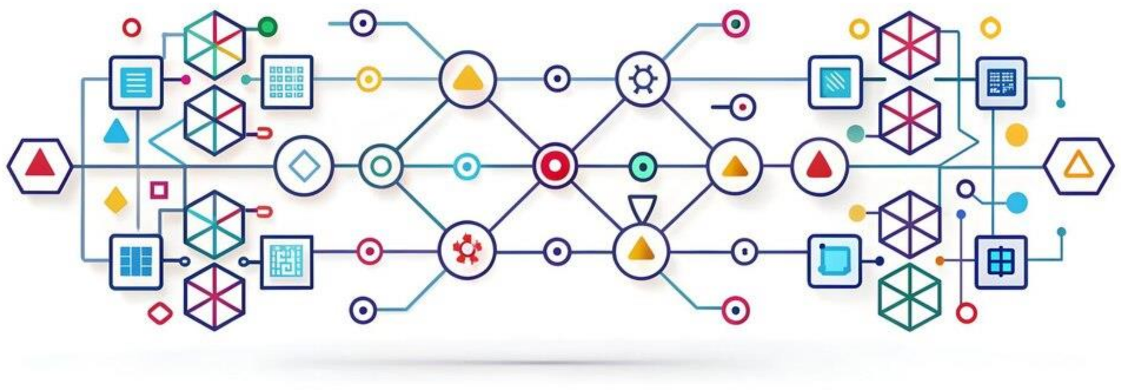
Project One

CSCI419 – Theory of Computing

Report
NFA to DFA Converter

subset construction method



Team members:

-	Omar Tawfik	221000307
-	Shahd Elkazzaz	221000923
-	Mariam Abdulbary	221000919
-	Radwa Ahmed	221000674
-	Moslim Eldawi	221000010

Project Description

This project implements a tool that converts a **Non-deterministic Finite Automaton (NFA)**, possibly with ε-transitions, into a **Deterministic Finite Automaton (DFA)** using the **subset construction method**. The core functionality involves computing ε-closures, handling transitions based on input symbols, and generating a visual and tabular representation of the equivalent DFA.

What Does It Do?

- Takes user-defined NFA transitions and final states as input.
- Computes ε-closures and reachable states.
- Constructs the equivalent DFA using subset construction.
- Generates a **transition table**, **textual DFA description**, and **visual diagram** of the DFA.
- Saves the results into DFArepresentation.txt and an image DFAvisualization.png.

Input Format

- Number of NFA states.
- Transitions for each state (input symbol and corresponding end states).
- Final states of the NFA.

Output Format

- ε-closures for each state.
- DFA transition table (printed and saved).
- Final states and start state of DFA.
- A .txt file (DFArepresentation.txt) and an image (DFAvisualization.png) representing the DFA.

Inside Mechanism

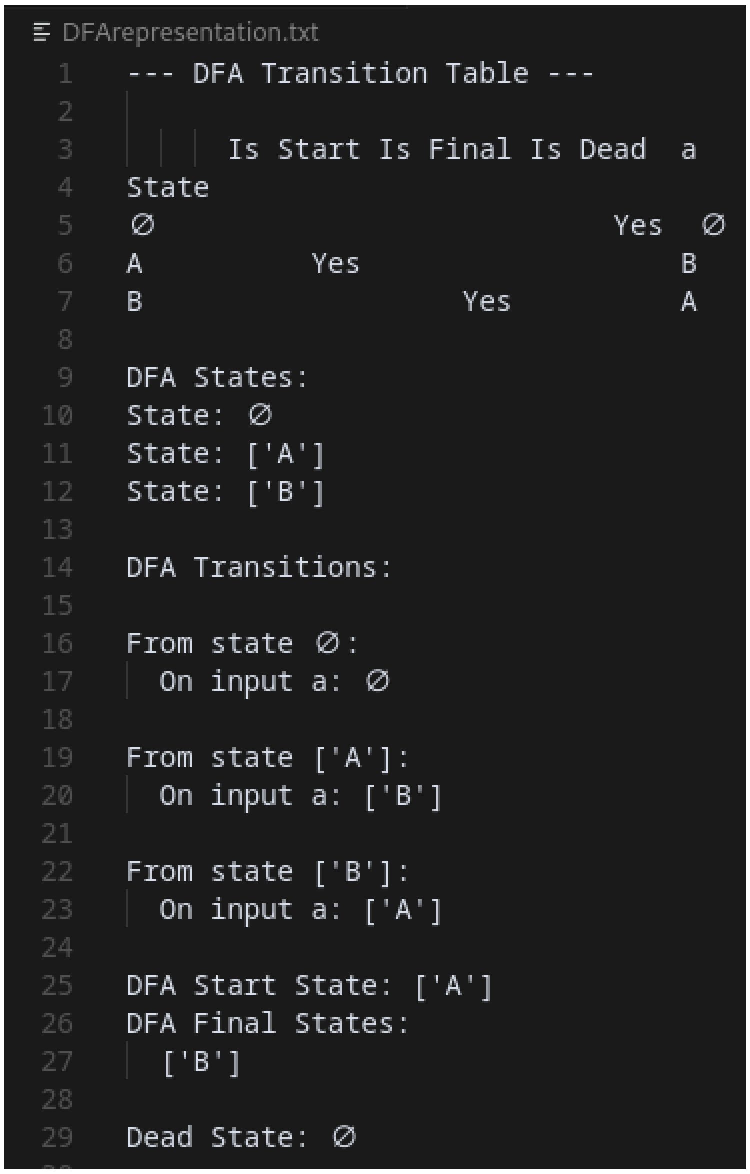
- **computeEpsilonClosure**: Recursively collects all states reachable via ε-transitions.
- **computeTransition**: Computes the resulting states from a set of states on a given symbol.
- **createDFA**: Implements the subset construction method by iterating over power sets of state combinations and tracking transitions.
- **visualizeDFA**: Uses Graphviz to render the DFA as a state diagram.
- **createDFATable**: Uses pandas to format the DFA transitions into a readable table.

Programming Language, Tools & Libraries Used

- **Programming Language**: Python 3
- **Libraries**:
  - pandas – For tabular transition representation
  - graphviz – For state diagram visualization
  - itertools – For powerset generation
  - os – For file and environment operations (optional use)

Project Output Screenshot

1. Sample DFA Transition Table



DFArepresentation.txt

```
--- DFA Transition Table ---

        Is Start Is Final Is Dead  a
State
∅                              Yes  ∅
A           Yes                     B
B                    Yes            A

DFA States:
State: ∅
State: ['A']
State: ['B']

DFA Transitions:

From state ∅:
  On input a: ∅

From state ['A']:
  On input a: ['B']

From state ['B']:
  On input a: ['A']

DFA Start State: ['A']
DFA Final States:
  ['B']

Dead State: ∅
```

2. DFA Visualization