Programming Exercise 01

Strings and DFA

Galang, Kent Michael

Masayon, Christian Ace

Poledo, Clent Japhet

University of the Philippines – Mindanao

Programming Exercise 01 **CMSC 129**

Strings and DFA **Galang, Masayon, Poledo**

Description

A program that recognizes strings based on given deterministic finite automata.

Modules

StateMachine

Main author: Poledo, Clent Japhet
Class to represent a DFA and its processes.

Attributes

alphabet : list[str]
Contains the list of alphabets, list must be of length 2
states : list[str]
Contains a list of states in the DFA, state[0] is the start state
f_states : list[str]
Contains a list of final states
transition : list[list[str]]
Contains the transitions, i.e. transition[x][y] is destination state from state[x] when alphabet[y] is inputted, second dimension must be of length 2

Methods

move(src, buf)
Does the logic for state transitions

Parameters

src : str
Source state

buf : str
Input letter

Modules *(con't)*

Methods *(con't)*

Raises

Exception
If an invalid state or input letter is passed

Returns

str
destination state

is_final(state)
Determines if a given state is final

Parameter

state : str
State to test

Returns

bool
True if state is final state, false otherwise

get_start_state()
Gets the start state of the DFA

Returns

str
The start state

Modules *(con't)*

FileParser

Main author: Poledo, Clent Japhet
A class that parses .in and .dfa files into usable elements in the program.

Methods

in_parser(src)

Parses a .in file

Parameter

src : str
A file path to the .in file

Returns

list[str]
A list of all strings from the .in file

dfa_parser(src)

Parses a .dfa file

Parameter

src : str
A file path to the .dfa file

Raises

Exception
If there are invalid inputs in the file

Returns

StateMachine
A working StateMachine object based on the .dfa file

Modules *(con't)*

StringChecker

Main author: Galang, Kent Michael
A class that contains the methods for checking for valid strings

Methods

is_valid(input, state_machine)

Checks if a string is valid

Parameters

input : str
An input string to test

state_machine : StateMachine
A state machine object for recognizing valid words

Returns

bool
True if string is valid, False otherwise

check_multiple(inputs, state_machine)

Checks multiple strings if those are valid

Parameters

input : list[str]
A list of input strings to test

state_machine : StateMachine
A state machine object for recognizing valid words

Returns

list[bool]
A list of bools per string, True if string is valid, False otherwise

Modules *(con't)*

Methods *(con't)*

save_output(output_bools, filename)

Saves the output as a properly formatted strings.out file

Parameter

output_bools : list[bool]

A list of bools from check_multiple() method

filename : str

The filename to store the outputs

Modules *(con't)*

App

Main author: Masayon, Christian Ace
A class that represents the UI of the app

Attributes

dfa : StateMachine
A reference to the currently loaded dfa
inputs : list[str]
A list of input strings
file_parser : FileParser
A file parser object used to read .in and .dfa files
string_checker : StringChecker

