Creating a Finite Automation Simulator and Visualiser

CS4099 - Senior Honours Project

James Hart

210010341

Finite State Automaton

Deterministic

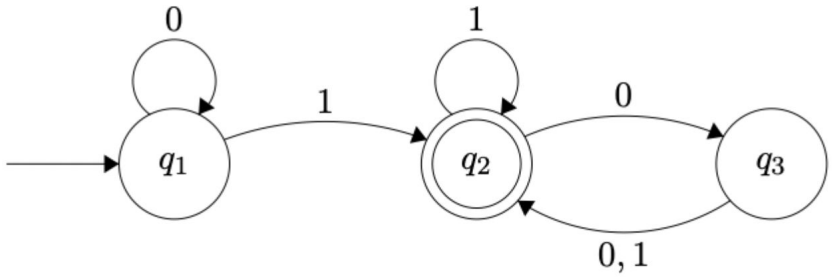


Figure: A finite automaton M_1 with 3 states [Sipser, Figure 1.4]

Nondeterministic

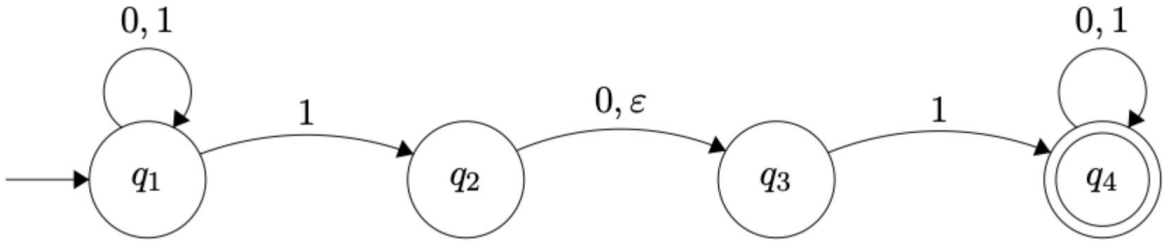
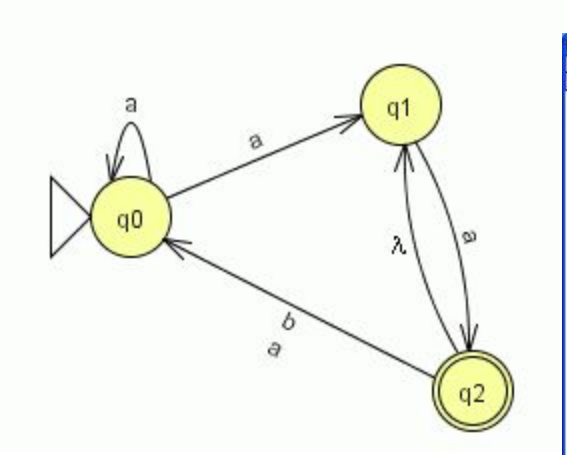


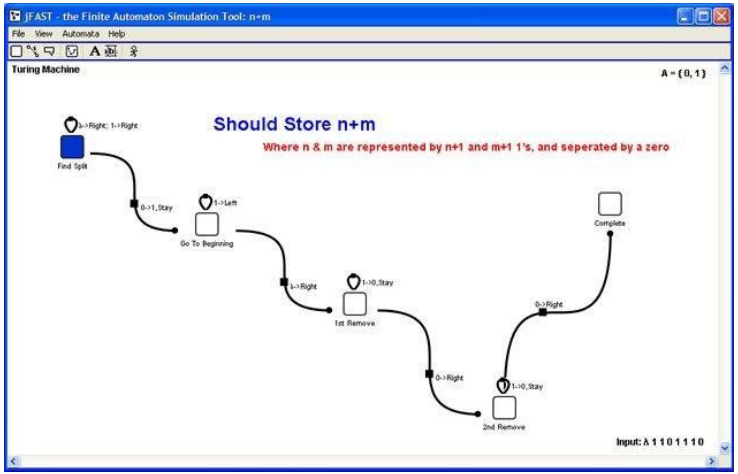
Figure: A nondeterministic finite automaton N_1 [Sipser, Figure 1.27]

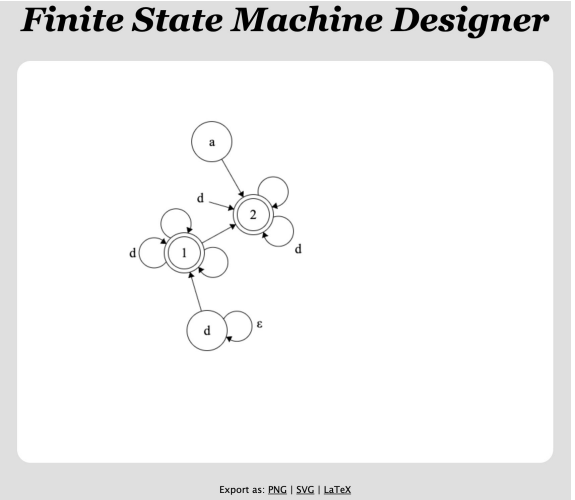
[1]
M. Young, “CS3052 Computational Complexity Week 1: Regular languages,” *St-andrews.ac.uk*, 2021. https://studres.cs.st-andrews.ac.uk/2023_2024/CS3052/Lectures/W01_Regular_languages.pdf (accessed Apr. 17, 2025).

