

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

Львов А.В.

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 длина 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 STATES хранится словарь состояний. Далее считывается строка в переменную memory; переменная index хранит значение текущего места на ленте, переменная state – текущее положение. Далее, с помощью цикла while изменяются значения memory[index], index и state в соответствии с словарём STATES. Условием завершения цикла является терминальное состояние 'qt'.

Таблица 1 – состояния машины Тьюринга

	« »	a	b	c
q1	(' ', 1, 'q1')	('a', 1, 'q2')	('b', 1, 'q2')	('c', 1, 'q2')
q2	(' ', 0, 'q3')	('a', 1, 'q2')	('b', 1, 'q2')	('c', 1, 'q2')
q3	(' ', -1, 'q3')	('a', -1, 'q4')	('b', -1, 'q3')	('c', -1, 'q3')
q4	(' ', -1, 'q3')	('a', -1, 'q5')	('b', -1, 'q3')	('c', -1, 'q3')
q5	(' ', 3, 'qt')	('a', 3, 'q6')	('b', 3, 'q7')	('c', 3, 'q8')
q6	('a', 0, 'qt')	('a', 0, 'qt')	('a', 0, 'qt')	('a', 0, 'qt')
q7	('b', 0, 'qt')	('b', 0, 'qt')	('b', 0, 'qt')	('b', 0, 'qt')
q8	('c', 0, 'qt')	('c', 0, 'qt')	('c', 0, 'qt')	('c', 0, 'qt')
qt	(' ', 0, 'qt')	(' ', 0, 'qt')	(' ', 0, 'qt')	(' ', 0, 'qt')

q1 – начальное состояние, в котором машина ищет первый непробельный СИМВОЛ.

q2 – состояние, в котором машина ищет конец строки.

q3 – состояние, в котором машина ищет первый с конца символ 'а'.

q4 – состояние, в котором машина проверяет предшествующий символ первому с конца символу 'а'.

q5 – состояние, в котором машина проверяет предшествующий двум подряд идущим символам 'а' символ.

q6 – состояние, в котором машина заменяет любой символ, идущий после двух подряд идущих 'а' на 'а'.

q7 – состояние, в котором машина заменяет любой символ, идущий после двух подряд идущих 'а' на 'b'.

q8 – состояние, в котором машина заменяет любой символ, идущий после двух подряд идущих 'а' на 'с'.

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
1.	abcaabc	abcaacc
2.	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

```
STATES = {'q1': {' ': (' ', 1, 'q1'),
                'a': ('a', 1, 'q2'),
                'b': ('b', 1, 'q2'),
                'c': ('c', 1, 'q2')},
          'q2': {' ': (' ', 0, 'q3'),
                'a': ('a', 1, 'q2'),
                'b': ('b', 1, 'q2'),
                'c': ('c', 1, 'q2')},
          'q3': {' ': (' ', -1, 'q3'),
                'a': ('a', -1, 'q4'),
                'b': ('b', -1, 'q3'),
                'c': ('c', -1, 'q3')},
          'q4': {' ': (' ', -1, 'q3'),
                'a': ('a', -1, 'q5'),
                'b': ('b', -1, 'q3'),
                'c': ('c', -1, 'q3')},
          'q5': {' ': (' ', 3, 'qt'),
                'a': ('a', 3, 'q6'),
                'b': ('b', 3, 'q7'),
                'c': ('c', 3, 'q8')},
          'q6': {' ': ('a', 0, 'qt'),
                'a': ('a', 0, 'qt'),
                'b': ('a', 0, 'qt'),
                'c': ('a', 0, 'qt')},
          'q7': {' ': ('b', 0, 'qt'),
                'a': ('b', 0, 'qt'),
                'b': ('b', 0, 'qt'),
                'c': ('b', 0, 'qt')},
          'q8': {' ': ('c', 0, 'qt'),
                'a': ('c', 0, 'qt'),
                'b': ('c', 0, 'qt'),
                'c': ('c', 0, 'qt')},
          'qt': {' ': (' ', 0, 'qt'),
                'a': (' ', 0, 'qt'),
                'b': (' ', 0, 'qt'),
                'c': (' ', 0, 'qt')},
          }

state = 'q1'
index = 0
memory = list(' ') + list(input()) + list(' ')
while state != 'qt':
    char = memory[index]
    memory[index] = STATES[state][char][0]
    index += STATES[state][char][1]
    state = STATES[state][char][2]
print(''.join(memory))
```