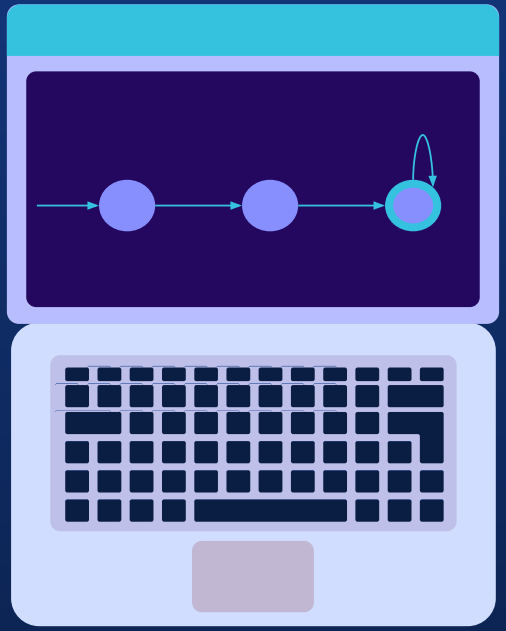
Improved Visualization for Formal Language: **Project Plan 2**



https://kmcnear2022.github.io/

Group Members: Chris Pinto-Font, Vincent Borrelli, Andrew Bastien, Keegan McNear

Who is involved?

Faculty Advisor: Dr. Luginbuhl
Serves the role of academic advisor for the project; overseeing product needs and design goals. Providing guidance in the progression of our project while keeping us on track and focused on our goals.

Client: Dr. Luginbuhl
The genesis for project was based on the needs and preferences of Dr. Luginbuhl, specifically his experiences with other graphing software. His close involvement with this project will allow us to quickly address his user needs as he tests our program regularly.

Goals and Motivations

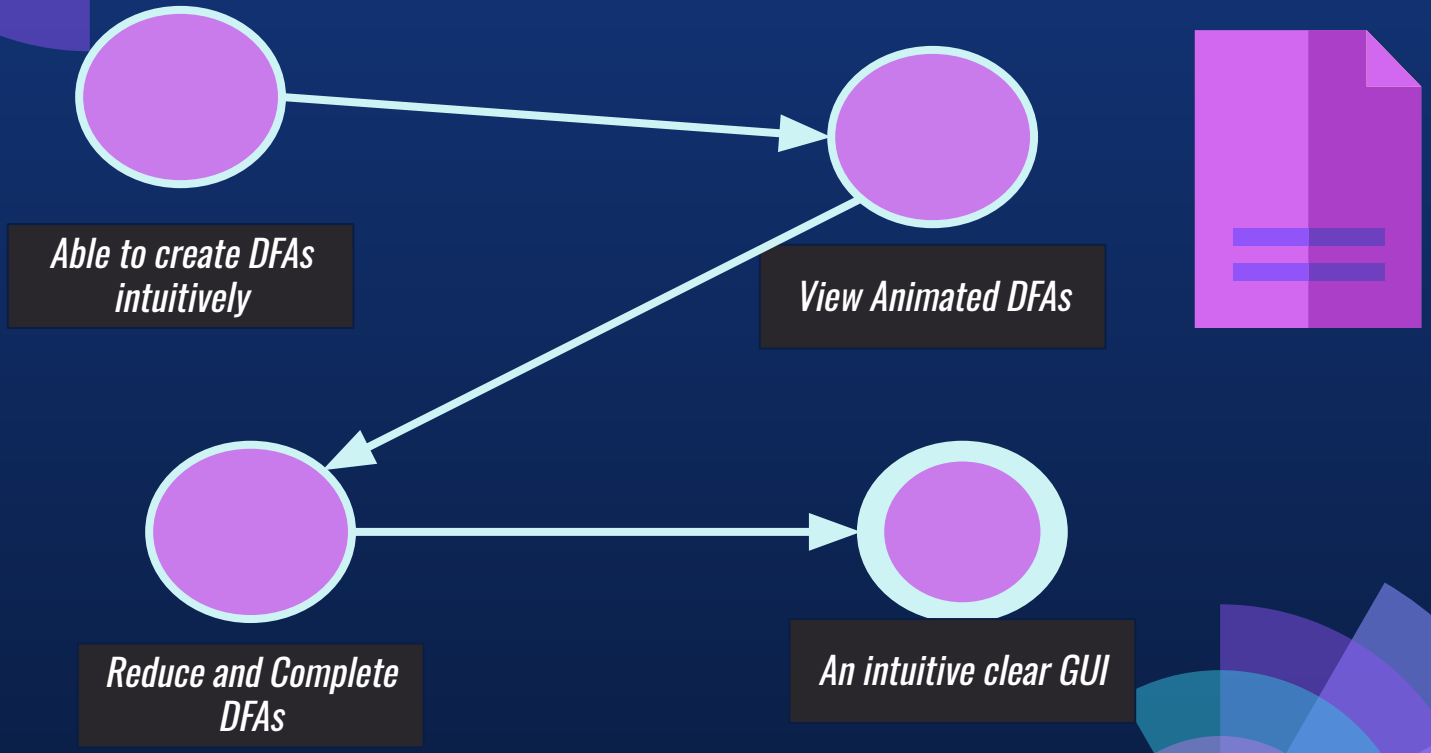
Goals

- Build a Computer Program for both teachers and students
- User Friendly and legible
- Engaging and intuitive animations

