

NFA to DFA conversion in C

Janhavi Gangurde¹, Komal Thakare², Aarti Ghuge³, Shivam More⁴, Rohan Waware⁵

^{1,2,3,4,5} Student, Dept. of Artificial Intelligence & Data Science, Vishwakarma Institute of Technology, Maharashtra, India

Abstract - Automata Theory is an enthralling domain in computer science in which we develop abstract models of the logic or the computation that runs in the background of any machine. DFAs and NFAs play a tremendous role in designing, analyzing the dynamic behavior of these models. Each DFA is an NFA, but each NFA is not a DFA. However each NFA can be translated or made equivalent to a DFA. There are a few methods already available for converting a NFA into its equivalent DFA. Our aim through this paper is to design an efficient algorithm for NFA to DFA conversion in C language.

Key Words: NFA, DFA, conversion

1. INTRODUCTION

In automata theory, a machine can be termed as DFA if on reading the input and the source state the obtained transition does not contain two distinct paths for the same input. On the other hand a machine can be termed as NFA if there exists two paths for the same input. The '5 tuple definition' of DFA and NFA are the same except the transition function:-

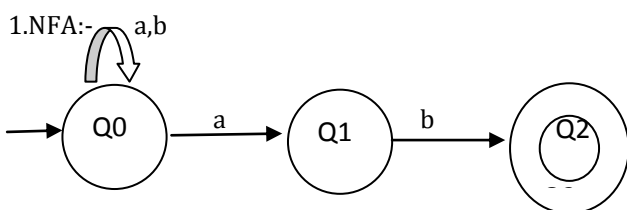
$$\text{DFA- } \delta: Q \times \Sigma \rightarrow Q$$

$$\text{NFA- } \delta: Q \times \Sigma \rightarrow 2^Q$$

The conversion of NFA to DFA in C is implemented in the same way as it is done manually. Following are the steps to convert a NFA to DFA manually:-

1. Construct a NFA for the given statement.
2. Generate its transition table accordingly.
3. From the transition table of NFA generate the transition table of DFA.
4. Construct the DFA from the transition table of DFA.

Eg:-NFA-DFA which accepts strings containing {a.b} with ab at the end.



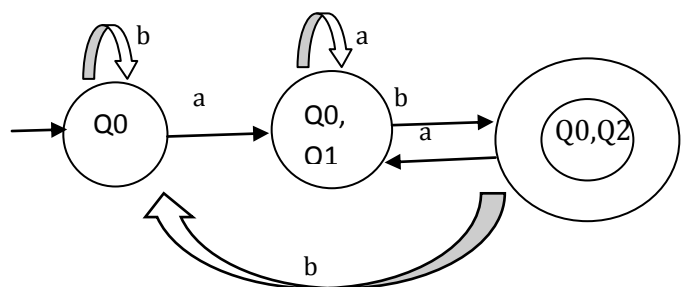
2. Transition table of NFA:-

State	a	b
Q0	Q0,Q1	Q0
Q1	-	Q2
Q2		

3. Transition table of DFA:-

State	a	b
Q0	Q0,Q1	Q0
Q0,Q1	Q0,Q1	Q0,Q2
Q0,Q2	Q0,Q1	Q0

3.DFA:-



3. LITERATURE REVIEW

[1]. A paper by M. Davoudi-Monfared¹, R. shafiezadehgarousi, E. S. Hagi¹, S. Zeinali¹ and S. Mohebbali illustrates the NFA to DFA conversion in C++. It uses data structures like array and linked list and other OOP concepts like classes, functions, grafted list, etc.

[2]. Ho Ngoc Vinh and Nguyen Thi Thu Ha together compiled a paper in which they have depicted the conversion of NFA to DFA using the concept of bounded words. In this paper, concepts of bounded words on an alphabet A, languages and monoid morphism are introduced.

[3]. Ghafoor has contributed to a paper in CSE Spring 18 which uses the Gallier's algorithm for NFA to DFA conversion. It uses the various data structures in C++ like vectors, parser classes, etc. The final output obtained is the DFA tuple which is actually represented in the form of strings.

[4]. A paper entitled A New Method for Translating NFA into DFA Cheng Yuanbin School of Mathematics & Computer Science, Jiangnan University, Wuhan, China. In this algorithm of NFA to DFA conversion when we have the epsilon case it not required to input the epsilon It can directly convert a NFA to DFA.

[5]. 'NFA to DFA transformation for finite languages' a paper contributed by Kai Salomaa & Sheng Yu has been published in the International Workshop on Implementing Automata. This paper has majorly focused on the upper bound of the number of states in DFA that is translated into a NFA.

