**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №3**

**по дисциплине «Информатика»**

**Тема: Машина Тьюринга**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3344 |  | Анахин Е.Д. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Изучить принцип машины Тьюринга и научиться имитировать ее работу на языке программирования Python.

## Задание.

Вариант 1

На вход программе подается строка неизвестной длины. Каждый элемент является значением в ячейке памяти ленты Машины Тьюринга.

Напишите программу, которая удаляет в исходной строке два символа, следующих за первым встретившимся символом ‘b’. Если первый встретившийся символ ‘b’ – последний в строке, то удалить его. Если первый встретившийся символ ‘b’ – предпоследний в строке, то удалить один символ, следующий за ним, т. е. последний в строке. Если в строке символ ‘b’ отсутствует, то удалить самый первый символ строки. После удаления в строке не должно оставаться пробелов и пустых мест!

Указатель на текущее состояние Машины Тьюринга изначально находится слева от строки с символами (но не на первом ее символе). По обе стороны от строки находятся пробелы.

На ленте находится последовательность латинских букв из алфавита {a, b, c}. Напишите программу, которая заменяет в исходной строке символ, идущий после последних двух встретившихся символов 'a', на предшествующий им символ(гарантируется, что это не пробел). Наличие в строке двух подряд идущих символов 'a' гарантируется.

Алфавит:

a

b

c

" " (пробел)

Соглашения:

1. Направление движения автомата может быть одно из R (направо), L (налево), N (неподвижно).

2. Гарантируется, что длинна строки не менее 5 символов и не более 13.

3. В середине строки не могут встретиться пробелы.

4. При удалении или вставке символов направление сдвигов подстрок не принципиально (т. е. результат работы алгоритма может быть сдвинут по ленте в любую ее сторону на любое число символов).

5. Курсор по окончании работы алгоритма может находиться на любом символе.

**Выполнение работы**

Был объявлены константы LEFT, RIGHT, STOP, которые обозначают шаги, на сколько нужно сместиться по ленте. Далее был создан словарь states, который включал в себя все состояния, представленные в виде таблицы. Далее было произведено считывание строки значений ленты. Далее, пока состояние не становится ‘finished’ происходит имитация работы машины Тьюринга с помощью цикла while. Данный цикл работал до того, как состояние не переходило в ‘finished’. Из словаря состояний считывались символов, смещение и состояние, в которое нужно будет перейти после выполнения.

Состояния:

q0 – начальное состояние, которое действует до тех пор, пока не будет найден первый значащий символ

q1 – состояние поиска символа b

b\_found - состояние, при котором найдем первый символ b

first\_deleted - состояние, когда удален один символ после b

b\_last - состояние, при котором b - последний символ строки

b\_not\_found - состояние, при котором не было найдено ни одного символа b

first\_word - состояние, при котором нужно удалить первый символ строки, если не было символов b

finished - состояние, которое повествует о том, что машина Тьюринга закончила свою работу

Таблица состояний представлена в табл. 1

Таблица 1 — Таблица состояний

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ‘ ’ | ‘a’ | ‘b’ | ‘c’ |
| q0 | ' ', 1, q0 | 'a', 1, q1 | 'b', 1, q1 | 'c', 1, q1 |
| q1 | ' ', -1, b\_not\_found | 'a', 1, q1 | ‘b’, 1, b\_found | ‘c’, 1, q1 |
| b\_found | ' ', -1, b\_last | ' ', 1, first\_deleted | ‘b’, 1, first\_deleted | ‘c’, 1, first\_deleted |
| first\_deleted | ‘ ‘, 0, finished | ' ‘, 0, finished | ' ‘, 0, finished | ' ‘, 0, finished |
| b\_last |  |  | ' ', 0, finished |  |
| b\_not\_found | ‘ ‘, 1, first\_word | 'a', -1, b\_not\_found |  | 'c', -1, b\_not\_found |
| first\_word |  | ‘ ’, 0, finished |  | ‘ ’, 0, finished |

## Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
|  | abcabc | abbc | - |
|  | ccbbaa | ccba | - |

## Выводы

Был получен навык составления таблиц для машины Тьюринга. Было получено знание о работе машины Тьюринга.

# **ПРИЛОЖЕНИЕ А** **ИСХОДНЫЙ КОД ПРОГРАММЫ**

Название файла: Anakhin\_Egor\_lb3.py

LEFT, STOP, RIGHT = -1, 0, 1

states = {

("q0", " "): (' ', RIGHT, "q0"),

("q0", "a"): ('a', STOP, "q1"),

("q0", "b"): ('b', RIGHT, "b\_found"),

("q0", "c"): ('c', STOP, "q1"),

("q1", " "): (' ', LEFT, "b\_not\_found"),

("q1", "a"): ('a', RIGHT, "q1"),

("q1", "b"): ('b', RIGHT, "b\_found"),

("q1", "c"): ('c', RIGHT, "q1"),

("b\_found", " "): (' ', LEFT, "b\_last"),

("b\_found", "a"): (' ', RIGHT, "first\_deleted"),

("b\_found", "b"): (' ', RIGHT, "first\_deleted"),

("b\_found", "c"): (' ', RIGHT, "first\_deleted"),

("first\_deleted", " "): (' ', STOP, "finished"),

("first\_deleted", "a"): (' ', STOP, "finished"),

("first\_deleted", "b"): (' ', STOP, "finished"),

("first\_deleted", "c"): (' ', STOP, "finished"),

("b\_last", "b"): (' ', STOP, 'finished'),

("b\_not\_found", " "): (' ', RIGHT, "first\_word"),

("b\_not\_found", "a"): ('a', LEFT, "b\_not\_found"),

("b\_not\_found", "b"): ('b', LEFT, "b\_not\_found"),

("b\_not\_found", "c"): ('c', LEFT, "b\_not\_found"),

("first\_word", "a"): (' ', STOP, 'finished'),

("first\_word", "c"): (' ', STOP, 'finished'),

}

string = list(' ' + input() + ' ')

position = 0

state = 'q0'

while state != 'finished':

symbol, move, state\_after = states[(state, string[position])]

string[position] = symbol

position += move

state = state\_after

print(''.join(string).replace(" ", ""))