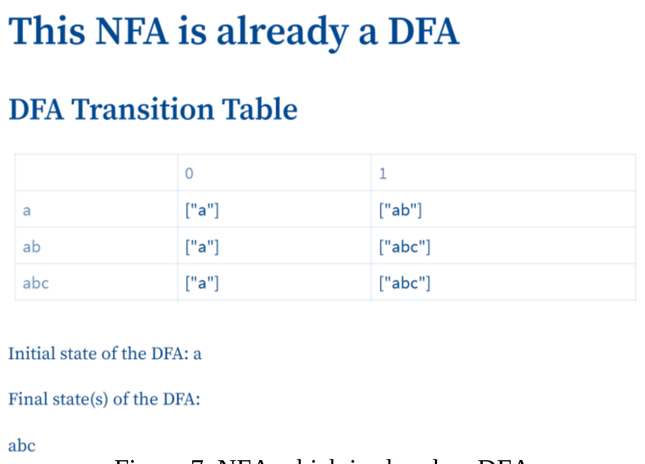
Initial state of the DFA: a

Final state(s) of the DFA:

abc

Figure 6. NFA and Equivalent DFA

Figure 6 represents the state of the Home page after selecting the submit button when the DFA is not the same as NFA. Here, we can see the Transition Table of the entered NFA and the results of the program, that is the DFA's Transition Table, it's Initial State and Final States.



This NFA is already a DFA

DFA Transition Table

	0	1
a	["a"]	["ab"]
ab	["a"]	["abc"]
abc	["a"]	["abc"]

Initial state of the DFA: a

Final state(s) of the DFA:

abc

Figure 7. NFA which is already a DFA

Figure 7 represents the state of the Home page after selecting the submit button when the NFA is already a DFA. Here, we can see the Transition Table of the entered NFA and it's Initial State and Final States as it is only the DFA.

V. CONCLUSION AND FUTURE SCOPE

While researching about the project, we came across many different approaches for converting NFA to DFA. The algorithm which we have used is implemented using the python3 programming language. Even the beginners can easily understand the working of this algorithm. The website built is also easy to use and provides a good user experience. We can convert an NFA to DFA using this website efficiently and within a few seconds. This will help to save time and also avoid human errors that may occur mostly while translating a complex NFA to DFA.

a. Limitations

- Our program cannot convert NFA with ε to DFA directly. We will have to manually convert it to NFA first and then we can find its equivalent DFA using the program.
- It can derive the transition table of DFA from the given NFA input but cannot create the DFA's state diagram.

b. Future Scope

Adding other conversions such as NFA to RE, DFA to RE, etc. can be very helpful to make a full fledged 'Automata Theory' website. Detailed steps of conversion can also be included for proper understanding. The state diagrams for for DFA and NFA can also be added. This website can be deployed for public usage. Mobile applications or more dynamic websites can also be designed for a better user experience.

REFERENCES

[1] Jing Maohua, G.-R. Li, W.-B. Shi, S.-X. Cai, "Improved conversion algorithm from NFA to DFA", Dongbei Daxue Xuebao/Journal of Northeastern University, **Vol.33(4), April 2012.**

[2] C. H. Chanh, R. Paige, "From regular expression to DFA using compressed NFA", Theoretical Computer Science, **Vol.178, Issue.1-2, pp.1-36, May 1997.**

[3] N.Murugesan, O. V. Shanmuga Sundaram, "A General Approach to DFA Construction", International Journal of Research in Computer Science, **Vol.2, Issue.4, pp.12-17, 2015.**

[4] P. Linz, "An Introduction to formal languages and Automata", D. C. Health and Company, **1996.**

[5] K. Salomaa, S. Yu, "NFA to DFA conversion for finite languages", Lecture Notes in Computer Science, **pp.149-158, 2006.**

[6] M. Sipser, "An Introduction to the Theory of Computation", Second Edition, Thomson Course Technology, **2006.**

[7] M. Davoudi-Monfared, R. shafiezadeh garousi, E. S. Haghi, S. Zeinali and S.Mohebali, "Converting different automata with programming C++", International Journal of Advanced Computer Research, **Vol.5, Issue.21, December 2015.**

[8] Raza, Mir Adil, Kuldeep Baban Vayadande, and H. D. Preetham. "DJANGO MANAGEMENT OF MEDICAL STORE.", International Research Journal of Modernization in Engineering Technology and Science, **Vol.2, Issue.11, November 2020.**

[9] K.B. Vayadande, Nikhil D. Karande," Automatic Detection and Correction of Software Faults: A Review Paper", International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653, **Vol.8, Issue.4, April 2020.**

[10] Kuldeep Vayadande, Ritesh Pokarne, Mahalaxmi Phaldesai, Tanushri Bhuruk, Tanmai Patil, Prachi Kumar, "SIMULATION OF CONWAY'S GAME OF LIFE USING CELLULAR AUTOMATA" International Research Journal of Engineering and Technology, **Vol.9, Issue.1, Jan 2022**, e-ISSN: 2395-0056, p-ISSN: 2395-0072.

[11] Kuldeep Vayadande, Harshwardhan More, Omkar More, Shubham Mulay, Atharva Pathak, Vishwam Talnikar, " Pac Man: Game Development using PDA and OOP", International Research Journal of Engineering and Technology (IRJET), **Volume: 09 Issue: 01 | Jan 2022**, e-ISSN: 2395-0056, p-ISSN: 2395-0072.

AUTHORS PROFILE

K B Vayadande has completed Bachelor of Engineering in Computer Science & Engineering from Shivaji University, Kolhapur in 2009. He has completed his M.Tech in Computer Science & Technology from Shivaji University in 2014. Also he has completed his Ph.D. in Computer Science & Engineering in year 2021.

Krisha Patel is currently pursuing B.tech degree in Artificial Intelligence and Data Science from Vishwakarma Institute of Technology, Pune. She has completed schooling at Gujarat Public School (CBSE), Vadodara, Gujarat.

Nikita Punde studied in School of Scholars Akola till Class 10th in CBSE Board. Currently she is pursuing B.tech in Artificial Intelligence and Data Science from Vishwakarma Institute of Technology, Pune.

Shreyash Patil is currently pursuing B.tech degree in Artificial Intelligence and Data Science from Vishwakarma Institute of Technology Pune. He has completed schooling at School of Scholars, Akola.

Srushti Nikam is currently studying Artificial Intelligence and Data Science at Vishwakarma Institute of Technology, Pune. She completed schooling from Humemchenry school, Pune and then went to KHS junior college with science field.

Sudhanshu Pathrabe is currently studying Artificial Intelligence and Data Science at Vishwakarma Institute of Technology, Pune. He has completed schooling from Sanskar Vidya Sagar, Nagpur and then went to Dr. Ambedkar college, Nagpur with science field.