

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_star t	'a'; L; 'q_foun d'	'b'; L; 'q_foun d'	'c'; L; 'q_foun d'					' '; R; 'q_star t'
q_fou nd	'a'; L; 'q_foun d'	'b'; L; 'q_foun d'	'c'; L; 'q_foun d'					' '; R; 'q_bac k'

q_bac k	'0'; L; 'q_a'	'0'; L; 'q_b'	'0'; L; 'q_c'	'A'; R; 'q_bac k'	'B'; R; 'q_bac k'	'C'; R; 'q_bac k'	'0'; R; 'q_bac k'	' '; L; 'q_end '
q_a				'A'; L; 'q_a'	'B'; L; 'q_a'	'C'; L; 'q_a'	'0'; L; 'q_a'	'A'; R; 'q_bac k'
q_b				'A'; L; 'q_b'	'B'; L; 'q_b'	'C'; L; 'q_b'	'0'; L; 'q_b'	'B'; R; 'q_bac k'
q_c				'A'; L; 'q_c'	'B'; L; 'q_c'	'C'; L; 'q_c'	'0'; L; 'q_a'	'C'; R; 'q_bac k'
q_end				'a'; L; 'q_end '	'b'; L; 'q_end '	'c'; L; 'q_end '	' '; L; 'q_end '	' ', N; 'q_sto p'
q_sto p								

q_start — начальное состояние. Машина движется 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 движется 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 a, состояние q_b — если b, и q_c — если c. Если символ был A, B или C он не меняется, машина продолжает двигаться вправо. Если машина в данном состоянии встречает пробел, машина переходит в состояние q_end.

q_a — машина движется влево, пока не находит первый пробел. После чего она заменяет его на А и переходит в состояние q_{back} .

q_b — машина движется влево, пока не находит первый пробел. После чего она заменяет его на В и переходит в состояние q_{back} .

q_c — машина движется влево, пока не находит первый пробел. После чего она заменяет его на С и переходит в состояние q_{back} .

q_{end} — машина движется 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(' ', ''))
```