**Own Issue Project Proposal**

|  |  |  |  |
| --- | --- | --- | --- |
| Name : |  | | |
| Provide an overview of what your program is intended to do. | Compile a regular expression into a deterministic finite automaton that can match a given regular expression so that regex can be used for efficient pattern matching, in O(n) time  DFAs are represented by a [graph](https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)), which represents the possible paths between states, overall representing a pattern to match. NFAs are a more general form of DFAs, but can have non-deterministic outputs from a state – i.e. one state can have multiple out paths which could both be activated by the same input char. This means that it is not always determinable which path should be traversed in an NFA. Powerset construction can be used to convert an NFA to a DFA, which solves this issue, and allows the pattern-matching to run in linear time, at the cost of exponential space (O(exp(n))). | | |
| What are the steps the program needs to complete to be effective?  /  What are the specifications that need to be met? | 1. *Parse the regular expression into an NFA* 2. *Optimise and minimize the NFA, and perform e-closure* 3. *Use powerset construction to convert the NFA into a DFA*   *The program should*   * *Produce a minimal, efficient DFA* * *Be able to evaluate the DFA on an input string* | | |
| Identify the Complex Programming Skills you intend to include and how you intend to use them. | *1. Classes* | | |
| *2.* | | |
| *3.* | | |
| *4.* | | |
| What will be the biggest challenges you will face with this project? | Powerset construction, e-closure and minimisation can be quite challenging.  Powerset construction especially can be quite mathmatical and can be hard to detect if two paths along the graph have any overlap, which means that their powerset would be reachable. | | |
| Status: | Approved Rejected | | |
| Teacher: |  | Date : |  |