Documentation

https://github.com/iliedr19/lftc4

EBNF for fa.txt:

states = "A" | "B" | "C" | "D";

alphabet = "a" | "b" | "c" | "d";

final_states = "D";

transitions =

"A" "a" "A" |

"A" "b" "B" |

"B" "c" "C" |

"C" "c" "C" |

"C" "d" "D";

The project encompasses the implementation of Finite Automata (FA) for recognizing specific patterns or sequences within defined alphabets. There are multiple FAs designed for distinct purposes, each defined in a separate file.

Files Description:

1. `fa.txt`:

- Represents an FA that recognizes sequences ending with a specific pattern.

- States: A, B, C, D

- Alphabet: a, b, c, d

- Initial State: A

- Final State: D

2. `fa_identifier.txt`:

- Defines an FA for recognizing valid identifiers (variables) in a programming language.

- States: A, B

Documentation

- Alphabet: a-z, A-Z, 0-9, _
- Initial State: A
- Final State: B

3. `fa_integer_constant.txt`:
- Represents an FA that recognizes integer constants with positive or negative signs.
- States: A, B, C, D
- Alphabet: +, -, 0-9
- Initial State: A
- Final States: B, D

4. `seminar_fa.txt`:
- Represents an FA recognizing sequences with specific patterns.
- States: A, B
- Alphabet: a, b
- Initial State: A
- Final States: A, B

Usage:

- Each FA is designed with specific states, alphabets, and transitions to recognize particular patterns or sequences within strings.
- The implemented code provides functionalities to read, interpret, and utilize the FAs to recognize sequences based on their defined rules.

Understanding FAs:

- Finite Automata have states, alphabets, initial and final states, and transitions that determine how sequences are recognized.
- They can validate sequences based on predefined rules and identify specific patterns within input strings.

Implementation:

Documentation

- The provided code implements FAs that can be used to check if sequences conform to certain patterns or recognize specific tokens, identifiers, or constants within a sequence of characters.