

Regex to NFA/DFA Converter: Visualizing Regex Magic

Project Documentation

Team: SoloTech

Team Lead: Sameer Rana

Email: sameer.rana.work@gmail.com

Date: June 28, 2025

Project Abstract

- The Regex to NFA/DFA Converter is a React-based web tool designed to transform regular expressions into Non-deterministic Finite Automata (NFA) and Deterministic Finite Automata (DFA), with interactive visualizations. Its main goal is to aid students, educators, and developers in understanding regex patterns by converting them into graphical state machines, enhancing learning and debugging. The project leverages JavaScript, Chart.js, and Viz.js to render dynamic diagrams, offering a user-friendly interface to input regex and view the conversion process. Developed over weeks, it addresses the need for an accessible, visual tool in computer science education. The tool successfully handles basic regex operations, providing immediate feedback through visual outputs. Future enhancements aim to support complex regex and export capabilities, making it a valuable educational resource. This project bridges theoretical concepts with practical application, fostering a deeper grasp of automata theory.

Updated Project Approach and Architecture

- The project adopts a client-side, single-page application (SPA) approach using React for a responsive interface. The architecture centers on a modular design: the `Regex-to-NFA-DFA-Converter.html` serves as the entry point, linking `Regex-to-NFA-DFA-Converter.js` for React components, `Regex-to-NFA-DFA-Converter.js` for core conversion logic, and `Regex-to-NFA-DFA-Converter.css` for styling. Communication is handled via DOM manipulation and event listeners, with no server-side processing required. Libraries include React for state management, Chart.js for initial visualization attempts, and Viz.js for rendering DOT language-based automata graphs. The conversion algorithm parses regex input, builds an NFA using Thompson's construction, and converts it to a DFA via subset construction, all computed client-side. The approach prioritizes simplicity and performance, avoiding external APIs. Future iterations may integrate a backend for complex regex support or WebSocket for real-time collaboration. The current setup is lightweight, hosted on GitHub Pages, ensuring broad accessibility.

Tasks Completed

- **Set up React environment and initial HTML structure** - Sameer Rana
- **Develop regex to NFA conversion logic** - Sameer Rana (with guidance)
- **Implement NFA to DFA conversion algorithm** - Sameer Rana
- **Integrate Viz.js for visual output** - Sameer Rana
- **Style the interface with CSS** - Sameer Rana
- **Commit and push code to GitHub repository** - Sameer Rana

These tasks established a functional prototype, enabling regex input and automata visualization. The project evolved from a basic setup to a working tool, with commits

tracking progress. Initial testing confirmed basic regex support, laying a solid foundation for enhancements.

Challenges/Roadblocks

- A major challenge was debugging the NFA-to-DFA conversion, where edge cases (e.g., empty strings) caused errors, resolved by refining the subset construction algorithm. Another roadblock was integrating Viz.js, which required adjusting DOT syntax for accurate graph rendering—solved through trial and error and documentation review. File naming conflicts (e.g., `nfa-dfa-converter` vs. `Regex-to-NFA-DFA-Converter`) delayed Git pushes, addressed by renaming and updating remotes. Limited React experience slowed initial setup, overcome with online tutorials. Future challenges include supporting complex regex (e.g., nested groups), planned via algorithm expansion. Authentication issues with GitHub PATs were mitigated by re-generating tokens. Ongoing efforts focus on optimizing performance for large inputs, potentially requiring backend support, which we'll explore with further research.

Tasks Pending

- **Add support for complex regex (e.g., nested groups)** - Sameer Rana
- **Implement export functionality (e.g., PNG, SVG)** - Sameer Rana
- **Enhance UI with interactive features (e.g., step-by-step conversion)** - Sameer Rana
- **Add unit tests for conversion logic** - Sameer Rana
- **Optimize performance for large regex inputs** - Sameer Rana

These tasks aim to expand functionality and reliability, with prioritization based on user feedback post-enhancement.

Project Outcome/Deliverables

- The key outcome is a functional web tool converting regex to NFA and DFA with visualizations, hosted on GitHub. Deliverables include the React-based interface (`Regex-to-NFA-DFA-Converter.html`), conversion scripts (`Regex-to-NFA-DFA-Converter.js`, `Regex-to-NFA-DFA-Converter.jsx`), and styling (`Regex-to-NFA-DFA-Converter.css`). The tool supports basic regex operations, offering educational value through visual automata graphs. Future deliverables include export options and complex regex support, enhancing usability. The project delivers a learning resource for automata theory, with potential for academic or professional use. Documentation (e.g., README) will accompany the code, ensuring accessibility.

Progress Overview

- The project is 70% complete, with core conversion and visualization functional. Basic regex support is ahead of schedule, completed within weeks. UI enhancements and complex regex support are behind schedule due to debugging and learning curves. Testing is partially done, with plans for comprehensive validation pending. Overall, the prototype is solid, and remaining tasks are feasible within a short timeframe. Progress is tracked via GitHub commits, ensuring transparency. The focus now shifts to refinement and testing to meet the 100% milestone.

Codebase Information

- Repository Link:** <https://github.com/sameer-rana-work/Regex-to-NFA-DFA-Converter>
- Branch:** `main`
- Important Commits:**
 - Initial setup: `342643a` - Added project files
 - Algorithm implementation: `[commit-hash]` - Integrated NFA/DFA conversion
 - Visualization: `[commit-hash]` - Added Viz.js integration

Testing and Validation Status

Test Type	Status (Pass/Fail)	Notes
Manual Input Testing	Pass	Verified basic regex (e.g., <code>a b</code>) conversion
Edge Case Testing	Partial Fail	Empty string fails; needs fix
Performance Testing	Pending	To be conducted post-optimization

Deliverables Progress

- React Interface:** Completed - Fully functional
- Conversion Logic:** In Progress - Supports basics, complex regex pending
- Visualization:** Completed - Viz.js integrated
- Export Functionality:** Pending - Planned for future release
- Documentation:** In Progress - README to be added