Theory of Computation

NFA TO DFA Converter Using React

By:

Anurag Ghosh

Ishan Kulkarni

Akshad Thole

About the Project

Problem Statement

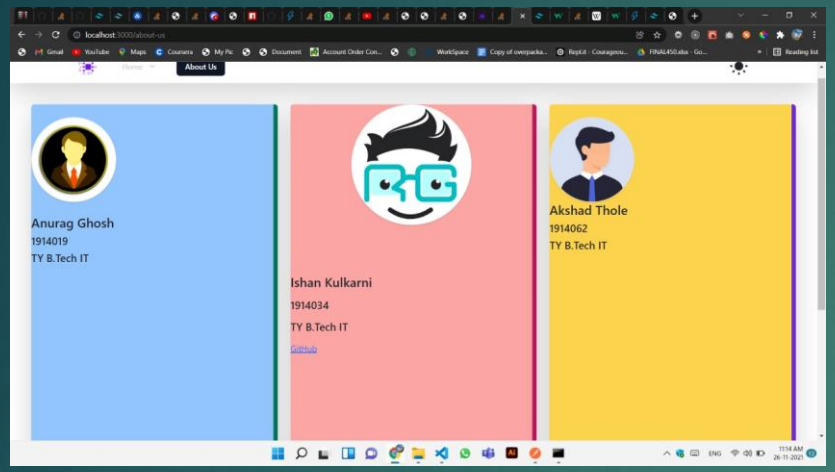
- Website designed to accept the input about the states and transtions of a Non-Deterministic Automata (NFA) and to convert it into an equivalent Deterministic Finite Automata (DFA) and visualize it.

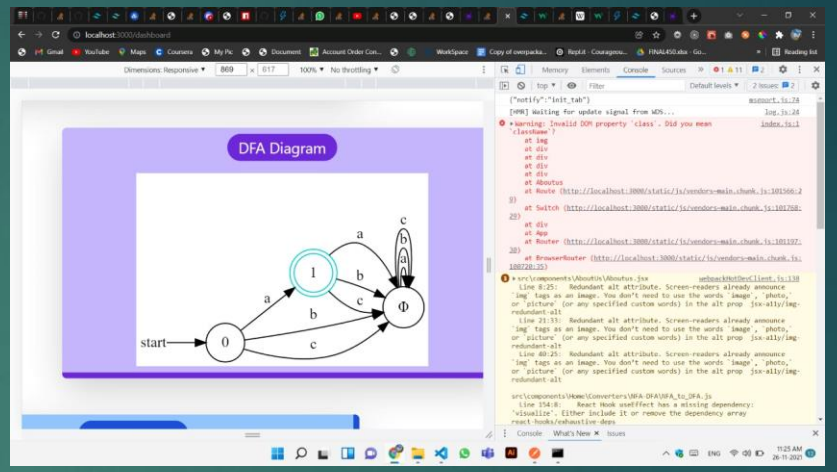
Getting Started with the Project

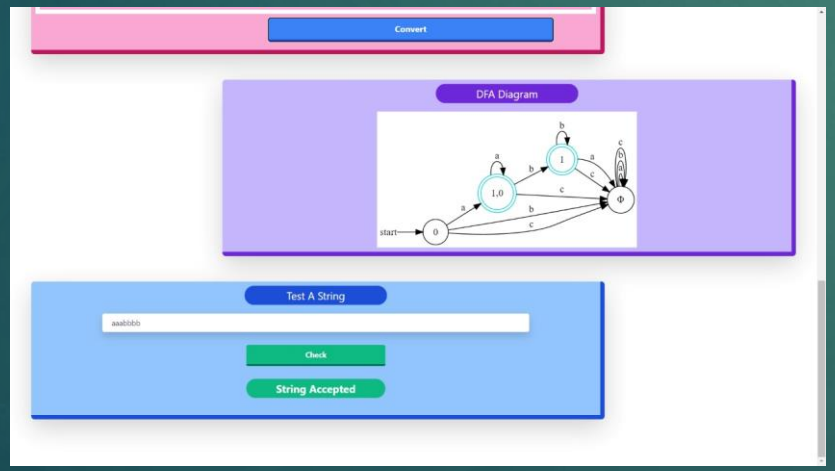
- These instructions will get you a copy of the project up and running on your local machine for development and testing purposes.
- **Prerequisites**
  - React
  - Tailwind CSS
- **Installing**
- A step by step series of examples that tell you how to get a development env running.
- Clone this repository
- Open command line in the cloned folder,
  - To install dependencies, run npm install
  - To run the application for development,
    - Run npm start in the root folder to start the backend app
    - Then run npm start in the client folder to start the frontend app
- Open localhost:3000 in the browser

