**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Информатика»**

Тема: «**Машина** **Тьюринга**»

|  |  |  |  |
| --- | --- | --- | --- |
| Студент гр. 3342 |  | Белаид Фарук | |
| Преподаватель |  | | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Изучить устройство и принцип работы машины Тьюринга, строение таблицы состояний и её использование для программирования машины. Научиться реализовывать машину на языке Python. На основе полученных сведений и навыков решить алгоритмическую задачу работы со строкой.

## 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,

предшествующий первому встретившемуся символу ‘c' на символ, следующий за первым встретившимся символом ‘a'. Если первый встретившийся символ ‘a' в конце строки, то используйте его в качестве заменяющего.

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: Направление движения автомата может быть одно из R (направо), L (налево), N (неподвижно). Гарантируется, что длинна строки не менее 5 символов и не более 15. В середине строки не могут встретиться пробе

## Выполнение работы

Для выполнения задачи реализована функция ***check\_crossroad(),*** Таблица состояний реализованной машины Тьюринга в табл. 1.

q0 'a', R, q2 'b', R, q1 'c', R, q1 ' ', R, q0 Описание состояний: q0 поиск начала строки, начальное состояние q1 – поиск первого вхождения символа 'a' q2 – “запоминание” символа после первого 'a' q3 – поиск начала строки для вставки 'a' q4 – поиск начала строки для вставки 'b' q5 – поиск начала строки для вставки 'c' q6 – поиск первого вхождения 'c' для вставки 'a' q7 – поиск первого вхождения 'c' для вставки 'b' q8 – поиск первого вхождения 'c' для вставки 'c' q9 – вставка 'a' перед первым встретившимся 'c' q10 – вставка 'b' перед первым встретившимся 'c' q11 – вставка 'c' перед первым встретившимся 'c' q12 – конечное состояние В программе использованы следующие переменные table – словарь, таблица состояний L, N, R – направление движения каретки и соответственно равны -1, 0, 1

## Тестирование

Результаты тестирования представлены в табл. 1.

Abcabc abcabc Ответ верный

## Выводы

Были изучены устройство и принцип работы машины Тьюринга, а также возможности её реализации в Python.

Разработана программа, выполняющая считывание ленты с клавиатуры. Для хранения состояний машины использовался словарь, а для “автоматизации” цикл *while*, в теле которого изменялось состояние, положение каретки на ленте и значение ячейки в соответствии с таблицей состояний. Во избежание возникновения исключительных ситуаций при отсутствии искомых символов было добавлено состояние для поиска начала слова и обработка пробела для состояний, ищущих первое вхождение символа.

# Приложение А Исходный код программы

Название файла: main.py

L, N, R = -1, 0, 1

table = {

'q0': {'a': ['a', R, 'q2'], 'b': ['b', R, 'q1'], 'c': ['c', R, 'q1'], ' ': [' ', R, 'q0']},'q1': {'a': ['a', R, 'q2'], 'b': ['b', R, 'q1'],'c':['c', R, 'q1'], ' ': [' ', N, 'q12']},'q2': {'a': ['a', L, 'q3'], 'b': ['b', L, 'q4'], 'c': ['c', L, 'q5'], ' ': [' ', L, 'q3']},

'q3': {'a': ['a', L, 'q3'], 'b': ['b', L, 'q3'], 'c': ['c', L, 'q3'], ' ': [' ', R, 'q6']},'q4': {'a': ['a', L, 'q4'], 'b': ['b', L, 'q4'], 'c': ['c', L, 'q4'], ' ': [' ', R, 'q7']},'q5': {'a': ['a', L, 'q5'], 'b': ['b', L, 'q5'], 'c': ['c', L, 'q5'], ' ': [' ', R, 'q8']},'q6': {'a': ['a', R, 'q6'], 'b': ['b', R, 'q6'], 'c': ['c', L, 'q9'], ' ': [' ', N, 'q12']},'q7': {'a': ['a', R, 'q7'], 'b': ['b', R, 'q7'], 'c': ['c', L, 'q10'], ' ': [' ', N, 'q12']},'q8': {'a': ['a', R, 'q8'], 'b': ['b', R, 'q8'], 'c': ['c', L, 'q11'], ' ': [' ', N, 'q12']},

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 'q9':  'q12'],' | {'a': ['a', N, 'q12'],  ': ['a', N, 'q12']}, | 'b': | ['a', | N, | 'q12'], | 'c': | ['a', | N, |
| 'q10': {'a': ['b', N, 'q12'],  'q12'], ' ': ['b', N, 'q12']},'q11': {'a':  ['c',N,'q12'], | | 'b':  'b': | ['b',  ['c', | N,  N, | 'q12'],  'q12'], | 'c':  'c': | ['b',  ['c', | N,  N, |

'q12'], ' ': ['c', N, 'q12']}}line = list(input()) state = 'q0' position = 0while state != 'q12':line[position], shift, state = table[state][line[position]] position += shift print(''.join(line))