**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

|  |  |  |
| --- | --- | --- |
| Студентка гр. 3341 |  | Кузнецова С.Е. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Целью работы является изучение работы конечных автоматов, в частности — машины Тьюринга. Требуется написать программу на языке Python, которая моделирует работу машины Тьюринга и решает определённую задачу.

## Задание

Вариант 4

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}, которая начинается с символа 'a'.

Напишите программу, которая оборачивает исходную строку. Результат работы алгоритма - исходная последовательность символов в обратном порядке.

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 (можно расширять при необходимости):

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 13.

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.

6. Нельзя использовать дополнительную ленту, в которую записывается результат.

Программа должна вывести полученную ленту после завершения работы.

## Выполнение работы

Объявлены переменные R=+1, L=-1, N=0 – направления движения автомата, переменные CHANGE\_TO=0, MOVE\_TO=1, NEXT\_STATE=2, определяющие действия и состояния автомата.

Объявлена функция def turing\_machine, которая принимает на вход программу, ленту, начальное состояние и начальную ячейку. Далее функция обрабатывает ленту: получает ряды таблицы текущего состояния и ряд состояния, заменяет символ на символ текущего ряда таблицы, передвигает указатель автомата на следующее или предыдущее значение и переходит в следующее состояние. Функция возвращает новое значение ленты в виде строки.

Программа работает так: находит первую букву, запоминает ее, заменяет ее нулем, передвигается в начало строки и вставляет данную букву перед первой буквой строки (только заглавную). То же самое проделывается со всеми остальными буквами – каждая следующая буква записывается перед следующей заглавной. Таким образом, получается лента, состоящая из строки заглавных букв (перевернутой первоначальной) и нулей. Далее все нули заменяются пробелами, а заглавные буквы – строчными.

Программа выводит результат выполнения функции машины Тьюринга от написанной программы состояний, ленты, начального состояния и начального индекса. Результат – строка, в которой мы удаляем все пробелы и выводим на экран.

Таблица состояний:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ‘a’ | ‘b’ | ‘c’ | ‘A’ | ‘B’ | ‘C’ | ‘0’ | ‘ ‘ |
| q\_start | ‘a’; L; ’q\_found’ | ‘b’; L; ‘q\_found’ | ‘c’; L; ‘q\_found’ |  |  |  |  | ‘ ‘; R; ‘q\_start’ |
| q\_found | ‘a’; L; ‘q\_found’ | ‘b’; L; ‘q\_found’ | ‘c’; L; ‘q\_found’ |  |  |  |  | ‘ ‘; R; ‘q\_back’ |
| q\_back | ‘0’; L; ‘q\_a’ | ‘0’; L; ‘q\_b’ | ‘0’; L; ‘q\_c’ | ‘A’; R; ‘q\_back’ | ‘B’; R; ‘q\_back’ | ‘C’; R; ‘q\_back’ | ‘0’; R; ‘q\_back’ | ‘ ‘; L; ‘q\_end’ |
| q\_a |  |  |  | ‘A’; L; ‘q\_a’ | ‘B’; L; ‘q\_a’ | ‘C’; L; ‘q\_a’ | ‘0’; L; ‘q\_a’ | ‘A’; R; ‘q\_back’ |
| q\_b |  |  |  | ‘A’; L; ‘q\_b’ | ‘B’; L; ‘q\_b’ | ‘C’; L; ‘q\_b’ | ‘0’; L; ‘q\_b’ | ‘B’; R; ‘q\_back’ |
| q\_c |  |  |  | ‘A’; L; ‘q\_c’ | ‘B’; L; ‘q\_c’ | ‘C’; L; ‘q\_c’ | ‘0’; L; ‘q\_a’ | ‘C’; R; ‘q\_back’ |
| q\_end |  |  |  | ‘a’; L; ‘q\_end’ | ‘b’; L; ‘q\_end’ | ‘c’; L; ‘q\_end’ | ‘ ‘; L; ‘q\_end’ | ‘ ‘, N, ‘q\_stop’ |
| q\_stop |  |  |  |  |  |  |  |  |

q\_start — начальное состояние. Машина двигается по ленте вправо, пока не найдет первый символ, отличный от пробела (a, b, c). После чего переходит в состояние q\_found.

