Project Title: Improved Visualization for Formal Languages

Group Members: Chris Pinto-Font (cpintofont2021@my.fit.edu), Vincent Borrelli (vborrelli2022@my.fit.edu), Andrew Bastien (abastien2021@my.fit.edu), Keegan McNear (kmcnear2022@my.fit.edu)

Faculty Advisor: Dr. Luginbuhl (dluginbuhl@fit.edu)

Client: Dr. Luginbuhl

Meeting time(s): Bi-weekly Tus

Goal & Motivation: The goal is to design and program a user-friendly application for visualizing Deterministic Finite Automata (DFA). The primary motivation is to create an effective teaching and learning aid that allows users to accurately plot DFAs and animate their execution. This project aims to replace an outdated application with a more intuitive interface and features requested by the client, such as step-by-step tracing.

Approach: Key Features:
Intuitive Creation: A GUI editor allowing users to plot DFAs, designate specific states (initial, final, dead), and track transitions based on grammatically valid strings.
Animation: A visual playback system that shows step-by-step formation and execution of DFAs, which aids the understanding of logic flow.
Minimization (Complete): The system allows for the minimization of DFAs that have additional and unnecessary states and transitions.
Improved GUI: A GUI that is seamless and easy to use, such as a toolbox interface that would allow for hover over for tooltips. As well as shortcuts integrated into the system.

Novel Features/Functionalities:
Multiple symbols on transition: To be able to place multiple symbols on a single transition and remain readable.
Integrated Documentation: In program documentation toolboxes that allow users to learn about the software without needing to locate and open external “read me” files.
Interactive Canvas: A drag-and-drop canvas system that integrates with the text-based backend.

Algorithms & Tools:

Language: Python is the primary language used for the software logic.
GUI Framework: Using wxGlade and Python for the graphical user interface.
Minimization Algorithm: The distinguishability algorithm is used. This would take unordered pairs of states, which would mark these pairs where one is accepting and the other is not, while iteratively making pairs based on transition outputs until no new marks can be made. The unmarked pairs are combined in the end.

Notes: (1/20/26)
Nice to have: Turn NFA to DFA (can't minimize an NFA, then provide info and optional ways to convert NFA to DFA) or just hardlock DFA only
Hardcoded demo (pre-record a demo)
Include Lambda only for NFAs (low priority)
Undoable minimization
Depth first to traverse nondeterminism during animation, backtracking would work for animation (or go through all paths for animation, each has its own color)

Technical Challenges:

1. **Design: system architecture diagram**
    - Animation Implementation: The team has to work on improving animation.
    - Minimization Logic: Implementing the minimization logic was a little on the rougher side. We were able to grasp the concept well enough to implement it, but we could make it more efficient and guarantee its correctness.
2. **Evaluation: how to measure success? Some ideas:**
    - Speed: The software achieves fast enough speeds that occur almost instantly.
    - Accuracy: All DFAs have been correct so far, including minimization.
    - Reliability: The software does achieve the goal.

3. **Progress Summary:**

Module/feature	**Completion %**	**To do**
Interactive canvas for graphing	100%	Improve interactivity

Basic animation in graphing space	50%	Improve animation
Tie text based to visual version	80%	Full implementation
DFA minimization	80%	Can be refined
Update “Read Me” file	50%	Needs to be updated

4. **Milestone 4 (Feb 23): itemized tasks:**
    - Implement better animation for visual flow
    - Check minimization and see if it can be improved
    - GUI improvements to make the program more visually appealing
    - Implement lambda if including NFAs
5. **Milestone 5 (Mar 30): itemized tasks:**
    - Complete tutorial mode
    - Animation refined
    - Conduct evaluation and analyze results
    - Create a poster for Senior Design Showcase
6. **Milestone 6 (Apr 20): itemized tasks:**
    - Final polish
    - Test/demo of the entire system
    - Conduct evaluation and analyze results
    - Create user/developer manual
    - Create a demo video

Task	Chris	Vincent	Andrew	Keegan
Implement simple	Bug Fixer/Code	Bug Fixer/Code	Co-Lead coder and	Co-Lead coder and

animations	Contributor and designer	Contributor and designer	development head	development head
Implement minimization	Minimization logic implementation and researcher	Bug Fixer/Code Contributor and researcher	Co-Lead coder and development head	Co-Lead coder and development head
Make the GUI more involved and “cooler”	Bug Fixer/Code Contributor and researcher	Bug Fixer/Code Contributor and researcher	Co-Lead coder and development head	Co-Lead coder and development head
Include more transition types (different alphabets, including lambda)	Logic writer, DFA logic consultant, and bug tester	Logic writer, DFA logic consultant, and bug tester	Co-Lead code side implementor	Co-Lead code side implementor
Refine stability with new features and logic	Bug Fixer/Code Contributor and designer	Bug Fixer/Code Contributor and designer	Co-Lead code side implementor	Co-Lead code side implementor

1. Description (at least a few sentences) of each planned task for Milestone 4:
    - Task 1: Implement animation: We already have simple enough animation to visually show the flow of the DFA creation and transition, however it needs to be worked on to make it more impressive.
    - Task 2: Implement minimization: The implementation of minimization is roughly implemented but it needs to be further refined. As well as implementing the tutorial mode.
    - Task 3: Improve the GUI: We need to make the GUI more visually appealing while keeping the simplicity of use for the user interface.
    - Task 4: Include lambda: We need to add lambda transitions to the NFA logic if we implement NFAs.
    - Task 5: Refined stability: We need to make sure the program does not break when new features are added, and keep it stable for robustness.

2. Approval from Faculty Advisor
    - "I have discussed with the team and approve this project plan. I will evaluate the progress and assign a grade for each of the three milestones."
    - Signature: [illegible] Date: 1/26/26