**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 и конечные автоматы**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3342 |  | Песчатский С. Д. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Целью работы является освоение принципов работы машины Тьюринга и реализация её работы на языке Python.

## Задание

Вариант 2

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

Программа разработана на языке Python. В переменной table хранится словарь состояний. Далее считывается строка в массив memory и в переменной pos хранится текущее положение, с помощью переменной d изменяется pos. Далее, с помощью цикла while изменяются значения memory[pos], d и state в соответствии с словарём tables. Условием завершения цик1ла является терминальное состояние ‘qt’.

Таблица 1 – состояния машины Тьюринга

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | « » | a | b | c |
| q0 | (' ',1,'q0') | ('a', 1, 'q1') | ('b', 1, q1') | ('c', 1, 'q1') |
| q1 | (' ', -1, 'q2') | ('a', 1, 'q1') | ('b', 1, 'q1') | ('c', 1, 'q1') |
| q2 | - | ('a', -1, 'q2') | ('b', -1, 'q2') | ('c', -1, 'q3') |
| q3 | - | ('a', -1, 'q4') | ('b', -1, 'q2') | ('c', -1, 'q2') |
| q4 | - | ('a', 1, 'q5') | ('b', 1, 'q6') | ('c', 1, 'q7') |
| q5 | ('a', 0, 'qt') | ('a', 1, 'q5') | ('a', 0, 'qt') | ('a', 0, 'qt') |
| q6 | ('b', 0, 'qt') | ('b', 1, 'qt') | ('b', 0, 'qt') | ('b', 0, 'qt') |
| q7 | ('c', 0, 'qt') | ('c', 1, 'qt') | ('c', 0, 'qt') | ('c', 0, 'qt') |

q0 – начальное состояние, в котором машина ищет первый не пробельный символ.

q1 – состояние, в котором машина ищет конец строки.

q2 – состояние, в котором машина ищет первый с конца символ ‘a’.

q3 – состояние, в котором машина проверяет предшествующий символ первому с конца символу ‘a’.

q4 – состояние, в котором машина проверяет предшествующий двум подряд идущим символам ‘a’ символ.

q5 – состояние, в котором машина заменяет любой символ, идущий после двух подряд идущих ‘a’ на ‘a’.

q6 – состояние, в котором машина заменяет любой символ, идущий после двух подряд идущих ‘a’ на ‘b’.

q7 – состояние, в котором машина заменяет любой символ, идущий после двух подряд идущих ‘a’ на ‘c’.

qt – терминальное состояние.

Разработанный программный код см. в Приложении А.

## 

## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |
| --- | --- | --- |
| № п/п | Входные данные | Выходные данные |
|  | |  |  | | --- | --- | | abcaabc |  | | abcaacc |
|  | aabbaa | aabbaab |

## Выводы

В ходе выполнения лабораторной работы была усвоена и реализована работа машины Тьюринга на языке Python.

# Приложение А Исходный код программы

Название файла: main.py

table={

'q0':{'a': ('a',1,'q1'), 'b': ('b',1,'q1'), 'c': ('c',1,'q1'), ' ': (' ',1,'q0')},

'q1':{'a': ('a',1,'q1'), 'b': ('b',1,'q1'), 'c': ('c',1,'q1'), ' ': (' ',-1,'q2')},

'q2':{'a': ('a',-1,'q3'),'b': ('b',-1,'q2'),'c': ('c',-1,'q2')},

'q3':{'a': ('a',-1,'q4'),'b': ('b',-1,'q2'),'c': ('c',-1,'q2')},

'q4':{'a': ('a',1,'q5'), 'b': ('b',1,'q6'), 'c': ('c',1,'q7')},

'q5':{'a': ('a',1,'q5'), 'b': ('a',0,'qt'), 'c': ('a',0,'qt'), ' ': ('a',0,'qt')},

'q6':{'a': ('a',1,'q6'), 'b': ('b',0,'qt'), 'c': ('b',0,'qt'), ' ': ('b',0,'qt')},

'q7':{'a': ('a',1,'q7'), 'b': ('c',0,'qt'), 'c': ('c',0,'qt'), ' ': ('c',0,'qt')},

}

memory = list(input())

pos = 0

state = 'q0'

while(state != 'qt'):

memory[pos], d, state = table[state][memory[pos]]

pos += d

print(''.join(memory), sep='\n')