**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. 3344 |  | Волков А.А. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Изучить механизм работы машины Тьюринга, освоить табличный способ записи программ для машины Тьюринга, реализовать алгоритм для решения поставленной задачи при помощи машины Тьюринга, имитировав её работу в Python.

## Задание

Вариант 3.

Напишите программу, которая заменяет в исходной строке символ, предшествующий первому встретившемуся символу ‘c' на символ, следующий за первым встретившимся символом ‘a'. Если первый встретившийся символ ‘a' 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.

Алфавит:

1) a

2) b

3) c

4) " " (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 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.

## Выполнение работы

Перед написание алгоритма, необходимо составить таблицу состояний для МТ, которая описывает действия для каждого состояния и каждого символа алфавита по отдельности. В ячейке записана тройка: записываемый в текущую ячейку символ, направление следующего шага (R – вправо, L – влево, N – остаться на месте), следующее состояние.

Таблица для решения задачи (табл. 1):

Таблица 1 – Таблица состояний для машины Тьюринга

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Состояние** | **a** | **b** | **c** | **“ “ (пробел)** |
| **q1** | a, R, q3 | b, R, q1 | c, R, q2 | “ “, R, q1 |
| **q2** | a, R, q4 | b, R, q2 | c, R, q2 |  |
| **q3** | a, R, q14 | b, R, q15 | c, L, q16 |  |
| **q4** | a, L, q5 | b, L, q7 | c, L, q6 | “ “, L, q5 |
| **q5** | a, L, q5 | b, L, q5 | c, L, q5 | “ ”, R, q8 |
| **q6** | a, L, q6 | b, L, q6 | c, L, q6 | “ “, R, q10 |
| **q7** | a, L, q7 | b, L, q7 | c, L, q7 | “ “, R, q12 |
| **q8** | a, R, q8 | b, R, q8 | c, L, q9 |  |
| **q9** | a, N, qT | a, N, qT | a, N, qT | a, N, qT |
| **q10** | a, R, q10 | b, R, q10 | c, L, q11 |  |
| **q11** | c, N, qT | c, N, qT | c, N, qT | c, N, qT |
| **q12** | a, R, q12 | b, R, q12 | c, L, q13 |  |
| **q13** | b, N, qT | b, N, qT | b, N, qT | b, N, qT |
| **q14** | a, R, q14 | b, R, q14 | c, L, q17 |  |
| **q15** | a, R, q15 | b, R, q15 | c, L, q18 |  |
| **q16** | c, N, qT |  |  |  |
| **q17** | a, N, qT | a, N, qT | a, N, qT |  |
| **q18** | b, N, qT | b, N, qT | b, N, qT |  |

Описание состояний:

q1 – состояние для нахождения того, что встретилось первым («**a**» или «**c**»).

q2 – попадаем, когда «**c**» встретилась раньше «**a**», и находим первое вхождение «**a**».

q3 – попадаем, когда «**а**» встретилась раньше «**с**», переходим в нужные состояния в зависимости от буквы, которая оказалась после «**а**».

*Группа состояний для случая q2:*

q4 – в зависимости от символа после «**а**» переходим в нужное состояние.

q5 – состояние, куда попадаем, когда надо заменить символ до «**с**» на «**а**». Идём до начала слова (движемся влево), а затем переходим в состояние **q8**.

q6 – состояние, куда попадаем, когда надо заменить символ до «**с**» на «**с**». Идём до начала слова (движемся влево), а затем переходим в состояние **q10**.

q7 – состояние, куда попадаем, когда надо заменить символ до «**с**» на «**b**». Идём до начала слова (движемся влево), а затем переходим в состояние **q12**.

q8 – ищем первое вхождение «**с**», затем переходим в состояние **q9**, сдвинув автомат влево.

q9 – заменяем текущий символ «а» и переходим в терминальное состояние **qT**, т.е завершаем работу.

q10 – ищем первое вхождение «**с**», затем переходим в состояние **q11**, сдвинув автомат влево.

q11 – заменяем текущий символ «**с**» и переходим в терминальное состояние **qT**, т.е завершаем работу.

q12 – ищем первое вхождение «**с**», затем переходим в состояние **q13**, сдвинув автомат влево.

q13 – заменяем текущий символ «**b**» и переходим в терминальное состояние **qT**, т.е завершаем работу.

*Группа состояний для случая q3:*

q14 – ищем первое вхождение «**с**», затем переходим в состояние **q17**, сдвинув автомат влево.

q15 – ищем первое вхождение «**с**», затем переходим в состояние **q18**, сдвинув автомат влево.

q16 – попадаем, когда «**а**» стоит перед «**с**». Предварительно сдвинув автомат влево в состояние **q3**, заменяем предшествующую «**а**» на «**с**». Переходим в терминальное состояние **qT**, т.е завершаем работу.

q17 – заменяем текущий символ на «**а**» и переходим в терминальное состояние **qT**, т.е завершаем работу.

q18 – заменяем текущий символ на «**b**» и переходим в терминальное состояние **qT**, т.е завершаем работу.