Motivation

- Need for tools to better teach and explore DFAs
- Remove undue confusion from the process
- Need to make learning and using DFAs easier

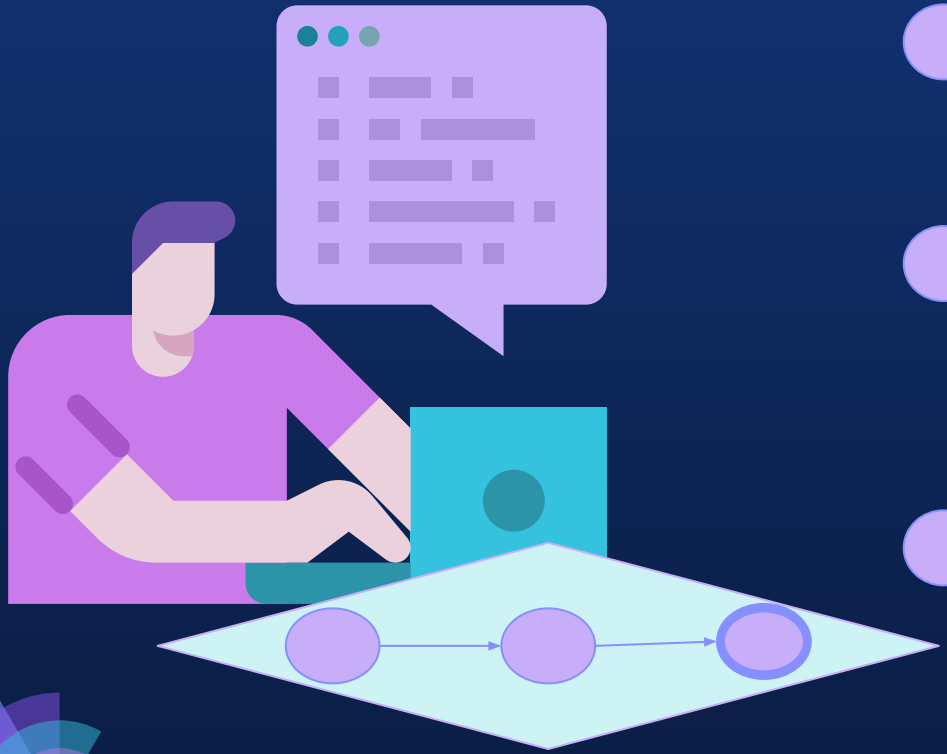
Key Features



Novel Features

- **Multiple symbols on a single transition between arcs:**
  - Interface will let the user make more complex and varied DFAs.
- **Animation of DFAs**
  - DFA building and progression will happen in a step by step animated manner to let the user follow and view the process as it happens.
- **In-Program documentation files**
  - Information for the program will be contained within the app for easy use and navigation.
- **Intuitive structuring to make both teaching and learning DFAs an easier process.**
  - Inspired by DESMOS the creation and tracing of DFAs will be incredibly easy for a new user.

Technical Challenges



Learning Python Library

Building an interactive visual program in Python will require our team to become familiar with what will likely be a complex and unfamiliar python library.

Learning Animation

Application will need to animate DFA building and progression, thus we will need to learn how to properly animate in the GUI space.

Algorithmic Complexity

The process of the program being able to build and complete DFAs based on entered text means the program will need to employ one or more complex algorithms to render out a correct and most ideal DFA.

