**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 |  | Валиев Р.А. |
| 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**

Состояния:

table – словарь таблицы состояний машины Тьюринга. Memory – начальная строка state – переменная, хранящая текущее состояние, ind – переменная, хранящая текущий номер ячейки на лекте.

q1 – начальное состояние, которое необходимо, чтобы найти первую букву в строке; если первая буква – b, то машина переходит в состояние q3; если первая буква – a или c, то в состояние q2.

q2 – состояние, которое необходимо для обработки случая, когда буква b стоит не на первой позиции (после нахождения переходит в состояние q3) или её вообще нет в строке; в таком случае, машина пропускает все ячейки со значениями a и c, доходит до конца буквенной строки и переходит в состояние q4

q3 – состояние, которое необходимо для удаления первого символа, идущего сразу за первой найденной буквой b; если данный символ – ‘ ’, то машина переходит в состояние q5.

q4 - состояние, которое необходимо для сдвига каретки с конца буквенной строки в начало; при нахождении символа ‘ ’ сдвигается на первый символ и переходит в состояние q5.

q5 – состояние, которое необходимо для удаления второго символа, идущего после первой найденной буквы b, для удаления последнего символа в буквенной строке, либо для удаления первого символа в строке, если буквы b нет в строке; после обработки каждого символа машина переходит в состояние q0.

q0 – конечное состояние, завершение работы машины Тьюринга.

Таблица 1 — Таблица состояний

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ‘a’ | ‘b’ | ‘c’ | ‘ ’ |
| q1 | ‘a’, R, ‘q2’ | ‘b’, R, ‘q3’ | ‘c’, R, ‘q2’ | ‘\_’, R, ‘q1’ |
| q2 | ‘a’, R, ‘q2’ | ‘b’, R, ‘q3’ | ‘c’, R, ‘q2’ | ‘\_’, L, ‘q4’ |
| q3 | ‘’, R, ‘q5’ | ‘’, R, ‘q5’ | ‘’, R, ‘q5’ | ‘\_’, L, ‘q5’ |
| q4 | ‘a’, L, ‘q4’ | – | ‘c’, L, ‘q4’ | ‘\_’, R, ‘q5’ |
| q5 | ‘’, N, ‘q0’ | ‘’, N, ‘q0’ | ‘’, N, ‘q0’ | ‘’, N, ‘q0’ |

## 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

L, R, N = -1, 1, 0

table = {

'q1': {'a': ['a', R, 'q2'], 'b': ['b', R, 'q3'], 'c': ['c', R, 'q2'], ' ': ['\_', R, 'q1']},

'q2': {'a': ['a', R, 'q2'], 'b': ['b', R, 'q3'], 'c': ['c', R, 'q2'], '\_': ['\_', L, 'q4']},

'q3': {'a': ['', R, 'q5'], 'b': ['', R, 'q5'], 'c': ['', R, 'q5'], '\_': ['\_', L, 'q5']},

'q4': {'a': ['a', L, 'q4'], 'c': ['c', L, 'q4'], '\_': ['\_', R, 'q5']},

'q5': {'a': ['', N, 'q0'], 'b': ['', N, 'q0'], 'c': ['', N, 'q0'], '\_': ['', N, 'q0']},

}

memory = list(input())

state = 'q1'

ind = 0

while state != 'q0':

char = memory[ind] if ind < len(memory) else '\_'

act = table[state].get(char, ['\_', N, 'q0']) # Обработка неизвестных символов

memory[ind] = act[0]

ind += act[1]

if ind >= len(memory) and act[0] != '':

memory.append('\_')

state = act[2]

result = ''.join(memory).rstrip('\_')

print(result)