После создания таблицы можно спокойно реализовать программу на Python, которая будет имитировать работу МТ. Создаём словарь (*table*) и заводим переменные для хранения текущего состояния (*state*), ленты, воссозданной в виде списка строк (*memory*), номера ячейки на ленте, на которой стоит в данный момент автомат (*index*). Запускаем цикл *while* до того момента, пока *state* не примет значение qT, обращаемся по ключам ячейкам таблицы, записывая новый символ (*symbol*), шаг автомата (*step*), и новое состояние (*new\_state*). Записываем новый символ на ленту, обновляем текущее состояние, делаем шаг. Итоговая лента выводится на экран.

Результаты тестирования представлены в табл. 2, раздел «Тестирование».

## Тестирование

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № | Входные данные | Выходные данные | Список состояний |
| 1. | « cbbba » | « acbbba » | q1 q1 q1 q1 q2 q2 q2 q2 q4 q5 q5 q5 q5 q5 q5 q8 q9 qT |
| 2. | « abcabc » | « abcabc » | q1 q1 q1 q1 q1 q1 q1 q1 q1 q1 q1 q1 q1 q1 q1 q3 q15 q18 qT |
| 3. | « accabcbabaacab » | « cccabcbabaacab » | q1 q1 q1 q1 q3 q16 qT |

## Выводы

Был изучен механизм работы машины Тьюринга и способ табличный записи программы для неё. При помощи этого был разработан алгоритм решения для поставленной задачи по замене необходимых символов в строке на ленте.

Была разработана программа на Python, которая имитирует работу машины Тьюринга, где лента представлена списком строк, а таблица реализована словарём, в котором первый ключ – состояние, второй – текущий символ.

# Приложение А Исходный код программы

Название файла: lb\_3.py

L, N, R = -1, 0, 1

table = {

"q1": {

"a": ("a", R, "q3"),

"b": ("b", R, "q1"),

"c": ("c", R, "q2"),

" ": (" ", R, "q1"),

},

"q2": {

"a": ("a", R, "q4"),

"b": ("b", R, "q2"),

"c": ("c", R, "q2"),

},

"q3": {

"a": ("a", R, "q14"),

"b": ("b", R, "q15"),

"c": ("c", L, "q16"),

},

"q4": {

"a": ("a", L, "q5"),

"b": ("b", L, "q7"),

"c": ("c", L, "q6"),

" ": (" ", L, "q5"),

},

"q5": {

"a": ("a", L, "q5"),

"b": ("b", L, "q5"),

"c": ("c", L, "q5"),

" ": (" ", R, "q8"),

},

"q6": {

"a": ("a", L, "q6"),

"b": ("b", L, "q6"),

"c": ("c", L, "q6"),

" ": (" ", R, "q10"),

},

"q7": {

"a": ("a", L, "q7"),

"b": ("b", L, "q7"),

"c": ("c", L, "q7"),

" ": (" ", R, "q12"),

},

"q8": {

"a": ("a", R, "q8"),

"b": ("b", R, "q8"),

"c": ("c", L, "q9")

},

"q9": {

"a": ("a", N, "qT"),

"b": ("a", N, "qT"),

"c": ("a", N, "qT"),

" ": ("a", N, "qT"),

},

"q10": {

"a": ("a", R, "q10"),

"b": ("b", R, "q10"),

"c": ("c", L, "q11"),

},

"q11": {

"a": ("c", N, "qT"),

"b": ("c", N, "qT"),

"c": ("c", N, "qT"),

" ": ("c", N, "qT"),

},

"q12": {

"a": ("a", R, "q12"),

"b": ("b", R, "q12"),

"c": ("c", L, "q13"),

},

"q13": {

"a": ("b", N, "qT"),

"b": ("b", N, "qT"),

"c": ("b", N, "qT"),

" ": ("b", N, "qT"),

},

"q14": {

"a": ("a", R, "q14"),

"b": ("b", R, "q14"),

"c": ("c", L, "q17"),

},

"q15": {

"a": ("a", R, "q15"),

"b": ("b", R, "q15"),

"c": ("c", L, "q18"),

},

"q16": {

"a": ("c", N, "qT"),

},

"q17": {

"a": ("a", N, "qT"),

"b": ("a", N, "qT"),

"c": ("a", N, "qT"),

},

"q18": {

"a": ("b", N, "qT"),

"b": ("b", N, "qT"),

"c": ("b", N, "qT"),

},

}

state = "q1"

memory = list(input())

index = 0

while state != "qT":

symbol, step, new\_state = table[state][memory[index]]

memory[index] = symbol

state = new\_state

index += step

print("".join(memory))