

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

Преподаватель

Бойцов В.А.

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”	“ ”
“q1”	“a”, N, “q2”	“b”, N, “q2”	“c”, N, “q2”	“ ”, R, “q1”
“q2”	“a”, R, “q3”	“b”, R, “q2”	“c”, R, “q2”	“ ”, N, “qT”
“q3”	“a”, L, “q4”	“b”, L, “q7”	“c”, L, “q10”	“ ”, L, “q4”
“q4”	“a”, L, “q4”	“b”, L, “q4”	“c”, L, “q4”	“ ”, R, “q5”
“q5”	“a”, R, “q5”	“b”, L, “q5”	“c”, L, “q6”	“ ”, N, “qT”
“q6”	“a”, N, “qT”	“a”, N, “qT”	“a”, N, “qT”	“a”, N, “qT”
“q7”	“a”, L, “q7”	“b”, L, “q7”	“c”, L, “q7”	“ ”, R, “q8”
“q8”	“a”, R, “q8”	“b”, R, “q8”	“c”, L, “q9”	“ ”, N, “qT”
“q9”	“b”, N, “qT”	“b”, N, “qT”	“b”, N, “qT”	“b”, N, “qT”
“q10”	“a”, L, “q10”	“b”, L, “q10”	“c”, L, “q10”	“ ”, R, “q11”
“q11”	“a”, R, “q11”	“b”, R, “q11”	“c”, L, “q12”	“ ”, N, “qT”
“q12”	“c”, N, “qT”	“c”, N, “qT”	“c”, N, “qT”	“c”, N, “qT”

Опишем подробнее назначение каждого состояния:

- q1 – поиск начала слова;
- q2 – поиск первого символа “a”. Если таких не нашлось, происходит переход в терминальное состояние;
- q3 – определение, какой символ стоит после “a”; если это “a” или “ ”, происходит переход в состояние “q4”, если “b” – в “q7”, если в “c” – в “q10”. Для каждого из этих случаев идёт свой набор из трёх идентичных состояний;
- q4 (q7, q10) – поиск начала слова;
- q5 (q8, q11) – поиск первого символа “c”; если такого не находится, происходит переход в терминальное состояние;

- q6 (q9, q12) – замена символа до “с” на “а”, “b” или “с” соответственно.

Приведенная выше таблица состояний реализована программно следующим образом. Создаётся список *memory*, в который считывается лента, переменные *q*, в которой будет храниться текущее состояние, и *ind*, в которой будет храниться текущий индекс. Далее в огромном цикле *while(q!='qT')*, который будет выполняться до тех пор, пока машина не перейдет в терминальное состояние “qT”, каждое состояние реализовано следующим образом:

```
if q=='q1':  
    if memory[ind]=='a':  
        memory[ind]='a'  
        ind+=0  
        q='q2'
```

После выполнения цикла создаётся строка *ans*, в которую посимвольно записываются элементы ленты *memory*, после чего *ans* выводится на экран.

