**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. Вариант 1

|  |  |  |
| --- | --- | --- |
| Студент гр. 3343 |  | Коршков А.А. |
| Преподаватель |  | Иванов Д. В. |

Санкт-Петербург

2023

## Цель работы

Научится создавать модель машины Тьюринга на языке Python, с применением его встроенных функций, писать алгоритм для обработки строки на ленте.

## 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}.

Напишите программу, которая удаляет в исходной строке два символа, следующих за первым встретившимся символом ‘b’. Если первый встретившийся символ ‘b’ – последний в строке, то удалить его. Если первый встретившийся символ ‘b’ – предпоследний в строке, то удалить один символ, следующий за ним, т. е. последний в строке. Если в строке символ ‘b’ отсутствует, то удалить самый первый символ строки. После удаления в строке не должно оставаться пробелов и пустых мест!

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:

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.

Ваша программа должна вывести полученную ленту после завершения работы.

## Выполнение работы

На вход подаётся строка, которая расположена на ленте. В начальном состоянии (q0) ищем символ «b», чтобы перейти в состояние удаления (q2) или символы «a» и «c», чтобы перейти в состояние «q1». Если мы встретили символ «b», то переходим в состояние «q2».

Если в состоянии q1, мы не нашли символа «b», тогда переходим в состояние поиска (qDs) и ищем пробел, чтобы найти начало строки и заменить первый символ на пробел, после чего завершить программу.

Если был найден символ b, тогда переходим в состояние q2, делаем проверку, что следующий символ не пробел (символ b не является конечным). Если это пробел, тогда переходим в состояние «qD», которое удалит b и завершит программу.

Если символ не является пробелом, а какой-то буквой, тогда с помощью состояний «q2» и «q2.1» удаляем два впереди стоящих символа (а если второй символ является пробелом, то ничего не меняется), переходим к следующему символу и переключаем состояние поиска(qS) и перемещения символа.

Если буквенных символов больше нет, а стоит пробел, тогда достаточно просто завершить программу. В ином случае необходимо узнать, какой символ стоит, заменить этот символ на пробел, сдвинуть головку влево и поменять состояние на «qA» или «qB» или «qC». Затем головка перемещается влево и меняет состояние на «qA1» или «qB1» или «qC1». В этих состояниях пробел заменяется на нужный символ, головка перемещается вправо, а состояние меняется на состояние возврата «qP». В этом состоянии головка перемещается вправо (на изначальное место символа), а состояние меняется на «qP1». После этого происходит перемещение головки вправо на позицию следующего символа (или пробела, если происходило перемещение последнего символа в строке), а состояние меняется на «qS».

Таблица 1 – Таблица состояний

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a | b | c | " " |
| q0 | "a"; R; q1 | "b"; R; q2 | "c"; R; q1 | " "; R; q0 |
| q1 | "a"; R; q1 | "b"; R; q2 | "c"; R; q1 | " "; L; qDs |
| q2 | " "; R; q2.1 | " "; R; q2.1 | " "; R; q2.1 | " "; L; qD |
| q2.1 | " "; R; qS | " "; R; qS | " "; R; qS | " "; R; qS |
| qS | " "; L; qA | " "; L; qB | " "; L; qC | " "; N; qT |
| qA | - | - | - | " "; L; qA1 |
| qB | - | - | - | " "; L; qB1 |
| qC | - | - | - | " "; L; qC1 |
| qA1 | - | - | - | "a"; R; qP |
| qB1 | - | - | - | "b"; R; qP |
| qC1 | - | - | - | "c"; R; qP |
| qP | - | - | - | " "; R; qP1 |
| qP1 | - | - | - | " "; R; qS |
| qD | " "; N; qT | " "; N; qT | " "; N; qT | " "; N; qT |

Краткое описание состояний данной машины Тьюринга:

1) «q0» – начальное состояние

2) «qT» – терминальное (конечное состояние)

3) «q1» – состояние, когда был найден символ, отличный от «b» и пробела

4) «qDs» – состояние, когда не были найдены символы «b» (в состоянии q1 встретили пробел), ищет первый символ строки.

5) «qD» – удаляет один символ.

6) «q2» – состояние для удаления первого символа после «b».

7) «q2.1» – состояние для удаления второго символа после «b».

8) «qS» – состояние для проверки существования символа и запуска алгоритма перемещения этого символа на пустое место в строке.

9) «qA», «qB», «qC» – состояния для сохранения перемещённого символа и перемещение к первому удалённому символу.

10) «qA1», «qB1», «qC1» – заменяют символ на нужный и перемещают головку вправо.

11) «qP» – перемещает головку на изначальное место перенесённого символа.

12) «qP1» – перемещает головку на следующий символ.

## Тестирование

Таблица 2 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № | Входные данные | Выходные данные | Комментарии |
| 1 | " abcabc " | " abbc " | Удалены символы «ca» после символа «b» |
| 2 | " cac " | " ac " | Удалён первый символ «с» (нет символов «b») |
| 3 | " cabc " | " cab " | Удалён последний символ «с» (b - предпоследний символ) |
| 4 | " cab " | " ca " | Удалён символ «b» (конец строки) |

## Выводы

Был изучен способ создания модели машины Тьюринга на языке Python, программа обрабатывает текст и в зависимости от присутствия и места нахождения символа «b» удаляет символа и сдвигает оставшуюся часть строки на необходимое положение, при котором не существует символов пробела внутри строки.

# Приложение А Исходный код программы

Название файла: main.py

memory = list(input())

q = "q0"

idx = 0

while q != "qT":

if q == "q0":

if memory[idx] == "b":

idx += 1

q = "q2"

elif memory[idx] != " ":

idx += 1

q = "q1"

else:

idx += 1

elif q == "q1":

if memory[idx] == "b":

idx += 1

q = "q2"

elif memory[idx] == " ":

idx -= 1

q = "qDs"

else:

idx += 1

elif q == "q2":

if memory[idx] == " ":

idx -= 1

q = "qD"

else:

memory[idx] = " "

idx += 1

q = "q2.1"

elif q == "q2.1":

memory[idx] = " "

idx += 1

q = "qS"

elif q == "qS":

if memory[idx] == " ":

q = "qT"

elif memory[idx] == "a":

memory[idx] = " "

idx -= 1

q = "qA"

elif memory[idx] == "b":

memory[idx] = " "

idx -= 1

q = "qB"

elif memory[idx] == "c":

memory[idx] = " "

idx -= 1

q = "qC"

elif q == "qA" or q == "qB" or q == "qC":

idx -= 1

q = q + "1"

elif q == "qA1":

memory[idx] = "a"

idx += 1

q = "qP"

elif q == "qB1":

memory[idx] = "b"

idx += 1

q = "qP"

elif q == "qC1":

memory[idx] = "c"

idx += 1

q = "qP"

elif q == "qP":

idx += 1

q = "qP1"

elif q == "qP1":

idx += 1

q = "qS"

elif q == "qDs":

if memory[idx] != " ":

idx -= 1

else:

idx += 1

q = "qD"

elif q == "qD":

memory[idx] = " "

q = "qT"

print("".join(memory))