INSTITUTE OF TECHNOLOGY OF CAMBODIA
Department of Information Technology Communications Engineering

Design Finite Automation

LECTURER

VALY DONA

SUBJECT

AUTOMATA THEORY

N0	Name	ID	Gender
1	TOM TITO	e20200061	M
2	VEASNA DARA	e20200268	M
3	TY SOPHEAKTRA	e20200958	M
4	SORPORN SOVORTEY	e20200988	F
5	TEANG SREYROTH	e20200826	F

LINK

Contents

Introduction

- The purpose of this project is to practice the theory of the Finite Automata that we have learned in lesson 1 to lesson 5.

- The project aimed to develop a application that allows users to work with Finite Automata, perform various operations such as check Deterministic, convert NFA to DFA and minimize DFA, and save the automata data to a MySQL database.

Function and Features

The developed application, named "**Finite Automaton App**" offers the following key functions and features:

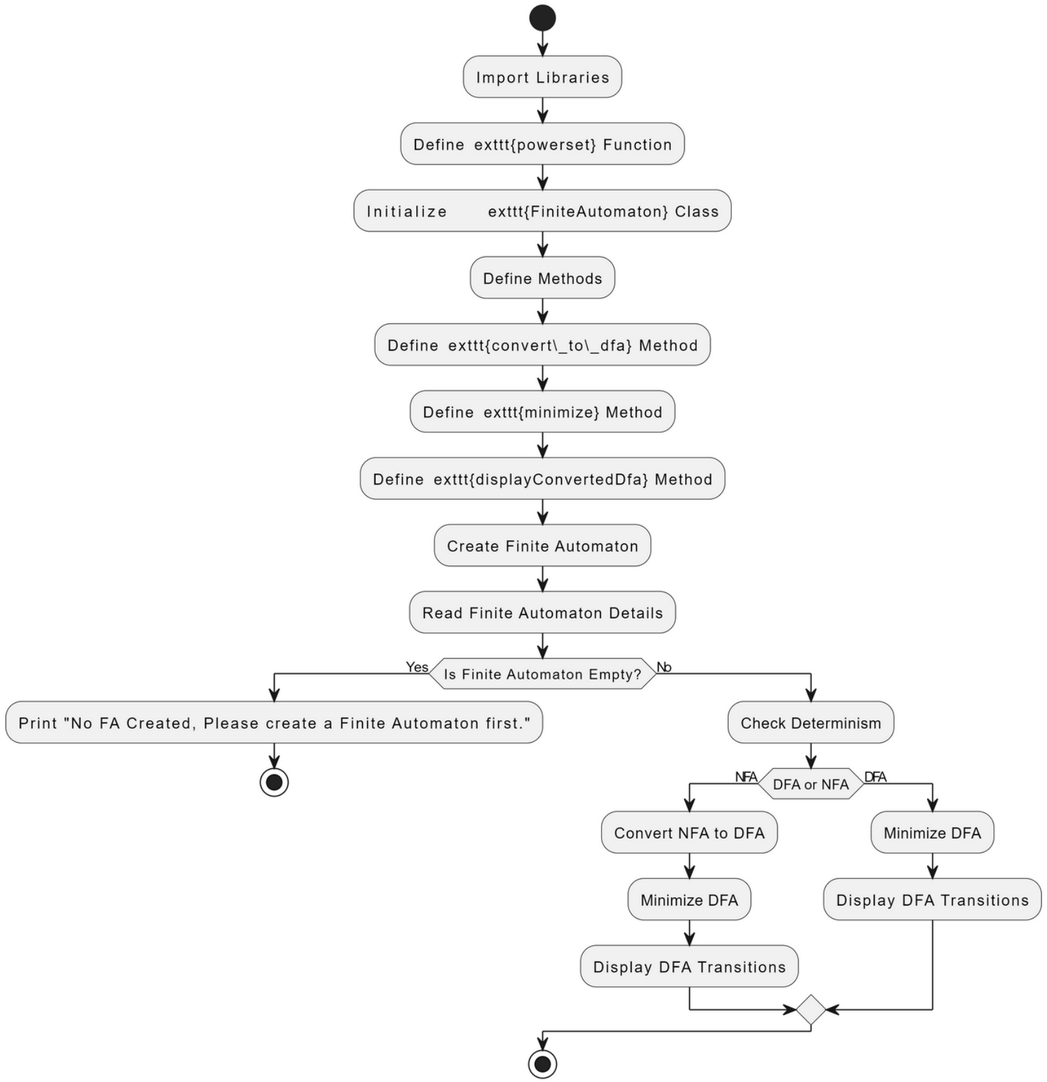
1. Create Finite Automata
2. Check Deterministic
3. Check Acceptance
4. Convert NFA to DFA
5. Minimize DFA
6. Save and Load to Database
7. Design UI

Data Structure

The primary data structure used in the project is the Finite Automaton class. The class encapsulates the necessary attributes and methods to represent a finite automaton. The key attributes of the Finite Automaton class include:

- States: A set to store the states of the automaton.
- Alphabet: A set to store the symbols of the alphabet.
- Transitions: A dictionary to represent transitions between states and symbols.
- Initial_state: A single state representing the initial state of the automaton.
- Accepting_states: A set to store the accepting states of the automaton.