A string checker object used to check the validity of strings given a dfa

Methods

update_status_bar(message)
Changes the text in the status bar

Parameter

message : str
Message to write in the status bar

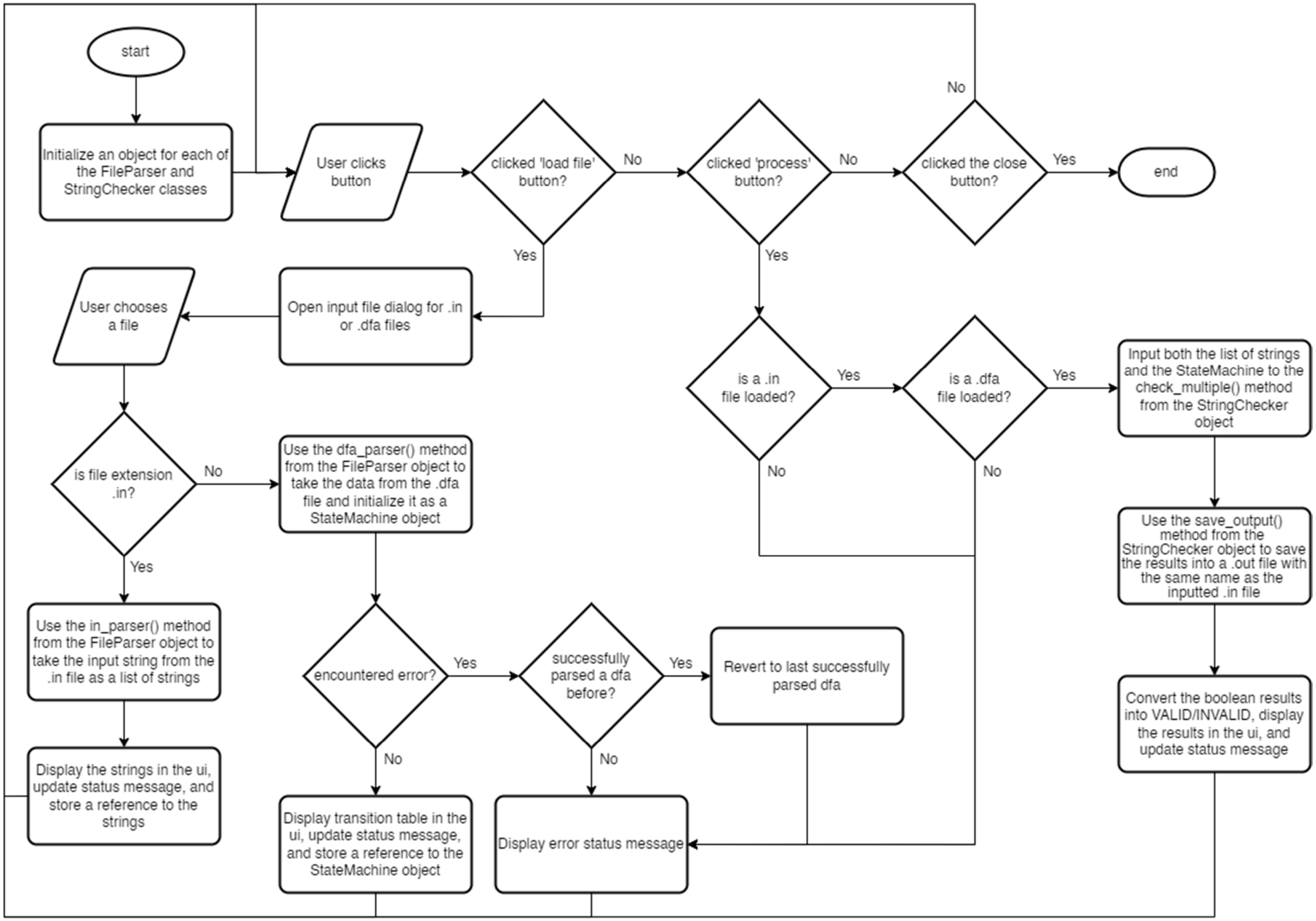
def load_file()
Handles loading of files and displaying the outputs

def process_file()
Handles checking inputs to a dfa

Programming Exercise 01 **CMSC 129**

Strings and DFA **Galang, Masayon, Poledo**

Control Flow Diagram


start
Initialize an object for each of the FileParser and StringChecker classes
User clicks button
clicked 'load file' button?
No
clicked 'process' button?
No
clicked the close button?
Yes
No
end
Yes
Yes
Open input file dialog for .in or .dfa files
User chooses a file
is file extension .in?
No
Yes
Use the dfa_parser() method from the FileParser object to take the data from the .dfa file and initialize it as a StateMachine object
Use the in_parser() method from the FileParser object to take the input string from the .in file as a list of strings
Display the strings in the ui, update status message, and store a reference to the strings
encountered error?
Yes
No
successfully parsed a dfa before?
Yes
No
Revert to last successfully parsed dfa
Display transition table in the ui, update status message, and store a reference to the StateMachine object
Display error status message
is a .in file loaded?
Yes
No
is a .dfa file loaded?
Yes
No
Input both the list of strings and the StateMachine to the check_multiple() method from the StringChecker object
Use the save_output() method from the StringChecker object to save the results into a .out file with the same name as the inputted .in file
Convert the boolean results into VALID/INVALID, display the results in the ui, and update status message