

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

Студент гр. 3341

Романов А.К.

Преподаватель

Иванов Д. В.

Санкт-Петербург

2023

Цель работы

Целью лабораторной работы является изучение работы конечных автоматов, в частности машины Тьюринга

Для достижения поставленной цели требуется решить следующие задачи:

- 1) Ознакомиться с концепцией машины Тьюринга
- 2) Создать программу, моделирующую работу машины Тьюринга, выполняющую определенную задачу.

Задание

Вариант работы №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'.

			a	c	a	a	b	c	b	a	b	a	a	c	a	b			
--	--	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--

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:

			a	c	a	a	b	c	b	a	b	a	a	b	a	b			
--	--	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--

Алфавит:

- ⓐ a
- ⓑ b
- ⓒ c
- ⓓ " " (пробел)

Соглашения:

1. Направление движения автомата может быть одно из R (направо), L (налево), N (неподвижно).
2. Гарантируется, что длина 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.

Ваша программа должна вывести полученную ленту после завершения работы.

Выполнение работы

Для решения задачи таблица состояний машины Тьюринга (см. ниже) была реализована в виде словаря *table*.

Далее осуществляется ввод строки с клавиатуры в переменную *tape* (лента машины). Затем в массив *tape* добавляется несколько (13+1) пустых ячеек. Дополнительно добавляется одна пустая ячейка в начало массива (перед началом слова согласно заданию).

Переменная *index* используется для обозначения текущей ячейки, обрабатываемой машиной Тьюринга. (Изначально равна 0). Переменная *state* содержит текущее состояние машины Тьюринга. Изначально *q_start*.

Далее в цикле *while* перебираются символы ленты (массива *tape*). Текущий символ записывается в переменную *symb*. Далее обновляются переменные *new_symb* (на что меняется текущий символ), *movement* (сдвиг ленты машины Тьюринга: 1 — вправо, -1 — влево), *state* (новое состояние машины Тьюринга).

Функция возвращает строку, полученную из списка *tape* методом *join()*.

Массиву *table* соответствует таблица состояний машины Тьюринга:

	'a'	'b'	'c'	'A'	'B'	'C'	'0'	' '
q_start	a; 1; q_f	b; 1; q_f	c; 1; q_f					' '; 1; q_start
q_f	a; 1; q_f	b; 1; q_f	c; 1; q_f					' '; -1; q_rv
q_rv	0; 1; q_a	0; 1; q_b	0; 1; q_c	A; -1; q_rv	B; -1; q_rv	C; -1; q_rv	0; -1; q_rv	' '; 1; q_start
q_a	a; 1; q_a	b; 1; q_a	c; 1; q_a	A; 1; q_a	B; 1; q_a	C; 1; q_a	0; 0; q_a	A; -1; q_rv
q_b	a; 1; q_a	b; 1; q_a	c; 1; q_a	A; 1; q_a	B; 1; q_a	C; 1; q_a	0; 0; q_a	B; -1; q_rv

q_c	a; 1; q_a	b; 1; q_a	c; 1; q_a	A; 1; q_a	B; 1; q_a	C; 1; q_a	0; 0; q_a	C; -1; q_rv
q_end				a; 1; q_end	b; 1; q_end	c; 1; q_end	' '; 1; q_end	''; 0; qT
qT								

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_f.

q_f — машина нашла первый символ строки, после чего двигается вправо до конца строки, пока снова не найдет пробел. Потом переходит в состояние q_rv.

q_rv — машина двигается по строке в обратном направлении (влево), пока не встретит пробел. (Т.е. пока не дойдет до начала строки). Если в данном состоянии машина встречает символ отличный от 0, 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 влево. Если машина в данном состоянии встречает пробел, машина переходит в состояние q_end.

q_a — машина двигается вправо, пока не находит первый пробел. После чего она заменяет его на A и переходит в состояние q_rv.

q_b — машина двигается вправо, пока не находит первый пробел. После чего она заменяет его на B и переходит в состояние q_rv.

q_c — машина двигается вправо, пока не находит первый пробел. После чего она заменяет его на C и переходит в состояние q_rv.

q_end — машина двигается вправо, пока не найдет пробел. Она заменяет 0 на пробелы, а строчные буквы на прописные. После чего машина переходит в терминальное состояние.

Разработанный программный код см. в приложении А.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные
1.	abca	acba
2.	abababaca	acabababa
1.	aaaaca	асаааа

Выводы

Была разработана программа на языке программирования Python, симулирующая работу машины Тьюринга. Была описана программа машины Тьюринга, с помощью которой автомат способен обработать строку символов определённым образом.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

Программный код:

```
table = {'q_start': {'a': ('a', 1, 'q_f'),
                     'b': ('b', 1, 'q_f'),
                     'c': ('c', 1, 'q_f'),
                     ' ': (' ', 1, 'q_start')
          },
         'q_f': {'a': ('a', 1, 'q_f'),
                 'b': ('b', 1, 'q_f'),
                 'c': ('c', 1, 'q_f'),
                 ' ': (' ', -1, 'q_rv')
          },
         'q_rv': {'a': ('0', 1, 'q_a'),
                  'b': ('0', 1, 'q_b'),
                  'c': ('0', 1, 'q_c'),
                  'A': ('A', -1, 'q_rv'),
                  'B': ('B', -1, 'q_rv'),
                  'C': ('C', -1, 'q_rv'),
                  '0': ('0', -1, 'q_rv'),
                  ' ': (' ', 1, 'q_end'),
          },
         'q_a': {'a': ('a', 1, 'q_a'),
                 'b': ('b', 1, 'q_a'),
                 'c': ('c', 1, 'q_a'),
                 'A': ('A', 1, 'q_a'),
                 'B': ('B', 1, 'q_a'),
                 'C': ('C', 1, 'q_a'),
                 '0': ('0', 1, 'q_a'),
                 ' ': ('A', -1, 'q_rv')
          },
         'q_b': {'a': ('a', 1, 'q_b'),
                 'b': ('b', 1, 'q_b'),
                 'c': ('c', 1, 'q_b'),
                 'A': ('A', 1, 'q_b'),
                 'B': ('B', 1, 'q_b'),
                 'C': ('C', 1, 'q_b'),
                 '0': ('0', 1, 'q_b'),
                 ' ': ('B', -1, 'q_rv')
          },
         'q_c': {'a': ('a', 1, 'q_c'),
                 'b': ('b', 1, 'q_c'),
                 'c': ('c', 1, 'q_c'),
                 'A': ('A', 1, 'q_c'),
                 'B': ('B', 1, 'q_c'),
                 'C': ('C', 1, 'q_c'),
                 '0': ('0', 1, 'q_c'),
                 ' ': ('C', -1, 'q_rv')
          },
         'q_end': {'0': (' ', 1, 'q_end'),
```

```

        'A': ('a', 1, 'q_end'),
        'B': ('b', 1, 'q_end'),
        'C': ('c', 1, 'q_end'),
        ' ': (' ', 0, 'qT')
    }

    }

tape = list(input())
tape += [' ' for x in range(14)]
tape = [' '] + tape
tape += [' ']
index = 0
state = 'q_start'

while state != 'qT':
    symb = tape[index]

    new_symb, movement, state = table[state][symb]

    tape[index] = new_symb
    index += movement

print(''.join(tape).replace(' ', ''))

```