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

```
memory=list(input())
q='q1'
ind=0
while(q!='qT'):
    if q=='q1':
        if memory[ind]=='a':
            memory[ind]='a'
            ind+=0
            q='q2'
        elif memory[ind]=='b':
            memory[ind]='b'
            ind+=0
            q='q2'
        elif memory[ind]=='c':
            memory[ind]='c'
            ind+=0
            q='q2'
        elif memory[ind]==' ':
            memory[ind]=' '
            ind+=1
            q='q1'
    elif q=='q2':
        if memory[ind]=='a':
            memory[ind]='a'
            ind+=1
            q='q3'
        elif memory[ind]=='b':
            memory[ind]='b'
            ind+=1
            q='q2'
        elif memory[ind]=='c':
            memory[ind]='c'
            ind+=1
            q='q2'
        elif memory[ind]==' ':
            memory[ind]=' '
            ind+=0
            q='qT'
    elif q=='q3':
        if memory[ind]=='a':
            memory[ind]='a'
            ind-=1
            q='q4'
        elif memory[ind]=='b':
            memory[ind]='b'
            ind-=1
            q='q7'
        elif memory[ind]=='c':
            memory[ind]='c'
            ind-=1
```

```

        q='q10'
    elif memory[ind]==' ':
        memory[ind]=' '
        ind-=1
        q='q4'
elif q=='q4':
    if memory[ind]=='a':
        memory[ind]='a'
        ind-=1
        q='q4'
    elif memory[ind]=='b':
        memory[ind]='b'
        ind-=1
        q='q4'
    elif memory[ind]=='c':
        memory[ind]='c'
        ind-=1
        q='q4'
    elif memory[ind]==' ':
        memory[ind]=' '
        ind+=1
        q='q5'
elif q=='q5':
    if memory[ind]=='a':
        memory[ind]='a'
        ind+=1
        q='q5'
    elif memory[ind]=='b':
        memory[ind]='b'
        ind+=1
        q='q5'
    elif memory[ind]=='c':
        memory[ind]='c'
        ind-=1
        q='q6'
    elif memory[ind]==' ':
        memory[ind]=' '
        ind+=0
        q='qT'
elif q=='q6':
    if memory[ind]=='a':
        memory[ind]='a'
        ind+=0
        q='qT'
    elif memory[ind]=='b':
        memory[ind]='a'
        ind+=0
        q='qT'
    elif memory[ind]=='c':
        memory[ind]='a'
        ind+=0
        q='qT'
    elif memory[ind]==' ':
        memory[ind]='a'
        ind+=0
        q='qT'
elif q=='q7':
    if memory[ind]=='a':

```

```

        memory[ind]='a'
        ind-=1
        q='q7'
    elif memory[ind]=='b':
        memory[ind]='b'
        ind-=1
        q='q7'
    elif memory[ind]=='c':
        memory[ind]='c'
        ind-=1
        q='q7'
    elif memory[ind]==' ':
        memory[ind]=' '
        ind+=1
        q='q8'
elif q=='q8':
    if memory[ind]=='a':
        memory[ind]='a'
        ind+=1
        q='q8'
    elif memory[ind]=='b':
        memory[ind]='b'
        ind+=1
        q='q8'
    elif memory[ind]=='c':
        memory[ind]='c'
        ind-=1
        q='q9'
    elif memory[ind]==' ':
        memory[ind]=' '
        ind+=0
        q='qT'
elif q=='q9':
    if memory[ind]=='a':
        memory[ind]='b'
        ind+=0
        q='qT'
    elif memory[ind]=='b':
        memory[ind]='b'
        ind+=0
        q='qT'
    elif memory[ind]=='c':
        memory[ind]='b'
        ind+=0
        q='qT'
    elif memory[ind]==' ':
        memory[ind]='b'
        ind+=0
        q='qT'

elif q=='q10':
    if memory[ind]=='a':
        memory[ind]='a'
        ind-=1
        q='q10'
    elif memory[ind]=='b':
        memory[ind]='b'
        ind-=1

```

```

        q='q10'
    elif memory[ind]=='c':
        memory[ind]='c'
        ind-=1
        q='q10'
    elif memory[ind]==' ':
        memory[ind]=' '
        ind+=1
        q='q11'
elif q=='q11':
    if memory[ind]=='a':
        memory[ind]='a'
        ind+=1
        q='q11'
    elif memory[ind]=='b':
        memory[ind]='b'
        ind+=1
        q='q11'
    elif memory[ind]=='c':
        memory[ind]='c'
        ind-=1
        q='q12'
    elif memory[ind]==' ':
        memory[ind]=' '
        ind+=0
        q='qT'
elif q=='q12':
    if memory[ind]=='a':
        memory[ind]='c'
        ind+=0
        q='qT'
    elif memory[ind]=='b':
        memory[ind]='c'
        ind+=0
        q='qT'
    elif memory[ind]=='c':
        memory[ind]='c'
        ind+=0
        q='qT'
    elif memory[ind]==' ':
        memory[ind]='c'
        ind+=0
        q='qT'
ans=''
for x in memory:
    ans=ans+x
print(ans)

```