

# REGEX TO DFA VISUALIZER – REPORT

---

## **Project Title**

Regex to DFA Visualizer using Python

## **Team Members**

Heba Youssef – 231000723

Nourhan Elsheikh – 231000853

Reem Ali – 231000165

Omar Magdy – 231000967

## **Project Description**

This project provides a fully interactive tool that converts a Regular Expression (Regex) into a deterministic finite automaton (DFA).

The system helps users understand the behavior of automata step-by-step by allowing them to convert, and test input strings directly on the generated DFA.

The input to the system is a valid regular expression, and upon conversion, the user can enter a string to check whether it is accepted or rejected by the resulting DFA.

## **Input Format**

User enters a valid regular expression such as: aabb

## **Output Format**

- Acceptance or rejection of a test input string

## **Inside Mechanism**

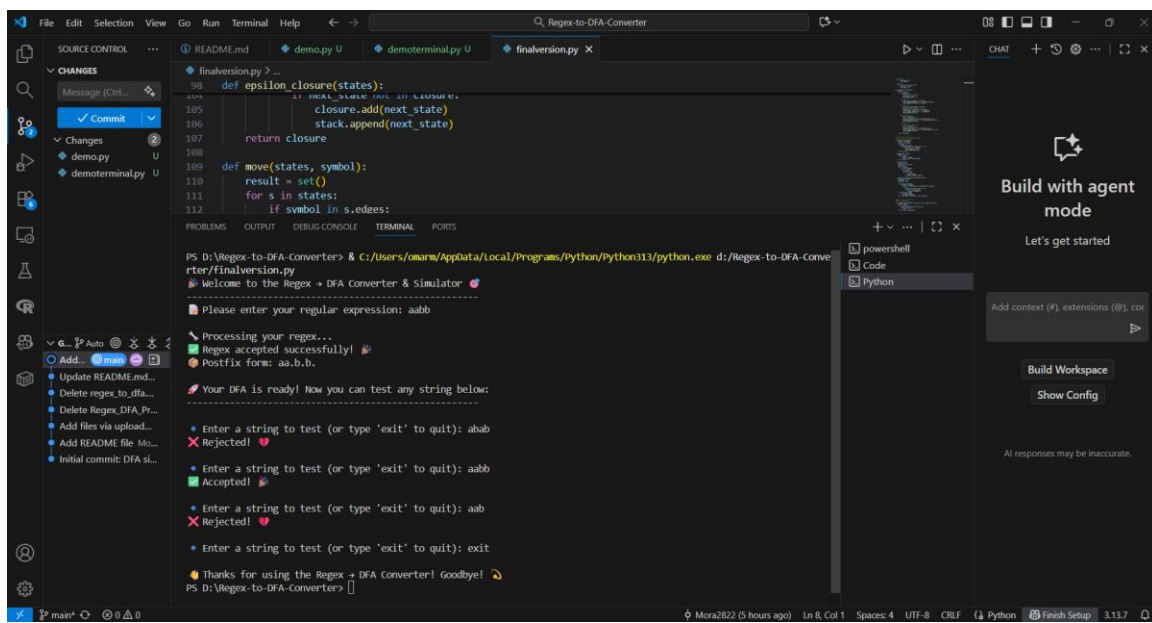
The system performs the full theoretical process of constructing finite automata:

1. Converts a given Regex into postfix form using operator precedence.
2. Builds an NFA using Thompson's Construction algorithm.
3. Applies Subset Construction to convert the NFA to DFA.
4. Simulates input strings through the DFA and determines acceptance.

# Programming Languages, Tools & Libraries

- Python
- Collections & Math Libraries

## Screenshots & Output Samples



The screenshot displays a Visual Studio Code workspace titled "Regex-to-DFA-Converter". The editor shows a file named `finalversion.py` with the following Python code:

```
98 def epsilon_closure(states):
99     closure = set()
100     closure.add(next_state)
101     stack.append(next_state)
102     while stack:
103         state = stack.pop()
104         for s in s.edges:
105             closure.add(s)
106             stack.append(s)
107     return closure
108
109 def move(states, symbol):
110     result = set()
111     for s in states:
112         if symbol in s.edges:
```

The terminal window shows the execution of the script using Python 3.13. The output is as follows:

```
PS D:\Regex-to-DFA-Converter> & C:/Users/omam/AppData/Local/Programs/Python/Python313/python.exe d:/Regex-to-DFA-Converter/finalversion.py
Welcome to the Regex to DFA Converter & Simulator
Please enter your regular expression: aabb
Processing your regex...
Regex accepted successfully!
Postfix form: aa.b.b.
Your DFA is ready! Now you can test any string below:
Enter a string to test (or type 'exit' to quit): abab
Rejected!
Enter a string to test (or type 'exit' to quit): aabb
Accepted!
Enter a string to test (or type 'exit' to quit): aab
Rejected!
Enter a string to test (or type 'exit' to quit): exit
Thanks for using the Regex to DFA Converter! Goodbye!
PS D:\Regex-to-DFA-Converter>
```

The interface also includes a Source Control panel on the left showing changes to `README.md`, `demo.py`, and `demoterminal.py`. On the right, there is a "Build with agent mode" sidebar with options like "Let's get started", "Build Workspace", and "Show Config".