TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

PROJECT PROPOSAL ON
" RegEx2DFA Converter "

Submitted To:

Daya Sagar Baral,

Department of Electronics and Computer Engineering,

IOE Pulchowk Campus

Submitted By:

Aman Dhaubanjar (079BCT010)

Ankit Pokhrel (079BCT016)

Ankit Prasad Kisi (079BCT017)

Acknowledgement

We would like to extend our sincere gratitude to **Mr. Daya Sagar Baral**, Lecturer, Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, for his guidance and support throughout this semester teaching us Object Oriented concepts in great detail.

In our project, we pay special thanks for the author and developer of C++, **Bjarne Stroustrup**, whose book laid the foundation for us to write effective code. Additionally, we would like to acknowledge the developers and maintainers of Graphviz, an open-source graph visualization software. Their contributions have been invaluable in providing a robust and flexible tool that has greatly facilitated the visualization of our DFA graphs.

Additionally, we are grateful to **Mr. Anuj Ghimire** for his invaluable insights on Automata, which have provided us with the knowledge to work in this field.

Table of Contents

A. Introduction

The process of converting regular expressions to deterministic finite automata (DFA) is a fundamental concept in computer science, particularly in the field of automata theory and compiler construction. This project aims to develop a C++ program that performs this conversion efficiently. The proposed system utilizes the Shunting-Yard Algorithm for parsing regular expressions, Thompson's Construction Algorithm for converting regular expressions to non-deterministic finite automata (NFA), and then further converts the NFA to DFA. Graphviz is used to generate and visualize the resulting graphs.

B. Objectives

· To create a C++ program that converts regular expressions into Deterministic Finite Automata (DFA), ensuring that the output is represented as a visual DFA diagram using Graphviz.

· To provide a practical tool for the Theory of Computation course, facilitating students' understanding and study of DFA concepts through hands-on application.

C. Existing System

In our research, we found some projects and web applications mostly based on Python and JavaScript. Upon inspection of the available source code of those projects, we have observed that they are using libraries like Pandas. Our approach is to write the core logic of the program using minimal amount or no third-party libraries. We plan on reading research papers and articles to fully understand the required algorithms and their underlying intricacies.

D. Proposed System

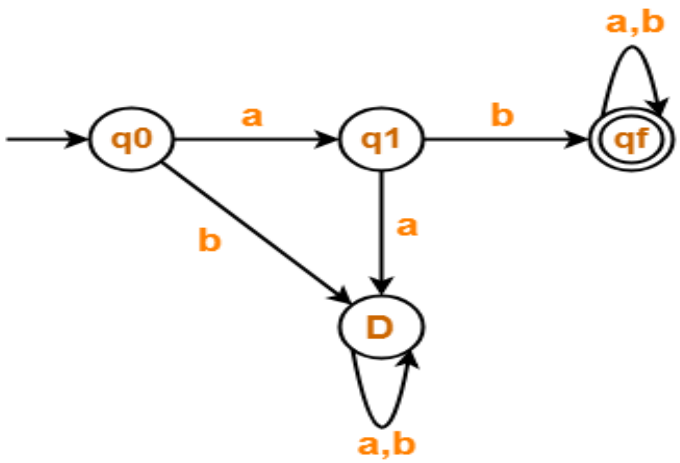
a. Description

As detailed previously, the proposed system is designed to convert regular expressions to Deterministic Finite Automata (DFA). The system will take a regular expression as input, parse it into postfix notation using the **Shunting-Yard Algorithm**, convert it to Non-Deterministic Finite Automata (NFA) using **Thompson's Construction Algorithm**, and finally convert Non-Deterministic Finite Automata (NFA) with epsilon transitions (ε-NFA) into an equivalent Deterministic Finite automata (DFA) using **Subset Construction Algorithm** (also known as **the Powerset Construction Algorithm**). The resulting DFA will be visualized using Graphviz.