[6]. Galina Jirásková and Ivana Krajňáková together contributed to a paper named "NFA-to-DFA Trade-Off for Regular Operations" in the International Conference on Descriptive Complexity of Formal Systems. In this type of conversion the NFA is represented in the form of operands of a Regular Expression and then it is converted into the DFA.

[7]. M Jing, W SHI, C YU contributed to a paper entitled "Subset:-An improved conversion Algorithm for NFA to DFA" in the American Journal of Engineering and Technology which stated the use of subset conversion algorithm as an efficient method for NFA to DFA conversion and thus is applicable to string matching.

[8]. "Deterministic vs. Non Deterministic Finite Automata in Automata Processing" is a paper compiled by FA Siddique, TJ Tracy II, N Brunelle. This paper performs an empirical study on the conversion of any optimized NFA to a minimized DFA.

[9]. NFA to DFA Conversion of Subregular Language Families is a paper compiled by H Bordihn, M Holzer, M Kutrib. It provides another algorithm for the efficient conversion of any NFA to its optimum DFA.

4. METHODOLOGY

A.) Proposed System:-

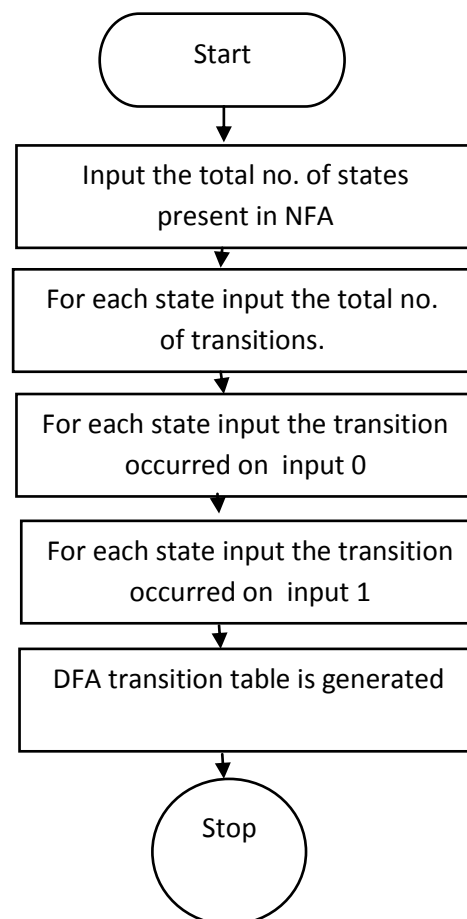
Our crucial goal for this project is to achieve an efficient algorithm for NFA to DFA conversion. And to implement it in the best possible way we have used data structures present in the C language so that it is as simple to convert a NFA to DFA programmatically as it is manually. Therefore this algorithm is easy to interpret for student who are beginners in Theory of Computation. In this system, initially we have declared structures for storing the NFA states and its related information. Later for accessing, traversing, adding, deleting the states we have used the concept of queue with linked list.

B.) Flowchart:-

NFA to DFA conversion in C is developed using the basic C concepts like structures, queues, linked lists, dynamic memory

allocation, etc. The flow of the program is such that initially when we run the program we need to input the total number of states present in our NFA. An extremely important point over here is that the variables in the language are represented as 0 or 1. Eventually even if our NFA contains language comprising of {a,b} it will still be represented as 0 or 1. Once we have inputted the total number of states, then for every state we need to mention the total number of transitions from that state on a given input either 1 or 0. Also at the end of each state the program enquires if that particular state is a final state or not. After providing all the information of the NFA it generates the transition table of its corresponding DFA. It also specifies the final state of DFA. The program then gets terminated successfully.

Here is the detailed flowchart of the same.



C.) Algorithm:-

1. Start
2. Input the total no. of states present in NFA
3. For each state input the total no. of transitions.
4. For each state input the transition occurred on input 0
5. For each state input the transition occurred on input 1
6. DFA transition table is generated along with the final state.
7. Stop

5. RESULTS AND DISCUSSIONS

For the purpose of converting a NFA to DFA there exists many different methods like bitmasking, subset conversion, bounded words, etc. Our aim through this paper is to study these existing methods and deliver a more efficient algorithm for the NFA to DFA conversion. NFA to DFA conversion in C using data structures is the most efficient algorithm as it functions the same way as we convert a NFA to DFA manually. On taking the transition table of NFA as input the transition table of DFA is successfully generated along with the mentioning of the final state. Following are the snapshots of the output.

