

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. 3341

Пчелкин Н.И.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Ознакомиться с принципом работы конечных автоматов в общем и машины Тьюринга в частности, написать на языке Python программу, которая реализует принцип работы машины Тьюринга, решающий определенное задание.

Задание

Вариант 3

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 'с' на символ, следующий за первым встретившимся символом 'а'. Если первый встретившийся символ 'а' 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.

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 отчет включите таблицу состояний. Отдельно кратко опишите каждое состояние, например:

q1 - начальное состояние, которое необходимо, чтобы найти первый встретившийся символ 'с'.

Основные теоретические положения

Конечные автоматы и машина Тьюринга представляют собой две важные концепции в области теории вычислений. Конечные автоматы представляют собой модель вычислительного устройства с ограниченным числом состояний, переходы между которыми происходят в ответ на входные символы. Эти устройства используются для решения задач, связанных с распознаванием и обработкой последовательностей символов. С другой стороны, машина Тьюринга, предложенная Аланом Тьюрингом, является более мощной моделью, способной моделировать любой алгоритмический процесс. Она состоит из бесконечной ленты, на которой могут быть записаны символы, и головки, способной читать и записывать данные. Машина Тьюринга является теоретической основой для понимания вычислимости и лежит в основе многих аспектов современной информатики.

Выполнение работы

Для выполнения поставленной задачи была написана таблица состояний (см. Таблица 1). В каждой её ячейке написаны через запятую:

1. Символ, на который заменят текущий символ;
2. Направление, куда нужно сдвинуться: R- вправо, N – не двигаться, L – влево;
3. На какое состояние следует переместиться.

Таблица 1 – Таблица состояний машины Тьюринга

Состояние	“a”	“b”	“c”	“ ”
“qp”	“a”, R, “q1”	“b”, R, “q0”	“c”, R, “q0”	“ ”, R, “qp”
“q0”	“a”, R, “q1”	“b”, R, “q0”	“c”, R, “q0”	“ ”, N, “qT”
“q1”	“a”, L, “q1a”	“b”, L, “q1b”	“c”, L, “q1c”	“ ”, L, “q1a”
“q1a”	“a”, L, “q1a”	“b”, L, “q1a”	“c”, L, “q1a”	“ ”, R, “q2a”
“q2a”	“a”, R, “q2a”	“b”, R, “q2a”	“c”, L, “q3a”	“ ”, N, “qT”
“q3a”	“a”, N, “qT”	“a”, N, “qT”	“a”, N, “qT”	“a”, N, “qT”
“q1b”	“a”, L, “q1b”	“b”, L, “q1b”	“c”, L, “q1b”	“ ”, R, “q2b”
“q2b”	“a”, R, “q2b”	“b”, R, “q2b”	“c”, L, “q3b”	“ ”, N, “qT”
“q3b”	“b”, N, “qT”	“b”, N, “qT”	“b”, N, “qT”	“b”, N, “qT”
“q1c”	“a”, L, “q1c”	“b”, L, “q1c”	“c”, L, “q1c”	“ ”, R, “q2c”
“q2c”	“a”, R, “q2c”	“b”, R, “q2c”	“c”, L, “q3c”	“ ”, N, “qT”
“q3c”	“c”, N, “qT”	“c”, N, “qT”	“c”, N, “qT”	“c”, N, “qT”

Опишем подробнее назначение каждого состояния:

- qp – поиск начала слова;
- q0 – поиск первого символа “a”. Если таких не нашлось, происходит переход в терминальное состояние;
- q1 – определение, какой символ стоит после “a”; если это “a” или “ ”, происходит переход в состояние “q1a”, если “b” – в “q1b”, если в “c” – в “q1c”. Для каждого из этих случаев идёт свой набор из трёх идентичных состояний;
- q1a (q1b, q1c) – поиск начала слова;

- q_{2a} (q_{2b} , q_{2c}) – поиск первого символа “с”; если такого не находится, происходит переход в терминальное состояние;
- q_{3a} (q_{3b} , q_{3c}) – замена символа до “с” на “а”, “б” или “с” соответственно.