Graphviz is open-source graph visualization software that represents structural information as diagrams of abstract graphs and networks. Graphviz layout programs take graph descriptions in a simple text language and produce diagrams in multiple formats, including images, SVG for web pages, and Postscript for documents. It is widely used in various fields such as networking, bioinformatics, software engineering, and machine learning.

For Example:

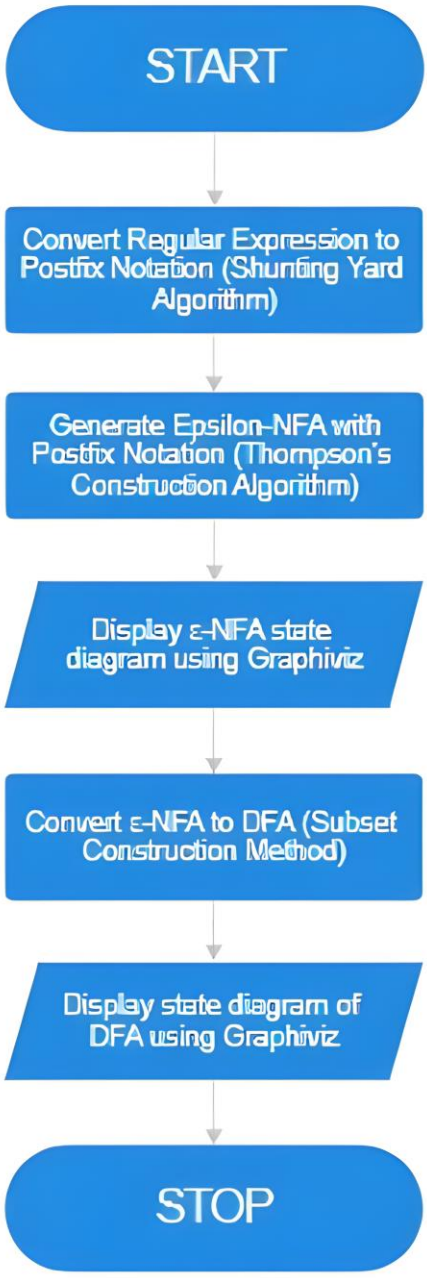
Regular Expression: ab(a + b)* (Input)



The required DFA (Output)

b. Block Diagram

The organization and development of our program, primarily involves the following-



E. Methodology

To convert a regular expression into a DFA, we will follow a systematic methodology. Initially, the regular expression will be transformed into postfix notation using the **Shunting Yard Algorithm**. Subsequently, the postfix expression will be processed by **Thompson's Algorithm** to generate an epsilon-NFA. This epsilon-NFA can then be visualized through a state diagram.

Once we have the epsilon-NFA, our next step is to convert it into a DFA using the **Subset construction method**. This method involves systematically constructing DFA states from subsets of NFA states. Once the DFA is constructed, its state diagram can be generated programmatically using the Graphviz library.

This approach ensures a step-by-step transformation from a regular expression to a DFA, leveraging various algorithms for accurate and efficient conversion.

F. Project Scope

The scope of this project includes:

- Developing a robust C++ program that handles various regular expressions.
- Ensuring the system efficiently converts complex regular expressions to DFA.
- Providing detailed documentation and user manual for educational purposes.
- Offering visualization tools to help understand the structure of the DFA.

G. Project Schedule

The complexities inherent in converting regular expressions to DFA make this project challenging. The process involves parsing expressions, constructing NFAs, converting to DFAs, and visualizing these automata with Graphviz. Each of these steps involves intricate algorithms and data structures that require careful implementation and testing.

We anticipate at least one month for further development. Currently, we have encountered and resolved issues related to parsing expressions and handling state transitions. However, we need to account for the unpredictable challenges that may arise, including efficient memory management, optimization of algorithms, and accurate graph visualization.

The development bottlenecks could include any or all of parsing, state conversion, and visualization, as these tasks grow in complexity with a larger codebase.

H. Conclusion

This proposal outlines our approach to converting regular expressions to DFA using C++. We believe that this project will significantly enhance our understanding of automata theory and provide a valuable tool for educational purposes.