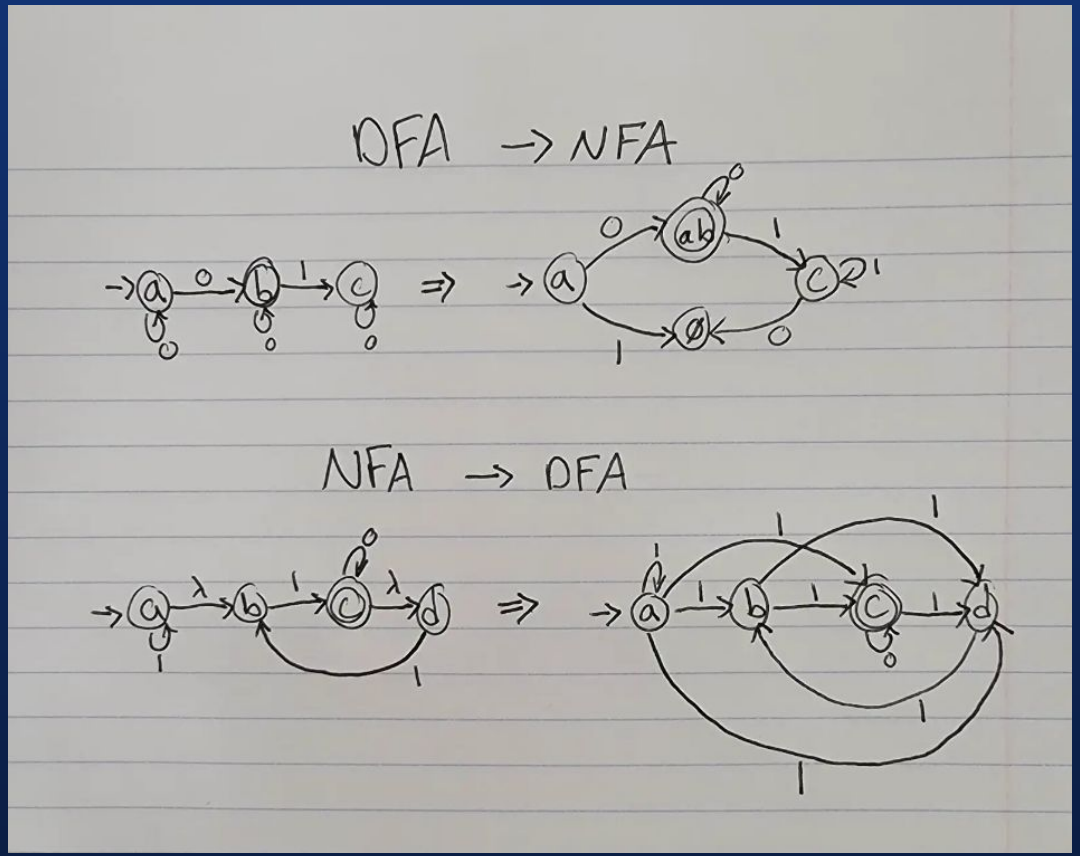
Building a functional and intuitive GUI

The basis for this project being improving the feel and ease of use for a DFA graphing program means we'll need to learn to create a GUI with usable tools and instant graphing capabilities.

Advisor Involvement (Cont)

We met again with Dr. Luginbuhl to further flesh out our goals for the project this semester, and primarily what we wish to accomplish for Milestone 4, primarily the need for us to consider whether to add in functionality to turn NFA into DFAs and to implement a system to discern between the two.

DFA to NFA & NFA to DFA



Potential Features as per advisor discussion

- Animated Process to turn NFAs into reducible DFAs. or at least functionality to detect when the user builds an NFA and prompt for the change/tell them it can't be minimized
    - If NFAs are included in app functionality, lambda transitions should be added
- Hardcoded visual demonstrations of features and app processes
- Reversible minimization (step backwards/state restore)
- Depth first nondeterminism before animations, backtracking within animation. (further expansion of teaching features)

Such features should be considered but aren't the primary focus of development.

Milestone Four features as per last semester

- Refine and expand minimization functions
  - (works in a basic state currently though the animated component is lacking)
- Develop a basic version of a graph builder that takes in a submitted syntactically correct string/equation
- Start developing a teacher mode for interactive DFA building
- Heavily bug test current features and ensure quality of new ones.
- Continue to update the “readme” file with new features as to make sure it maintains completeness.

Further features as per advisor meeting and group reassessment

- Improved GUI appearance and navigability
- Allow the program to discern between NFAs and DFAs and decide if the program will allow for NFA graphing (NFA specific features)
  - If so, expanded transition types to include lambda
- Refine stability
  - The scope of the DFAs the user can make means we need to stress test our algorithms for minimization and traversal

Future Milestone Goals (as per milestone 3/reviewed by advisor)

We hope to utilize Milestones 5 and 6 to refine our core features, program stability, and user experience. Our current scope of features is well realized, though our largest undertaking this semester will be the teaching mode/interactive minimization. Such as system will require bespoke coding to respond and handle every possible DFA (or exist solely in a demo style teaching format using one specific DFA as to teach the mechanics of that system)

Task Matrix

Task	Completion %	Chris	Vincent	Andrew	Keegan	To Do
Improve animation	50%	10%	20%	10%	10%	Have better interactivity, and make it smoother
Adjust minimization	50%	20%	10%	10%	10%	Minimization should be smoother (tutorial mode)
Improve GUI	80%	10%	10%	30%	30%	Adjust GUI to be more modern
Include lambda (NFAs)	0%	0%	0%	0%	0%	Need to implement NFA logic
Refined stability	100%	25%	25%	25%	25%	Make sure everything stays stable and functional

Questions?

Visit Our Site

https://kmcnear2022.github.io/