

Poor Person's GREP: Reflections and Learnings

Developing Poor Person's GREP (PPG), a simplified version of Unix grep, proved to be a rewarding and enriching project. As it demanded specific regular expression (RE) handling and parsing of text files, I was exposed to several programming nuances. Here's a student's perspective on the learning outcomes and challenges encountered during the project.

Learning Outcomes:

1. **Regular Expression Handling:** This project bolstered my understanding of regular expressions and their potential for pattern searching within text. The PPG supported only a limited number of metacharacters, which encouraged me to delve deeper into the functionality of each metacharacter.
2. **Implementation of Nondeterministic Finite Automata (NFA):** The project required translating the given RE into an NFA. This conversion process was an illuminating experience that cemented my understanding of automata theory and its practical applications.
3. **Depth-First Search (DFS) Implementation:** Employing DFS on the NFA to determine if the RE matches the input string was a crucial aspect of this project. This improved my knowledge of graph theory and searching algorithms, particularly DFS.
4. **File Handling and Parsing:** The project reinforced my skills in file handling. Reading a text file and applying the regular expression on each line was a practical exercise in efficient data handling and manipulation.

Challenges:

1. **Limited Metacharacter Support:** The PPG only supported five RE metacharacters. Figuring out how to build a fully functional grep utility with this constraint was challenging. It required a detailed understanding of these metacharacters and their interactions with each other.
2. **NFA Implementation:** Translating a regular expression into a NFA, especially in a programming context, was not straightforward. Understanding the correlation between the two and implementing it was initially difficult.

3. **Depth-First Search Execution:** Executing a DFS to evaluate the NFA was a complicated task. Finding a way to guide the search using the characters in the matched string required meticulous planning and implementation.
4. **Dealing with Docker:** Working with the Docker container to build and run the initial partial implementation was an added challenge. It required me to be comfortable with Docker commands and understand the nuances of the containerized development environment.

Overall, the project was a blend of complexity and learning. The challenges I faced while creating the PPG furthered my understanding of regular expressions, finite automata, search algorithms, and file handling. The knowledge and skills I gained from this project extend beyond the classroom and will be valuable in future programming endeavors.