q\_found — машина нашла первый символ строки. При поиске пробела переходит в состояние q\_back.

q\_back — машина двигается по строке в обратном направлении, пока не встретит пробел. (Т.е. пока не дойдет до конца строки). Если в данном состоянии машина встречает символ ‘a’, ‘b’ или ‘c’, она заменяет его на 0 и переходит в состояние q\_a, если символ был а, состояние q\_b — если b, и q\_c — если с. Если символ был A, B или С он не меняется, машина продолжает двигаться вправо. Если машина в данном состоянии встречает пробел, машина переходит в состояние q\_end.

q\_a — машина двигается влево, пока не находит первый пробел. После чего она заменяет его на А и переходит в состояние q\_back.

q\_b — машина двигается влево, пока не находит первый пробел. После чего она заменяет его на B и переходит в состояние q\_back.

q\_c — машина двигается влево, пока не находит первый пробел. После чего она заменяет его на C и переходит в состояние q\_back.

q\_end — машина двигается влево, пока не найдет пробел. Она заменяет 0 на пробелы, а прописные буквы на строчные. После чего машина переходит в состояние q\_stop.

См. приложение А.

## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |
| --- | --- | --- |
| № п/п | Входные данные | Выходные данные |
| 1 | abcabc | cbacba |
| 2 | abacbbc | cbbcaba |
| 3 | abccbaabb | bbaabccba |

## Выводы

Изучена работа конечных автоматов, в частности — машины Тьюринга. Написана программа на языке Python, которая моделирует работу машины Тьюринга и решает определённую задачу.

# ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main\_lb3.py

R = +1

L = -1

N = 0

CHANGE\_TO = 0

MOVE\_TO = 1

NEXT\_STATE = 2

def turing\_machine(program, tape, start\_state, current\_cell):

current\_state = start\_state

while True:

rows = program[current\_state]

current\_row = rows[tape[current\_cell]]

tape[current\_cell] = current\_row[CHANGE\_TO]

if not current\_row[MOVE\_TO]:

break

current\_cell += current\_row[MOVE\_TO]

current\_state = current\_row[NEXT\_STATE]

return ''.join(tape)

program = {'q\_start': {'a': ('a', L, 'q\_found'), 'b': ('b', L, 'q\_found'), 'c': ('c', L, 'q\_found'), ' ': (' ', R, 'q\_start')},

'q\_found': {'a': ('a', L, 'q\_found'), 'b': ('b', L, 'q\_found'), 'c': ('c', L, 'q\_found'), ' ': (' ', R, 'q\_back')},

'q\_back': {'a': ('0', L, 'q\_a'), 'b': ('0', L, 'q\_b'), 'c': ('0', L, 'q\_c'), 'A': ('A', R, 'q\_back'), 'B': ('B', R, 'q\_back'), 'C': ('C', R, 'q\_back'), '0': ('0', R, 'q\_back'), ' ': (' ', L, 'q\_end'),},

'q\_a': {'A': ('A', L, 'q\_a'), 'B': ('B', L, 'q\_a'), 'C': ('C', L, 'q\_a'), '0': ('0', L, 'q\_a'), ' ': ('A', R, 'q\_back')},

'q\_b': {'A': ('A', L, 'q\_b'), 'B': ('B', L, 'q\_b'), 'C': ('C', L, 'q\_b'), '0': ('0', L, 'q\_b'), ' ': ('B', R, 'q\_back')},

'q\_c': {'A': ('A', L, 'q\_c'), 'B': ('B', L, 'q\_c'), 'C': ('C', L, 'q\_c'), '0': ('0', L, 'q\_c'), ' ': ('C', R, 'q\_back')},

'q\_end': {'0': (' ', L, 'q\_end'), 'A': ('a', L, 'q\_end'), 'B': ('b', L, 'q\_end'), 'C': ('c', L, 'q\_end'), ' ': (' ', N, 'q\_stop')}}

tape = list(15\*' '+str(input())+' ')

index = 0

state = 'q\_start'

print(turing\_machine(program, tape, state, index).replace(' ',''))