COMP SCI 4TB3 Final Project

1. Overview

For this task, we will be creating simple animated graph visualization for the execution of algorithms in an interactive manner.

User will have the ability to select type of model, color and shape for the graph visualization. Type of model includes Finite State Machine, state minimization and the meaning of sentences. It will be based on Python Turtle library for the graphing part and on ImageMagick for the animated gif generation.

Furthermore, we could also provide the visualization of the algorithm for making FSM deterministic and equivalence.

2. Type of Models

2.1 Finite State Machines

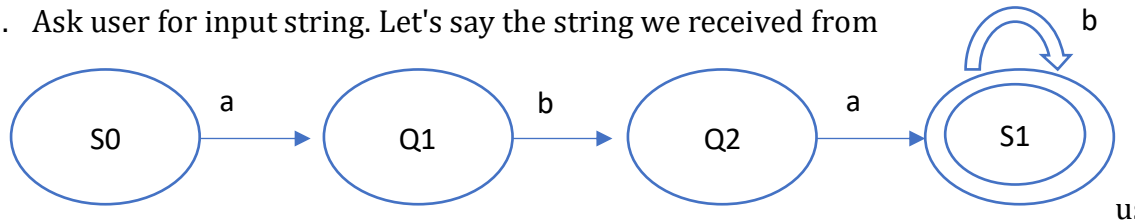
A deterministic finite state machine or acceptor deterministic finite state machine is a quintuple(Σ,S,s0,δ,F),where:

- Σ is the input alphabet (a finite, non-empty set of symbols).
- S is a finite, non-empty set of states .
- Q is states.
- s0 is an initial state, an element of S.
- δ is the state-transition function: $\delta : S \times \Sigma \rightarrow S$
- s1 is a final state, an element of S.

The following example illustrates the execution of the finite state machine F.

We have a finite state machine F denoted by the graph below:

1. Ask user for input string. Let's say the string we received from user is **"abab"**



2. Ask user for the color. Let's say the color user chosen is red. If the user does not select anything, the color will be randomly generated by the program.
3. illustration

1. F reads the first symbol in string, which is "a".
2. After reading the symbol "a", FSM F has transitioned from state s0 to state q1. Then F reads the next symbol, which is "b".
3. After reading the symbol "b", FSM F has transitioned from state s0 to state q1. Then F reads the next symbol, which is "a".
4. After reading the symbol "a", FSM F has transitioned from state q1 to state the accepting state S1. Then F reads the next symbol, which is "b".
5. The input string is now empty, the finite state machine has no more symbols to read and its current state is an accepting state, therefore the finite state machine F accepts the input string.

2.2 Meaning of Sentences

We want to recognize the meaning of sentences with vocabulary and syntax: These sentences should start with "You are" followed by

- an adjective or
- the word "not" followed by an adjective.

e.g. "You are pretty" → positive meaning, "You are ugly" → negative meaning,

"John is tired" → error

1. Ask the user to enter the string in the format assigned. The user entered "You are pretty"
2. Ask user for the color. Let's say the color user chosen is red. If the user does not select anything, the color will be randomly generate by the program.
3. Illustration
    1. F reads the first word in the string, which is "You"
    2. After reading "You", F has transitioned from state s0 to q1. Then F reads the next word, which is "are".
    3. After reading "are", F has transitioned from state q1 to S1. Then F reads the next word, which is "pretty".
    4. Do a sentiment analysis on the final word entered by user. "Pretty" is a positive word, so the result is positive
    5. The finite state machine has no more symbols to read and its current state is an accepting state, therefore the finite state machine F accepts the input string.

2.3 State Minimization

We will apply state minimization to eliminate the states we are unable to reach and redundant state.

3. Documentation

This project will be documented and demostrated by Jupyter notebooks throughout the development period. We hope to learn more about the execution and algorithms of FSM by completing this task. Furthermore, it also helps us to consolidate our knowledge on the topics and materials learned throughout the course.

4. Resources

- GIF Animation
  ImageMagick Studio LLC. "Convert, Edit, or Compose Bitmap Images." ImageMagick, imagemagick.org/index.php.
- Pygame
  "Drawing Objects and Shapes in PyGame." Python Programming Tutorials, pythonprogramming.net/pygame-drawing-shapes-objects/.
- Turtle
  Abraham. "Animation with Turtle Graphics." Animation with Turtle Graphics, 10 Sept. 2018, learn.wecode24.com/animation-with-turtle-graphics/.
- State Minimization
  "DFA Minimization." Tutorialspoint, www.tutorialspoint.com/automata_theory/dfa_minimization.htm.
- Generalized Parsing
  Qi, Siyuan. A Generalized Earley Parser for Human Activity Parsing and Prediction. https://vcla.stat.ucla.edu/Temp/PAMI2019_Prediction.pdf.

5. Division of Work

Team Member	Task
Jessia	poster design, meaning of sentences, state minimization
Mingnan	poster content, pygame animation code, FSM parsing algorithms

6.Weekly Schedule

Deadline	Task
Mar. 9	pygame animation code
Mar. 16	generalized pygame code
Mar. 23	utilize pygame code into FSM parsing
Mar. 31	finalize project and bug fixes
Apr. 7	poster design and content