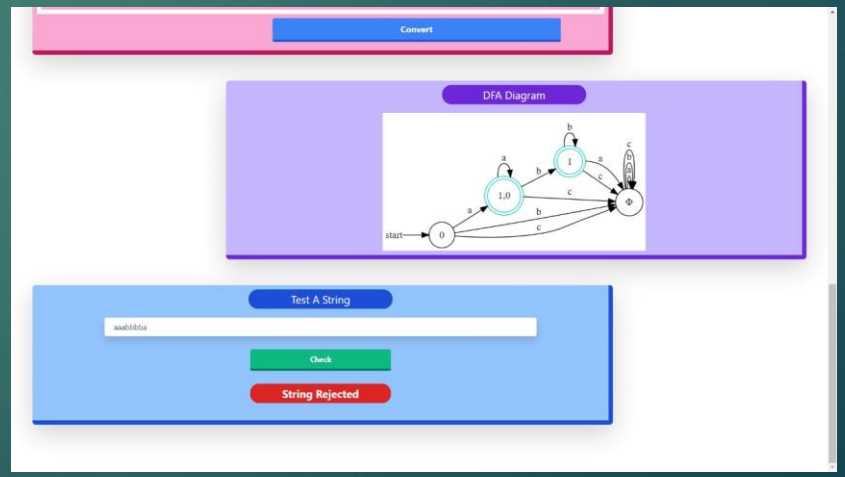
- **Built Using**
- React - Web Framework
- Tailwind CSS - CSS Framework
- GraphViz API - Library for visualization of the DFA
- quickchart.io - API for visualization
- Packages used:
  - Npm package
- Code Base:
  - GitHub: https://github.com/kulkarniishan/PWA-NFA-to-DFA-converter
- Deployment:
  - https://kulkarniishan.github.io/PWA-NFA-to-DFA-converter/

Project Screenshots:









What is NFA and DFA?

- **DFA stands for Deterministic Finite Automata**. :
  - A Finite Automata(FA) is said to be deterministic, if corresponding to an input symbol, there is single resultant state i.e. there is only one transition.
  - A deterministic finite automata is set of five tuples and represented as,
  - $M=\{ Q , \sum , \delta , q_0 , F \}$

Where,
Q: A non empty finite set of states present in the finite control(qo, q1, q2, …).
Σ: A non empty finite set of input symbols.
δ: It is a transition function that takes two arguments, a state and an input symbol, it returns a single state.
qo: It is starting state, one of the state in Q.
F: It is non-empty set of final states/ accepting states from the set belonging to Q.

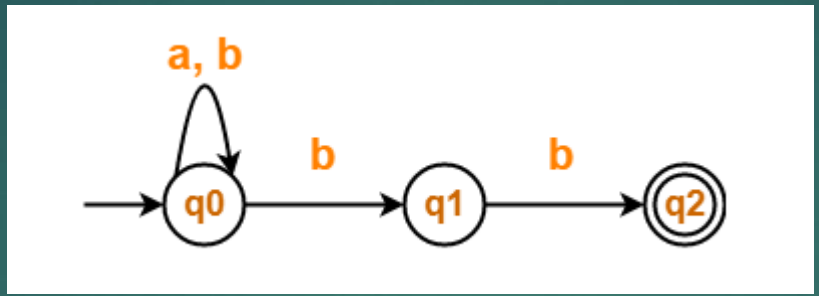
- **NFA stands for Nondeterministic Finite Automata**.:
  - Finite Automata(FA) is said to be non deterministic, if there is more than one possible transition from one state on the same input symbol.
  - A non deterministic finite automata is also set of five tuples and represented as,
  - $M=\{ Q , \sum , \delta , q_0 , F \}$

Where,
Q: A set of non empty finite states.
Σ: A set of non empty finite input symbols.
δ: It is a transition function that takes a state from Q and an input symbol from and returns a subset of Q.
qo: Initial state of NFA and member of Q.
F: A non-empty set of final states and member of Q.

Conversion from NFA to DFA

- An NFA can have zero, one or more than one move from a given state on a given input symbol. An NFA can also have NULL moves (moves without input symbol). On the other hand, DFA has one and only one move from a given state on a given input symbol.
- **Conversion from NFA to DFA**
  Suppose there is an NFA N < Q, ∑, q0, δ, F > which recognizes a language L. Then the DFA D < Q′, ∑, q0, δ′, F′ > can be constructed for language L as:
  Step 1: Initially Q′ = ϕ.
  Step 2: Add q0 to Q′.
  Step 3: For each state in Q′, find the possible set of states for each input symbol using transition function of NFA. If this set of states is not in Q′, add it to Q′.
  Step 4: Final state of DFA will be all states with contain F (final states of NFA).

- Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA)



- **Solution-**

Transition table for the given Non-Deterministic Finite Automata (NFA) is-

State / Alphabet	a	b
→q0	q0	q0, q1
q1	–	*q2
*q2	–	–

- **Step-01:**

Let Q' be a new set of states of the Deterministic Finite Automata (DFA).

Let T' be a new transition table of the DFA.

- **Step-02:**

Add transitions of start state q0 to the transition table T'.

State / Alphabet	a	b
→q0	q0	{q0, q1}

- **Step-03:**

New state present in state Q' is {q0, q1}.

