**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

|  |  |  |
| --- | --- | --- |
| Студен гр. 3343 |  | Кербель Д. А. |
| Преподаватель |  | Иванов Д. В. |

Санкт-Петербург

2023

**Цель работы**

Разобраться в устройстве работы машины Тьюринга и написать ее прототип на языке программирования Python.

**Задание**

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}.

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

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.

Для примера выше лента будет выглядеть так:

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

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.

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 обнаружить конец строки.

## Выполнение работы

**Таблица состояний:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ‘a’ | ‘b’ | ‘c’ | ‘!’ | ‘ ’ |
| q1 | "a", R, "q2" | "b", R, "q2" | "c", R, "q2" |  | " ", R, "q1" |
| q2 | "a", R, "q2" | "b", R, "q2" | "c", R, "q2" |  | " ", L, "q3" |
| q3 | "!", N, "q4" | "!", N, "q6" | "!", N, "q8" |  |  |
| q4 | "a", R, "q4" | "b", R, "q4" | "c", R, "q4" | "!", R, "q4" | " ", R, "q5" |
| q5 | "a", R, "q5" | "b", R, "q5" | "c", R, "q5" |  | "a", L, "q10" |
| q6 | "a", R, "q6" | "b", R, "q6" | "c", R, "q6" | "!", R, "q6" | " ", R, "q7" |
| q7 | "a", R, "q7" | "b", R, "q7" | "c", R, "q7" |  | "b", L, "q10" |
| q8 | "a", R, "q8" | "b", R, "q8" | "c", R, "q8" | "!", R, "q8" | " ", R, "q9" |
| q9 | "a", R, "q9" | "b", R, "q9" | "c", R, "q9" |  | "c", L, "q10" |
| q10 | "a", L, "q10" | "b", L, "q10" | "c", L, "q10" |  | " ", L, "q11" |
| q11 | "!", N, "q4" | "!", N, "q6" | "!", N, "q8" | "!", L, "q11" | " ", R, "q12" |
| q12 |  |  |  | " ", R, "q12" | " ", N, "0" |

**Следующая таблица состояний, включает 12 различных состояний для машины Тьюринга**

1. q1 – перемещение к первому символу строки
2. q2 – перемещение к последнему символу строки
3. q3 – замена последнего символа строки на символ «!». Если ячейка содержит символ «a», то вызывается q4, если «b», то q6, если «с», то q8
4. q4 – перемещение к первому символу перевернутой строки, вызов q5
5. q5 – перемещение к первому пробелу после строки, запись символа «a», переход к концу инвертированной строки
6. q6 – аналогично q4, но вызов q7
7. q7 – аналогично q5, но запись «b»
8. q8 – как q4, но вызов q9
9. q9 – абсолютно аналогично q5, но запись «с»
10. q10 – перемещение от конца перевернутой строки к начальной
11. q11 – обход строке, но при нахождении символов «a», «b», «c» вызов к начальному состоянияю, в противном случае переход к началу строки.
12. q12 – удаление всех символов «!» и выход из алгоритма.
13. В начале дается некоторое описание таблицы состояний в виде словаря, где ключ – состояние, значение – ещё один словарь, в котором ключ – символ на ленте, а значение – список, включающий новый символ, направление движения и новое состояние.

На первом этапе определяются начальные значения переменных:

L = -1: переменная, обозначающая направление движения на ленте;

N = 0: переменная, обозначающая отсутствие движения на ленте;

R = 1: переменная, обозначающая движение вправо на ленте;

table = list(" "\*20): переменная, представляющая ленту, инициализированную пробелами;

state = 'q1': переменная, обозначающая начальное состояние машины Тьюринга;

indx = 0: переменная, обозначающая текущий индекс на ленте;

array = table + list(input()) + table: переменная, представляющая состояние ленты, к которой добавляются элементы, введенные пользователем;

Затем программа выполняет цикл, пока состояние state не станет равным '0':

Для текущего состояния state и текущего символа на ленте array[indx] программа получает значение из словаря tape[state][array[indx]], которое представляет следующее действие машины Тьюринга.

Программа изменяет текущий символ на ленте array[indx] на значение dict[0] (новый символ на ленте).