Data Structure



Database Design

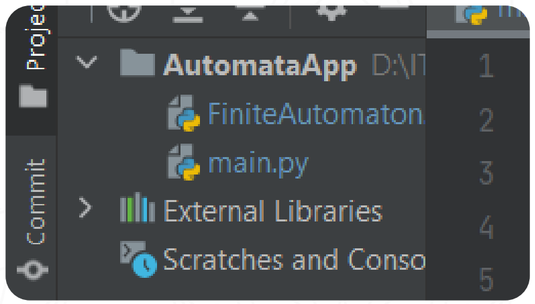
The application utilizes a MySQL database to store and manage Finite Automata data. The database design consists of a single table named "data," which stores the information for each Finite Automaton. The table includes the following columns:

- Id: An auto-incrementing primary key to uniquely identify each record.
- State: A varchar column to store the states of the automaton as a comma-separated string.
- Alphabet: A varchar column to store the alphabet symbols as a comma-separated string.
- Transition: A varchar column to store the transitions in the format "from_state , symbol, to_state " for each row.
- Initial: A varchar column to store the initial state of the automaton.
- Accepting_state : A varchar column to store the accepting states as a comma-separated string.

Implementation

The application was implemented using Python programming language and the Tkinter library for the graphical user interface.

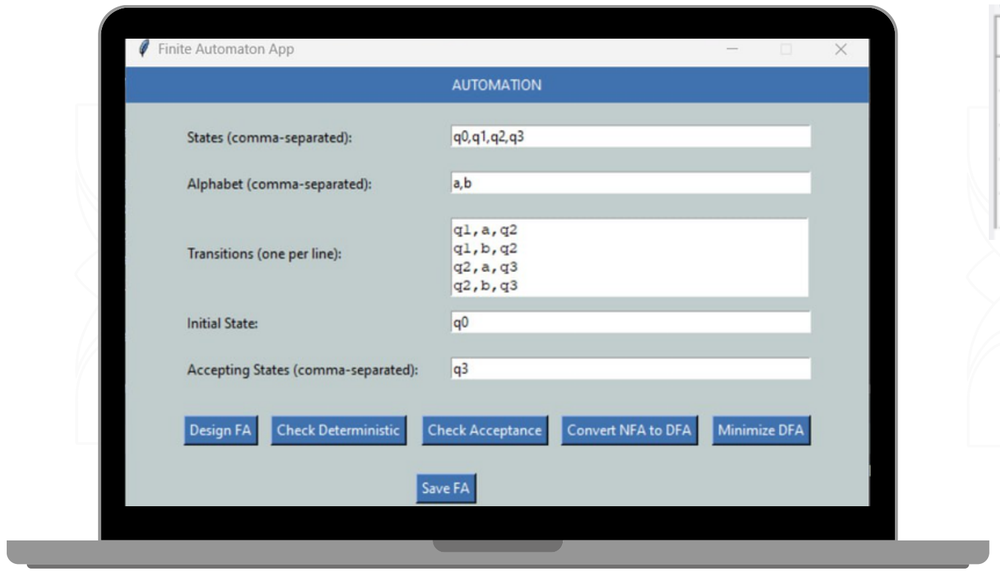
1. Project Structure



2. Class
   - Finite Automaton
   - Finite AutomatonApp

Result

The Finite Automaton App successfully achieved the project objectives. Users can create finite automata, check their determinism, acceptance of strings, convert NFA to DFA, and minimize DFA. The application allows users to save the finite automaton data to a MySQL database for future reference.



id	state	alphabet	transition	initial	accepting_state
1	q0,q1	a,b	q0,a,q1 ...	q0	q1
2	q0,q1	a,b	q0,a,q1 ...	q0	q1
3	q0,q1,q2,q3	a,b	q0,a,q0 ...	q0	q3
4	q0,q1,q2,q3	a,b	q0,a,q0 ...	q0	q3
NULL	NULL	NULL	NULL	NULL	NULL

Conclusion

Conclusion

In conclusion, the Finite Automata project provided valuable insights into the practical applications of automata theory. The project team successfully collaborated and developed a functional application with a user-friendly interface.

Perspective

Overall, the project's successful completion demonstrates the team's ability to apply theoretical concepts to practical solutions and work cohesively on a software development project. The Finite Automaton App presents an exciting opportunity for future expansion and refinement in automata theory and related domains.

Demonstration