**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 |  | Гусева Е.А. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Изучение принципов работы машины Тьюринга и написание программы для машины Тьюринга.

## Задание.

Вариант 1

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. 3

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

Были созданы переменные tape – содержащая входные данные, служащая памятью машины, current\_state – хранящая состояние машины на каждом такте. Был реализован словарь состояний, соответствующий таблице состояний.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ‘ ’(пробел) | ‘a’ | ‘b’ | ‘с’ |
| q1 | ‘ ’, R, q1 | ‘a’, N, qsearch | ‘b’, R, qdell | ‘c’, N, qsearch |
| qsearch | ‘ ’, L, qfirst | ‘a’, R, qsearch | ‘b’, R, qdel1 | ‘c’, R, qsearch |
| qfirst | ‘ ’, R, adel2 | ‘’a, L, qfirst |  | ‘c’, L, qsearch |
| qdel1 | ‘ ’, Ladel2 | ‘’, R, qdel2 | ‘’, R, qdel2 | ‘’, R, qdel2 |
| qdel2 | ‘ ’, N, qt | ‘’, N, qt | ‘’, N, qt | ‘’, N, qt |

q1 – начальное состояние, поиск первого символа, не являющегося пробелом, если первый символ ‘b’- то осуществляется переход к состоянию удаления символов. qsearch – состояние поиска первого символа ‘b’ во входной строке, если в строке нет символа ‘b’ - переход к состоянию удаления первого символа во входной строке. qfirst – состояние удаления первого символа, не являющегося пробелом, в строке. qdel1 – состояние удаления первого из двух символов после первого встреченного символа ‘b’. qdel2 – состояние удалениявторого из двух символов после первого встреченного символа ‘b’.

## Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
|  | aabbcc | aabc | - |
|  | ccbbaa | ccba | - |

## Выводы

Был изучен принцип работы машины Тьюринга, написан алгоритм работы машины и составлена таблица состояний для выполнения поставленной задачи.

# Приложение А Исходный код программы

Название файла: main.py

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

tape = list(input())

current\_state = 'q1'

head\_position = 0

asd={

'q1':{

' ': [' ', R, 'q1'],

'a': ['a', N, 'qsearch'],

'b': ['b', R, 'qdel1'],

'c': ['c', N, 'qsearch']

},

'qsearch': {

' ': [' ', L, 'qfirst'],

'a': ['a', R, 'qsearch'],

'b': ['b', R, 'qdel1'],

'c': ['c', R, 'qsearch']

},

'qfirst':{

' ': [' ', R, 'qdel2'],

'a': ['a', L, 'qfirst'],

'c': ['c', L, 'qfirst']

},

'qdel1': {

' ': [' ', L, 'qdel2'],

'a': ['', R, 'qdel2'],

'b': ['', R, 'qdel2'],

'c': ['', R, 'qdel2']

},

'qdel2':{

'a': ['', N, 'qt'],

'b': ['', N, 'qt'],

'c': ['', N, 'qt'],

' ': [' ', N, 'qt']

}

}

while current\_state != 'qt':

symbol, move\_direction, next\_state = asd[current\_state][tape[head\_position]]

tape[head\_position] = symbol

head\_position+=move\_direction

current\_state=next\_state

print(\*tape, sep='')