**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. 3343 Калиберов Н.И.

|  |  |  |
| --- | --- | --- |
| Преподаватель |  | Иванов Д. В. |

Санкт-Петербург

2023

**Цель работы**

Понять принцип работы машины Тьюринга и конечных автоматов и научиться использовать их для решения практических задач

**Задание**

**Вариант 2.**

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, идущий после последних двух встретившихся символов 'a', на предшествующий им символ(гарантируется, что это не пробел). Наличие в строке двух подряд идущих символов 'a' гарантируется.

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 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 отчете предоставьте таблицу состояний. Отдельно кратко опишите каждое состояние, например: q1 - начальное состояние, которое необходимо, чтобы найти первый

встретившийся символ ‘a’.

**Выполнение работы**

Для начала создаём словарь table, который хранит описание всех состояний. Начальное состояние машины - ‘q0’.

Таблица состояний представлена в табл. 1.

Таблица 1 – Состояния

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Состояние | ‘a’ | ‘b’ | ‘c’ | ‘ ‘ | Описание |
| q0 | a,R,q1 | b,R,q1 | c,R,q1 | ‘ ’,R,q0 | Ищем первый символ |
| q1 | a,R,q1 | b,R,q1 | c,R,q1 | ‘ ‘,R,q2 | Доходим до последнего символа |
| q2 | a,L,q3 | b,L,q2 | c,L,q2 |  | Ищем первую  ‘a’ |
| q3 | a,L,q4 | b,L,q2 | c,L,q2 |  | Ищем вторую  ‘a’ |
| q4 | a,R,q5 | b,R,q6 | c,R,q7 |  | Смотрим какой символ слева от ‘aa’ |
| q5 | a,R,q5 | a,N,qT | a,N,qT | ‘ ‘,N,qT | Если символ слева ‘a’ заменяем его справа на ‘a’ |
| q6 | a,R,q6 | b,N,qT | b,N,qT | ‘ ’,N,qT | Если символ слева ‘b’ заменяем его справа на ‘b’ |
| q7 | a,R,q7 | c,N,qT | c,N,qT | ‘ ‘,N,qT | Если символ слева ‘c’  заменяем его справа на  ‘c’ |

**Тестирование.**

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Входные данные | Выходные данные | Комментарии |
| 1. | acaabcbabaacab | acaabcbabaabab | Верно |
| 2. | abcaabc | abcaacc | Верно |
| 3. | aaaa | aaaaa | Верно |
| 4. | cabcaabb | cabcaacb | Верно |

**Выводы**

Было изучено строение машины Тьюринга и других конечных автоматов, получено представление об архитектуре компьютера на более низком уровне.

**ПРИЛОЖЕНИЕ А**

**ИСХОДНЫЙ КОД ПРОГРАММЫ**

Название файла: main.py

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

table = {

'q0': {'a': ['a', R, 'q1'], 'b': ['b', R, 'q1'], 'c': ['c', R, 'q1'], ' ': [' ', R, 'q0']},

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

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

'q3': {'a': ['a', L, 'q4'], 'b': ['b', L, 'q2'], 'c': ['c', L, 'q2']},

'q4': {'a': ['a', R, 'q5'], 'b': ['b', R, 'q6'], 'c': ['c', R, 'q7']},

'q5': {'a': ['a', R, 'q5'], 'b': ['a', N, 'qT'], 'c': ['a', N, 'qT'], ' ': ['a', N, 'qT']},

'q6': {'a': ['a', R, 'q6'], 'b': ['b', N, 'qT'], 'c': ['b', N, 'qT'], ' ': ['b', N, 'qT']},

'q7': {'a': ['a', R, 'q7'], 'b': ['c', N, 'qT'], 'c': ['c', N, 'qT'], ' ': ['c', N, 'qT']}

}

memory = list(input())

q = 'q0'

index = -1

states = [q]

while q != 'qT':

symb = memory[index]

symbol, i, q = table[q][symb]

memory[index] = symbol

index += i

states.append(q)

print(\*memory, sep='')