

## Overview

The DFA class implements a Deterministic Finite Automaton (DFA). A DFA is a mathematical model used to recognize patterns within input strings. It consists of a set of states, an alphabet, an initial state, a set of final states and transition functions.

The DFA class is designed to check whether a given string is accepted or rejected based on its transition rules. It supports resetting its state, processing individual characters, and verifying entire strings.

## DFA Class

### Attributes:

\_\_states: A list of states in the DFA.

\_\_alphabet: A list of valid characters in the DFA's alphabet.

\_\_initial\_state: The starting state of the DFA.

\_\_final\_states: A list of accepting states in the DFA.

\_\_transitions: A dictionary defining the transition function, where keys are concatenated state-character pairs, and values are the resulting states.

\_\_state: The current state of the DFA during execution.

### Methods:

\_\_init\_\_(states=[], alphabet=[], initial\_state=None, final\_states=[], transitions={})

Initializes a new DFA instance.

### Parameters:

states (list): The set of states in the DFA.

alphabet (list): The alphabet of valid input characters.

initial\_state (str): The state where the DFA begins execution.

final\_states (list): The set of accepting states.

transitions (dict): Transition function mapping state-character pairs to resulting states.

reset\_state()

Resets the DFA's current state to the initial state.

transition(char)

Processes a single input character and updates the DFA's current state based on its transition function.

### Parameters:

char (str): The input character to process. If the character is a digit, it is normalized to 'd'.

### Behavior:

If the transition for the current state and character exists, the DFA moves to the corresponding state.

Otherwise, the DFA transitions to a reject state ("REJECT").

is\_accepting()

Checks if the DFA is in an accepting state.

### Returns:

True if the current state is in the set of final states, otherwise False.

check\_string(string)

Evaluates whether a given string is accepted by the DFA.

### Parameters:

string (str): The input string to process.

### Returns:

True if the string is accepted (i.e., the DFA ends in a final state after processing the string).

False if the string is rejected.

Behavior:

Resets the DFA's state.

Processes each character in the string using the transition method.

Returns False immediately if an invalid character or a reject state is encountered.

After processing all characters, checks whether the DFA is in an accepting state.

\_\_repr\_\_() and \_\_str\_\_()

Provide string representations of the DFA.

Returns:

A string containing:

The list of states.

The alphabet.

The initial state.

The list of final states.

The transition dictionary.

Execution Flow:

Initialization:

Define the DFA's states, alphabet, transitions, initial state, and final states.

Processing Strings:

Use check\_string() to evaluate input strings against the DFA's rules.

State Transitioning:

Characters are processed sequentially via transition(), and invalid inputs are immediately rejected.

Acceptance Check:

After processing a string, is\_accepting() determines whether the DFA ends in an accepting state.

DFA for validating constants:

states:

p,d,r

alphabet:

+, -, ., e, E, d

initial state:

p

final states:

d

transition functions:

pdd

ddd

rdd

p+r

p-r

p.r

per

pEr

d+r

d-r

d.r

der

dEr

r+r  
r-r  
r.r  
rer  
rEr