Приведенная выше таблица состояний реализована программно следующим образом. Создаётся список *tape*, в который считывается лента, переменные *state*, в которой будет храниться текущее состояние, и *index*, в которой будет храниться текущий индекс. Далее в цикле *while*($q \neq q_T$), который будет выполняться до тех пор, пока машина не перейдет в терминальное состояние “ q_T ”.

Разработанный программный код см. в приложении А.

Тестирование

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	abcabc	abcabc	Верно
2.	abaaaaacb	abaaaabcb	Верно
3.	aaaaaa	aaaaaa	Верно

Выводы

В ходе выполнения работы были изучены принципы функционирования конечных автоматов и машины Тьюринга. Была написана программа, реализующая механизм работы машины Тьюринга на примере поставленной задачи.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
moves = {'R': 1, 'L': -1, 'N': 0}
```

```
tab = {
    'qn' : {'a': ('a', 'R', 'q1'),
            'b': ('b', 'R', 'q0'),
            'c': ('c', 'R', 'q0'),
            ' ': (' ', 'R', 'qn')
           },
    'q0' : {'a': ('a', 'R', 'q1'),
            'b': ('b', 'R', 'q0'),
            'c': ('c', 'R', 'q0'),
            ' ': (' ', 'N', 'qT')
           },
    'q1' : {'a': ('a', 'L', 'q1a'),
            'b': ('b', 'L', 'q1b'),
            'c': ('c', 'L', 'q1c'),
            ' ': (' ', 'L', 'q1a')
           },
    'q1a' : {'a': ('a', 'L', 'q1a'),
             'b': ('b', 'L', 'q1a'),
             'c': ('c', 'L', 'q1a'),
             ' ': (' ', 'R', 'q2a')
            },
    'q2a' : {'a': ('a', 'R', 'q2a'),
             'b': ('b', 'R', 'q2a'),
             'c': ('c', 'L', 'q3a'),
             ' ': (' ', 'N', 'qT')
            },
    'q3a' : {'a': ('a', 'N', 'qT'),
             'b': ('a', 'N', 'qT'),
             'c': ('a', 'N', 'qT'),
             ' ': ('a', 'N', 'qT')
            },
    'q1b' : {'a': ('a', 'L', 'q1b'),
             'b': ('b', 'L', 'q1b'),
             'c': ('c', 'L', 'q1b'),
             ' ': (' ', 'R', 'q2b')
            },
    'q2b' : {'a': ('a', 'R', 'q2b'),
             'b': ('b', 'R', 'q2b'),
             'c': ('c', 'L', 'q3b'),
             ' ': (' ', 'N', 'qT')
            },
    'q3b' : {'a': ('b', 'N', 'qT'),
             'b': ('b', 'N', 'qT'),
             'c': ('b', 'N', 'qT'),
             ' ': ('b', 'N', 'qT')
            },
    'q1c' : {'a': ('a', 'L', 'q1c'),
             'b': ('b', 'L', 'q1c'),
            }
```

```

        'c': ('c', 'L', 'q1c'),
        ' ': (' ', 'R', 'q2c')
    },
    'q2c': {
        'a': ('a', 'R', 'q2c'),
        'b': ('b', 'R', 'q2c'),
        'c': ('c', 'L', 'q3c'),
        ' ': (' ', 'N', 'qT')
    },
    'q3c': {
        'a': ('c', 'N', 'qT'),
        'b': ('c', 'N', 'qT'),
        'c': ('c', 'N', 'qT'),
        ' ': ('c', 'N', 'qT')
    },
}

tape = list(" " + input() + " ")
index = 0
state = 'qn'

while(state != 'qT'):
    char = tape[index]
    new_char, move, new_state = tab[state][char]

    tape[index] = new_char
    index += moves[move]
    state = new_state

print(''.join(tape).lstrip().rstrip())

```