**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. 3343 |  | Лихацкий В. Р. |
| Преподаватель |  | Иванов Д. В. |

Санкт-Петербург

2023

## Цель работы

Понять принцип работы машины Тьюринга и конечных автоматов и научиться использовать их для решения практических задач

## 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.

Ваша программа должна вывести полученную ленту после завершения работы.

В отчете предоставьте таблицу состояний. Отдельно кратко опишите каждое состояние, например:

q1 - начальное состояние, которое необходимо, чтобы найти первый встретившийся символ ‘a’.

## Выполнение работы

Для решения поставленной задачи моделируем работу реальной машины Тьюринга. Для этого создаем таблицы переходов состояний si (state indexes) и значений vi (value indexes), реализованные с использованием словарей. Направление головки задается словарем d (directions), в котором ключи это названия направлений, а значения — смещение головки вправо. Создаем таблицу состояний и пока наш автомат не достигнет конечного значения в зависимости от значения в текущей ячейке определяем смещение, новое значение и новое состояние. Алгоритм решения задачи следующий: перемещаемся вправо до конца строки в состоянии q1, а затем q2 и начинаем двигаться влево в состоянии q3. Если текущая клетка равна букве a, переходим в состояние q4 и проверяем, является ли следующая ячейка буквой a. Если да, то переходим в состояние q5, и переходим на ячейку перед двойной a. В зависимости от значения ячейки (a, b или c) переходим в соответствующее ей состояние q6, q7 или q8. В каждом из этих состояний буква а игнорируется, а любая другая заменяется на нужную, после чего автомат переходит в конечное состояние.

## Выводы

Было изучено строение машины Тьюринга и других конечных автоматов, получено представление об архитектуре компьютера на более низком уровне.

# Приложение А Исходный код программы

Название файла: main.py

isi = {

'q1': 0,

'q2': 1,

'q3': 2,

'q4': 3,

'q5': 4,

'q6': 5,

'q7': 6,

'q8': 7,

}

vi = {

'a': 0,

'b': 1,

'c': 2,

' ': 3

}

d = {

'R': 1,

'L': -1,

'N': 0

}

table = """aR2;bR2;cR2; R1

aR2;bR2;cR2; L3

aL4;bL3;cL3

aL5;bL3;cL3

aR6;bR7;cR8

aR6;aR9;aR9;aR9

aR7;bR9;bR9;bR9

aR8;cR9;cR9;cR9"""

table = [[[j[0], j[1], 'q'+j[2]] for j in i.split(";")]

for i in table.split("\n")]

cmd = list(input())

i = 0

h = ['q1']

while h[-1] != 'q9':

v = cmd[i]

s = h[-1]

c = table[si[s]][vi[v]]

cmd[i] = c[0]

i += d[c[1]]

h.append(c[2])

print(''.join(cmd))

# **Приложение Б Таблица состояний**

Таблица 1 – Таблица состояний

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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, R, q9 | a, R, q9 | a, R, q9 |
| q7 | a, R, q7 | b, R, q9 | b, R, q9 | b, R, q9 |
| q8 | a, R, q8 | c, R, q9 | c, R, q9 | c, R, q9 |