Программа изменяет текущий индекс на ленте indx на значение dict[1] (новый индекс на ленте).

Программа изменяет текущее состояние state на значение dict[2] (новое состояние машины Тьюринга).

После завершения цикла программа выводит содержимое ленты array без разделителей на экран.

Разработанный программный код см. в приложении А.

## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
|  | aaaaaaaaaaaaabbc | aaaaaaaaaaaaabbc | Выходные данные соответствуют ожиданиям. |
|  | aaaaacc | ccaaaaa | Выходные данные соответствуют ожиданиям. |
|  | bcaaaabaccc | cccabaaaacb | Выходные данные соответствуют ожиданиям. |

## Выводы

В ходе выполнения лабораторной работы, мною был освоен и изучен метод работы машины Тьюринга. Написанная программа успешно справляется с поставленными задачами.

# Приложение А Исходный код программы

Название файла: main.py

#На первом этапе определяются начальные значения переменных

R = 1 #L = -1: переменная, обозначающая направление движения на ленте

N = 0 #N = 0: переменная, обозначающая отсутствие движения на ленте

L = -1 #R = 1: переменная, обозначающая движение вправо на ленте

table = list(" "\*50) #table = list(" "\*20): переменная, представляющая ленту, инициализированную пробелами;

state = 'q1' #state = 'q1': переменная, обозначающая начальное состояние машины Тьюринга

indx = 0 #indx = 0: переменная, обозначающая текущий индекс на ленте

array = table + list(input()) + table #array = table + list(input()) + table: переменная, представляющая состояние ленты, к которой добавляются элементы, введенные пользователем

tape = {'q1': {'a':['a',R,'q2'],'b':['b',R,'q2'],'c':['c',R,'q2'],' ':[' ',R,'q1']},

'q2': {'a':['a',R,'q2'],'b':['b',R,'q2'],'c':['c',R,'q2'],' ':[' ',N,'q3']},

'q3': {'a':['d',N,'q4'],'b':['d',N,'q6'],'c':['d',N,'q8'],' ':[' ',L,'q3']},

'q4': {'a':['a',R,'q4'],'b':['b',R,'q4'],'c':['c',R,'q4'],'d':['d',R,'q4'],' ':[' ',R,'q5']},

'q5': {'a':['a',R,'q5'],'b':['b',R,'q5'],'c':['c',R,'q5'],' ':['a',L,'q10']},

'q6': {'a':['a',R,'q6'],'b':['b',R,'q6'],'c':['c',R,'q6'],'d':['d',R,'q6'],' ':[' ',R,'q7']},

'q7': {'a':['a',R,'q7'],'b':['b',R,'q7'],'c':['c',R,'q7'],' ':['b',L,'q10']},

'q8': {'a':['a',R,'q8'],'b':['b',R,'q8'],'c':['c',R,'q8'],'d':['d',R,'q8'],' ':[' ',R,'q9']},

'q9': {'a':['a',R,'q9'],'b':['b',R,'q9'],'c':['c',R,'q9'],' ':['c',L,'q10']},

'q10': {'a':['a',L,'q10'],'b':['b',L,'q10'],'c':['c',L,'q10'],' ':[' ',L,'q11']},

'q11': {'a':['d',N,'q4'],'b':['d',N,'q6'],'c':['d',N,'q8'],'d':['d',L,'q11'],' ':[' ',R,'q12']},

'q12': {'d':[' ',R,'q12'],' ':[' ',N,'0']}}

#Затем программа выполняет цикл, пока состояние state не станет равным '0'

while state != '0':

dict = tape[state][array[indx]] #Для текущего состояния state и текущего символа на ленте array[indx] программа получает значение из словаря tape[state][array[indx]], которое представляет следующее действие машины Тьюринга

array[indx] = dict[0] #Программа изменяет текущий символ на ленте array[indx] на значение dict[0] (новый символ на ленте)

indx += dict[1] #Программа изменяет текущий индекс на ленте indx на значение dict[1] (новый индекс на ленте)

state = dict[2] #Программа изменяет текущее состояние state на значение dict[2] (новое состояние машины Тьюринга)

print(\*array, sep='') #Вывод результата