Add transitions for set of states {q0, q1} to the transition table T'.

State / Alphabet	a	b
→q0	q0	{q0, q1}
{q0, q1}	q0	{q0, q1, q2}

Step-04:

New state present in state Q' is {q0, q1, q2}.

Add transitions for set of states {q0, q1, q2} to the transition table T'.

State / Alphabet	a	b
→q0	q0	{q0, q1}
{q0, q1}	q0	{q0, q1, q2}
{q0, q1, q2}	q0	{q0, q1, q2}

Step-05:

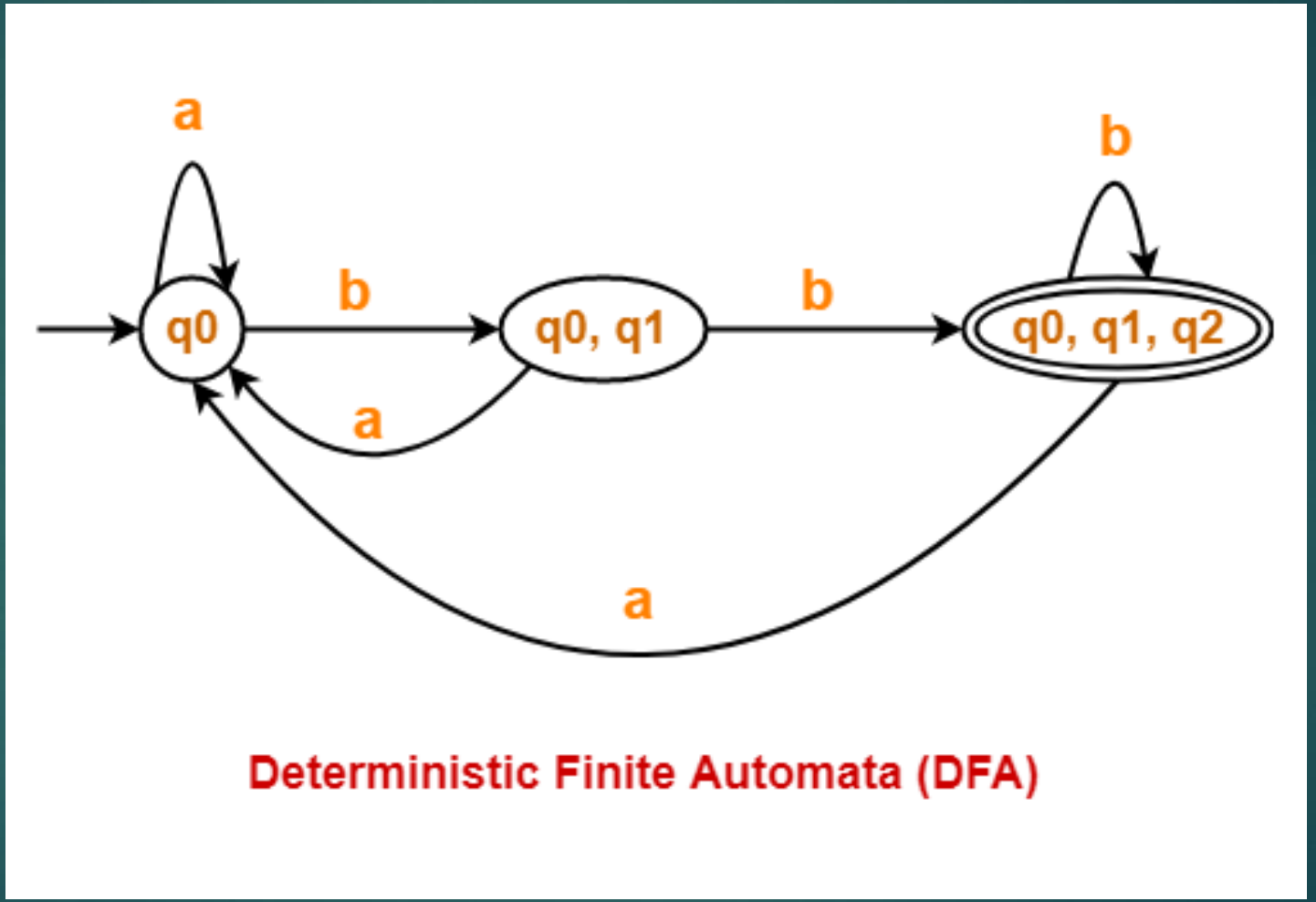
Since no new states are left to be added in the transition table T', so we stop.

States containing q2 as its component are treated as final states of the DFA.

Finally, Transition table for Deterministic Finite Automata (DFA) is-

State / Alphabet	a	b
→q0	q0	{q0, q1}
{q0, q1}	q0	*{q0, q1, q2}
*{q0, q1, q2}	q0	*{q0, q1, q2}

- Now, Deterministic Finite Automata (DFA) may be drawn as-



Deterministic Finite Automata (DFA)