1.

```
PS C:\Users\janhavi Gangurde> cd "c:\Users\janhavi Gangurde"
Number of states : 3
State q0 :
Transitions : input(0)
Number of transitions : 1
Transition 0 : 0
Transitions : input(1)
Number of transitions : 2
Transition 0 : 0
Transition 1 : 1
Final? (y/n) : n
State q1 :
Transitions : input(0)
Number of transitions : 0
Transitions : input(1)
Number of transitions : 1
Transition 0 : 2
Final? (y/n) : n
State q2 :
Transitions : input(0)
Number of transitions : 0
Transitions : input(1)
Number of transitions : 0
Final? (y/n) : y
```

The given snapshot depicts the input provided such as the total number of states, number of transitions, transition states obtained on the given input, checking if the state is a final state, etc.

2.

```
DFA Transition table :
[q0] [q0] [q0q1]
[q0q1] [q0] [q0q1q2]
[q0q1q2] [q0] [q0q1q2] *FINAL STATE*
PS C:\Users\janhavi Gangurde\Downloads> |
```

The given snapshot depicts the generated transition table of DFA. Here the state 'Q0Q1Q2' is the obtained final state.

5. CONCLUSION

The proposed system converts the given NFA to DFA based on the transition table of NFA it generates the transition table of DFA and thus the DFA is generated. This algorithm can be used in various areas where the NFA to DFA conversion is required.

6. FUTURE SCOPE

NFA is used in the making of regular expressions. DFA has a wide range of applications including Lexical Analyzers for parsers, Elevators, thermostats, Train track switches, security properties, grep etc. With these tremendous applications NFA to DFA conversion of these models is required frequently. Hence NFA to DFA conversion in C is the most efficient and reliable algorithm.

7. ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our guide Dr. Kuldeep Vayadande Sir for giving us the opportunity to work on this proposed system entitled "NFA to DFA conversion in C". Through this system we were able to get in depth knowledge about finite automata, automata and its applications as well.

8. REFERENCES

- [1]. Kuldeep Vayadande, Aditya Bodhankar, Ajinkya Mahajan, Diksha Prasad, Shivani Mahajan, Aishwarya Pujari and Riya Dhakalkar, "Classification of Depression on social media using Distant Supervision", ITM Web Conf. Volume 50, 2022.
- [2]. Kuldeep Vayadande, Rahebar Shaikh, Suraj Rothe, Sangam Patil, Tanuj Baware and Sameer Naik, "Blockchain-Based Land Record System", ITM Web Conf. Volume 50, 2022.
- [3]. Kuldeep Vayadande, Kirti Agarwal, Aadesh Kabra, Ketan Gangwal and Atharv Kinage, "Cryptography using Automata Theory", ITM Web Conf. Volume 50, 2022
- [4]. Samruddhi Mumbare, Kunal Shivam, Priyanka Lokhande, Samruddhi Zaware, Varad Deshpande and Kuldeep Vayadande, "Software Controller using Hand Gestures", ITM Web Conf. Volume 50, 2022
- [5]. Preetham, H. D., and Kuldeep Baban Vayadande. "Online Crime Reporting System Using Python Django."
- [6]. Vayadande, Kuldeep B., et al. "Simulation and Testing of Deterministic Finite Automata Machine." International Journal of Computer Sciences and Engineering 10.1 (2022): 13-17.
- [7]. Vayadande, Kuldeep, et al. "Modulo Calculator Using Tkinter Library." EasyChair Preprint 7578 (2022).

