Documentation

link to git:

lab-work-computer-science-2024-carla-mirea/1-Mini-Language-And-Scanner at main · cs-ubbcluj-ro/lab-work-computer-science-2024-carla-mirea

---

FA

My **FA** (Finite Automaton) class represents a finite automaton with methods for loading its
components from a file, displaying its elements, checking if the automaton is deterministic, and verifying if a given sequence is accepted by the FA. This implementation supports both deterministic (DFA) and non-deterministic finite automata (NFA).

Attributes:

- ***states:*** List of states in the FA.
- ***alphabet:*** List of symbols in the alphabet.
- ***transitions:*** List of transition rules, each represented by a Transition object.
- ***initialState:*** The initial state of the FA.
- ***finalStates:*** List of accepting or final states.
- ***isDeterministic:*** Boolean flag indicating if the FA is deterministic.

Constructor:

- ***FA(String filename):*** Reads the FA components from the specified file and initializes the FA. It also calls ***checkIfDeterministic()*** to set the ***isDeterministic*** field.

Private Methods:

- ***init():*** Reads lines from the file and calls ***parseLine*** on each line to populate the FA's fields.

- ***checkIfDeterministic():*** Checks if the FA is deterministic by verifying that each state-symbol pair leads to at most one transition. If any state has multiple transitions for the same symbol, it returns ***false***.
- ***parseLine(String line):*** Parses a line from the input file and assigns values to the FA fields based on the line content.
- ***parseList(String line):*** Extracts a list of values from a line in the format ***name= {value1, value2, ...}.***
- ***parseTransitions(String line):*** Parses the list of transitions from a line and creates ***Transition*** objects.
- ***printListOfString(String listname, List<String> list)***: Prints a list in a specific format.
- ***getNextState(String currentState, String symbol):*** Finds the next state given a current state and symbol. If no valid transition is found, it returns ***null.***

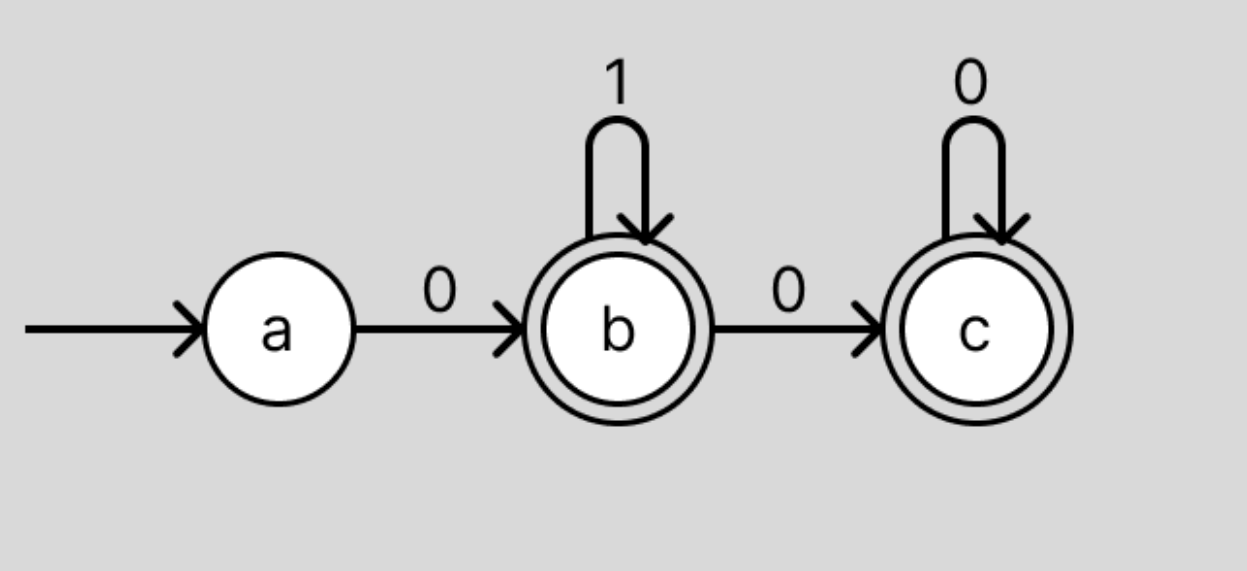
Public Methods:

- ***printStates()***: Prints the list of states in the FA.
- ***printAlphabet()***: Prints the FA's alphabet.
- ***printTransitions()***: Prints the transitions of the FA.
- ***printFinalStates()***: Prints the final states.
- ***printInitialState():*** Prints the initial state.
- ***isDeterministic():*** **Returns** the boolean ***isDeterministic*** indicating if the FA is deterministic.
- ***isAcceptedByFA(String sequence):*** For a deterministic FA, checks if a given sequence of symbols is accepted. If a symbol not in the alphabet is encountered, it displays an error message. If the FA is not deterministic, it displays a message and returns ***false***.

*I also kept the implementation of the ***Scanner*** from the last lab, together with the ***HashTable*** and ***SymbolTable*** classes, letting the user choose from a menu which operation wants to try out (run Scanner or FA).

*For the FA class I also used a ***Transition*** class for representing a transition of the form (from, to, label).

Representation of the fa.in:



(I made this using Figma)

non_zero_digit = 1|2| .. |9

digit = 0|1|..|9

number = non_zero_digit{digit}

letter = a|b|..|z|A|B..|Z

character = letter | digit

triple = "(" {character} "," {character} "," {character} ")"

firstLine = "states" "=" "{" {character} {"," character} "}"

secondLine= "alphabet" "=" "{" {character} {"," character} "}"

thirdLine= "transitions" "=" "{" triple {";" triple} "}"

fourthLine= “initial state””=”{character}

fifthLine= "final states" "=" "{" {character} {"," character} "}"

inputFile = firstLine "\n" secondLine "\n" thirdLine "\n" fourthLine "\n" fifthLine