**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. 3342 |  | Иванов Д. М. |
| Преподаватель |  | Иванов Д. В. |

Санкт-Петербург

2023

## Цель работы

Изучить устройство Машины Тьюринга и реализовать ее на зыке Python для работы со строками.

## Задание

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

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.

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.

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 15.

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

Первым делом необходимо смоделировать Машину Тьюринга.

Таблица 1 – Схема Машины Тьюринга.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a | b | c | “ ” |
| q1 | a; R; q2 | b; R; q2 | c; R; q2 | “ ”; R; q1 |
| q2 | a; R; q2 | b; R; q2 | c; R; q2 | “ ”; L; q3 |
| q3 | a; L; q4 | b; L; q3 | c; L; q3 |  |
| q4 | a; L; q5 | b; L; q3 | c; L; q3 |  |
| q5 | a; R; q6 | b; R; q7 | c; R; q8 |  |
| q6 | a; R; q6 | a; N; q9 | a; N; q9 | a; N; q9 |
| q7 | a; R; q7 | b; N; q9 | b; N; q9 | b; N; q9 |
| q8 | a; R; q8 | c; N; q9 | c; N; q9 | c; N; q9 |

Состояния:

* q1 – проход от левых пробелов до первой буквы
* q2 – проход от буква строки до первого пробела справа
* q3 – нахождения первой буквы “a”
* q4 – случай, когда найдена первая a, если на q4 тоже она будет, то нужная подстрока найдена и переход к q5
* q5 – выбор буквы слева от подстроки “aa”
* q6 – замена буквы после “aa” на “a”
* q7– замена буквы после “aa” на “a”
* q8– замена буквы после “aa” на “a”
* q9 – выход из Машины

Теперь осталось это реализовать на языке Python. Схему Машины сохраним в виде словаря, будем считывать строку, проходить по ней циклом. В зависимости от символа и текущего состояния брать значения ключа.

## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
| 1. | |  |  | | --- | --- | | abcaabc |  | | abcaacc | Верный вывод |
| 2. | aabbaa | aabbaab | Верный вывод |

## Выводы

Была разработана программа на языке Python, которая проводит работу со строкой по определенному алгоритму Машины Тьюринга. Изучена Машина Тьюринга.

# Приложение А Исходный код программы

Название файла: main.py

dic = {

'q1': {

'a': ('a', 1, 'q2'),

'b': ('b', 1, 'q2'),

'c': ('c', 1, 'q2'),

' ': (' ', 1, 'q1')

},

'q2': {

'a': ('a', 1, 'q2'),

'b': ('b', 1, 'q2'),

'c': ('c', 1, 'q2'),

' ': (' ', -1, 'q3')

},

'q3': {

'a': ('a', -1, 'q4'),

'b': ('b', -1, 'q3'),

'c': ('c', -1, 'q3')

},

'q4': {

'a': ('a', -1, 'q5'),

'b': ('b', -1, 'q3'),

'c': ('c', -1, 'q3')

},

'q5': {

'a': ('a', 1, 'q6'),

'b': ('b', 1, 'q7'),

'c': ('c', 1, 'q8')

},

'q6': {

'a': ('a', 1, 'q6'),

'b': ('a', 0, 'q9'),

'c': ('a', 0, 'q9'),

' ': ('a', 0, 'q9')

},

'q7': {

'a': ('a', 1, 'q7'),

'b': ('b', 0, 'q9'),

'c': ('b', 0, 'q9'),

' ': ('b', 0, 'q9')

},

'q8': {

'a': ('a', 1, 'q8'),

'b': ('c', 0, 'q9'),

'c': ('c', 0, 'q9'),

' ': ('c', 0, 'q9')

}

}

a = list(input())

index = 0

sost = ['q1']

while True:

new\_s, step, new\_sost = dic[sost[-1]][a[index]]

a[index] = new\_s

index += step

sost.append(new\_sost)

if sost[-1] == 'q9':

break

print(''.join(a))