**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Информатика»**

Тема: Машина Тьюринга

|  |  |  |
| --- | --- | --- |
| Студент гр. 3341 |  | Мальцев К.Л. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Написать программу, которая заменяет в исходной строке символ, идущий после последних двух встретившихся символов 'a', на предшествующий им символ, используя машину Тьюринга.

## Задание

Вариант 2

python:

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.

Для примера выше лента будет выглядеть так:

acaabcbabaabab

Алфавит: abc" " (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.

Ваша программа должна вывести полученную ленту после завершения работы.

## Основные теоретические положения

1. Машина Тьюринга состоит из неподвижной бесконечной ленты и автомата.

2. Лента используется для хранения информации и разбита на клетки.

3. Алфавит ленты - конечное множество символов, которые могут быть записаны на ленте.

4. Автомат является активной частью Машины Тьюринга и размещается под одной из клеток ленты.

5. Автомат видит содержимое только видимой клетки и находится в одном из конечного числа состояний.

6. В каждом состоянии автомат выполняет определенную операцию и может совершить определенные действия.

7. Программа для Машины Тьюринга может быть записана в виде таблицы, где указываются символы, направления и состояния.

## Выполнение работы

1. q1 - Начальное состояние, чтобы найти первый встретившийся символ 'a'. Переход на состояние q2 при обнаружении символа 'a', 'b', 'c', либо сразу смещение вправо, если встречен пробел.

2. q2 - Поиск предыдущих двух символов 'a'. В случае обнаружения символа 'a', переход на состояние q3 после смещения коретки на один символ вправо. Если обнаружен пробел, смещение влево на один символ для перехода на состояние q3.

3. q3 - Поиск последнего символа 'a' перед последними двумя символами 'a'. После нахождения этого символа, смещение влево на один символ и переход к состоянию q4.

4. q4 - Проверка символа после последних двух символов 'a'. Если этот символ 'a', переход к состоянию q5, если нет, смещение влево на один символ и возврат к состоянию q3.

5. q5 - "Запоминание" символа перед последними двумя символами 'a' и переход к соответствующему выполняющему операцию состоянию q6, q7 или q8.

6. q6 - Заменяет следующий символ после последних двух символов 'a' на 'a'.

7. q7 - Заменяет следующий символ после последних двух символов 'a' на 'b'.

8. q8 - Заменяет следующий символ после последних двух символов 'a' на 'c'.

Код осуществляет обработку каждого состояния и переходов между ними, в результате изменяя входную ленту согласно условиям задачи.

По шагам работы кода:

1. Инициализация начального состояния state = 'q1', установка коретки в начале ленты head=0.

2. Вход в цикл while state != 'q9'.

3. Извлечение символа и операции из таблицы переходов d[state][tape[head]].

4. Обновление символа на ленте tape[head] в соответствии с таблицей переходов.

5. Сдвиг коретки на нужное количество шагов в соответствии с операцией из таблицы переходов.

6. Изменение текущего состояния state на новое состояние, указанное в таблице переходов.

7. Повторение шагов 3-6 до достижения конечного состояния q9.

8. Вывод измененной ленты после завершения работы Машины Тьюринга.

## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

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

## Выводы

В результате работы было продемонстрировано применение Машины Тьюринга для решения задачи замены символа в строке.

# Приложение А Исходный код программы

Название файла: main.py

moves = {'R': 1, 'L': -1, 'N': 0}

d = {

'q1': {'a': 'a;R;q2', 'b': 'b;R;q2', 'c': 'c;R;q2', ' ': ' ;R;q1'}, # вход в область буквенных символов

'q2': {'a': 'a;R;q2', 'b': 'b;R;q2', 'c': 'c;R;q2', ' ': ' ;L;q3'}, # смещение коретки на пробел

'q3': {'a': 'a;L;q4', 'b': 'b;L;q3', 'c': 'c;L;q3'}, # поиск крайнего левого символа 'a' (переход к состоянию q4)

'q4': {'a': 'a;L;q5', 'b': 'b;L;q3', 'c': 'c;L;q3'}, # проверка символа перед крайним левым символом 'a' (если 'a', то перейти к q5, если нет - q3)

'q5': {'a': 'a;RRR;q6', 'b': 'b;RRR;q7', 'c': 'c;RRR;q8'}, # "запоминание" символа перед крайним 'aa' - переход к соответствующему состоянию q6, q7 или q8

'q6': {'a': 'a;N;q9', 'b': 'a;N;q9', 'c': 'a;N;q9', ' ': 'a;N;q9'}, # замена на a

'q7': {'a': 'b;N;q9', 'b': 'b;N;q9', 'c': 'b;N;q9', ' ': 'b;N;q9'}, # замена на b

'q8': {'a': 'c;N;q9', 'b': 'c;N;q9', 'c': 'c;N;q9', ' ': 'c;N;q9'}, # замена на c

}

def turing\_machine(tape):

state = 'q1'

head = 0

while state != 'q9':

symbol, directions, new\_state = d[state][tape[head]].split(';')

tape[head] = symbol

for i in directions:

head += moves[i]

state = new\_state

def main():

tape = [' '] + list(input()) + [' ']

turing\_machine(tape)

print(''.join(tape[1:-1]))

if \_\_name\_\_ == '\_\_main\_\_':

main()