[2]
M. Sipser, *Introduction to the theory of computation*. Boston: Course Technology, 2020.

The Problem







[3]
S. Rodger and T. Finley, “JFLAP -An Interactive Formal Languages and Automata Package,” 2005. Accessed: Mar. 19, 2025. [Online]. Available: https://www.jflap.org/jflapbook/jflapbook2006.pdf

[4]
T. M. White and T. Way, “jFAST,” *ACM SIGCSE Bulletin*, vol. 38, no. 1, Mar. 2006, doi: https://doi.org/10.1145/1121341.1121460.

[5]
E. Wallace, “Finite State Machine Designer - by Evan Wallace,” *madebyevan.com*, 2010. https://madebyevan.com/fsm/ (accessed Apr. 13, 2025).

Agreed Objectives

Primary - The application should:

- Allow users to create finite automata with their diagrams graphically displayed
- Allow users to choose and manipulate the exact positions of nodes
- Verify that the automaton is valid
- Allow for both deterministic and non-deterministic finite state automata
- Provide users with a way to automatically reconfigure the layout of the diagram to be in an organised and presentable manner, suitable for lectures
- Allow users to export the finite automata diagram in at least one known image format
- Allow the user to simulate an input word on the finite automata by displaying its path, output and whether the word was accepted or rejected

Secondary - The application may:

- Allow users to control the exact angle of edges between nodes
- Implement additional exporting formats such as SVG, PNG and LaTeX
- Utilise animations when simulating an input word to clearly show the path taken
- Allow users to save a video file of the simulation
- Allow users to create their finite automata configuration through text entry, which uses a specific language to communicate the automaton's formal definition
- Have a high level of usability by staff and students of Computer Science
- Perform automated testing of all parts of the program, including the user interface

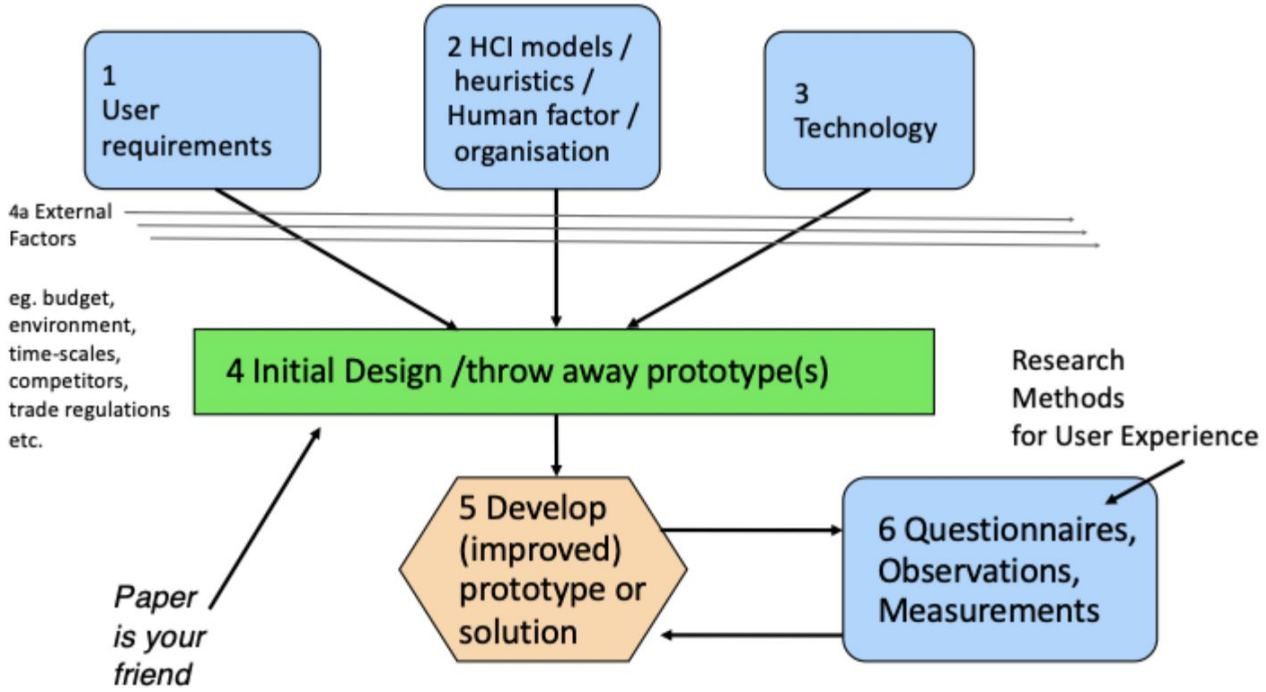
What Was Achieved

All primary objectives were fully achieved.

Secondary Goals:

- Fully Achieved
  - Additional exporting formats
- Partially Achieved
  - Automated UI testing on all parts of application - **84.11% Code Coverage**
  - Highly usable for end-users - **Two different control schemes after survey**
- Not Achieved
  - Text entry
  - Animations and video file
  - Editing angles of transitions

Interaction Design Process



[6]
A. Miguel, “CS3106 Human Computer Interaction Week 2 Interaction Design,” *St-andrews.ac.uk*, 2023. https://studres.cs.st-andrews.ac.uk/2023_2024/CS3106/Lectures/Week%202/CS3106%202024%20Week%202%20Interaction%20Design.pdf (accessed Mar. 23, 2025).