- [8].VAYADANDE, KULDEEP. "Simulating Derivations of Context-Free Grammar." (2022).
- [9].Vayadande, Kuldeep, Ram Mandhana, Kaustubh Paralkar, Dhananjay Pawal, Siddhant Deshpande, and Vishal Sonkusale. "Pattern Matching in File System." International Journal of Computer Applications 975: 8887.
- [10].Vayadande, Kuldeep, Ritesh Pokarne, Mahalakshmi Phaldesai, Tanushri Bhuruk, Tanmay Patil, and Prachi Kumar. "Simulation Of Conway's Game Of Life Using Cellular Automata." SIMULATION 9, no. 01 (2022).
- [11].Gurav, Rohit, Sakshi Suryawanshi, Parth Narkhede, Sankalp Patil, Sejal Hukare, and Kuldeep Vayadande. "Universal Turing machine simulator." International Journal of Advance Research, Ideas and Innovations in Technology, ISSN (2022).
- [12].Vayadande, Kuldeep B., Parth Sheth, Arvind Shelke, Vaishnavi Patil, Srushti Shevate, and Chinmayee Sawakare. "Simulation and Testing of Deterministic Finite Automata Machine." International Journal of Computer Sciences and Engineering 10, no. 1 (2022): 13-17.
- [13].Vayadande, Kuldeep, Ram Mandhana, Kaustubh Paralkar, Dhananjay Pawal, Siddhant Deshpande, and Vishal Sonkusale. "Pattern Matching in File System." International Journal of Computer Applications 975: 8887.
- [14].Vayadande, Kuldeep B., and Surendra Yadav. "A Review paper on Detection of Moving Object in Dynamic Background." International Journal of Computer Sciences and Engineering 6, no. 9 (2018): 877-880.
- [15].Vayadande, Kuldeep, Neha Bhavar, Sayee Chauhan, Sushrut Kulkarni, Abhijit Thorat, and Yash Annapure. Spell Checker Model for String Comparison in Automata. No. 7375. EasyChair, 2022.
- [16].Vayadande, Kuldeep, Harshwardhan More, Omkar More, Shubham Mulay, Atharva Pathak, and Vishwam Talnikar. "Pac Man: Game Development using PDA and OOP." (2022).
- [17].Preetham, H. D., and Kuldeep Baban Vayadande. "Online Crime Reporting System Using Python Django."
- [18].Vayadande, Kuldeep. "Harshwardhan More, Omkar More, Shubham Mulay, Atahrv Pathak, Vishwam Talanikar,"Pac Man: Game Development using PDA and OOP"." International Research Journal of Engineering and Technology (IRJET), e-ISSN (2022): 2395-0056.
- [19].Ingale, Varad, Kuldeep Vayadande, Vivek Verma, Abhishek Yeole, Sahil Zavar, and Zoya Jamadar. "Lexical analyzer using DFA." International Journal of Advance Research, Ideas and Innovations in Technology, www.IJARIT.com.
- [20]. Manjramkar, Devang, Adwait Gharpure, Aayush Gore, Ishan Gujarathi, and Dhananjay Deore. "A Review Paper on Document text search based on nondeterministic automata." (2022).
- [21]. Chandra, Arunav, Aashay Bongulwar, Aayush Jadhav, Rishikesh Ahire, Amogh Dumbre, Sumaan Ali, Anveshika Kamble, Rohit Arole, Bijin Jiby, and Sukhpreet Bhatti. Survey on Randomly Generating English Sentences. No. 7655. EasyChair, 2022.
- [22]. A paper by M. Davoudi-Monfared¹, R. shafiezadehgarousi, E. S. Haghi¹, S. Zeinali¹ and S.Mohebbali illustrates the NFA to DFA conversion in C++.[1]It uses data structures like array and linked list and other OOP concepts like classes,functions,grafted list,etc[1].
- [23]. Ho Ngoc Vinh and Nguyen Thi Thu Ha together compiled a paper in which they have depicted the conversion of NFA to DFA using the concept of bounded words [2]In this paper,concepts of bounded words on an alphabet A,languages and monoid morphism are introduced.[2]
- [24].Ghafoor has contributed to a paper in CSESpring18 which uses the Gallier's algorithm for NFA to DFA conversion[3].
- [25]. Hopcroft J, Motwani R, Ullman, Addison-Wesley, "Introduction to Automata Theory, Languages and Computation", Second Edition, ISBN 81-7808-347-7.
- [26]. John C. Martin," Introduction to Languages and The Theory of Computation", Fourth Edition, McGraw Hill,ISBN 978-0-07-319146-1.
- [27]. The C Programming Language. 2nd EditionBook by Brian Kernighan and Dennis Ritchie
- [28]. M Jing, W SHI, C YU contributed to a paper entitled "Subset:-An improved conversion Algorithm for NFA to DFA" in the American Journal of Engineering and Technology which stated the use of subset conversion algorithm as an efficient method for NFA to DFA conversion and thus is applicable to string matching.
- [29]. "Deterministic vs. Non Deterministic Finite Automata in Automata Processing" is a paper compiled by FA Siddique, TJ Tracy II, N Brunelle. This paper performs an empirical study on the conversion of any optimized NFA to a minimized DFA.
- [30]. Galina Jirásková and Ivana Krajňáková together contributed to a paper named "NFA-to-DFA Trade-Off for Regular Operations" in the International Conference on Descriptive Complexity of Formal Systems.In this type of conversion the NFA is represented in the form of operands of a Regular Expression and then it is converted into the DFA