

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

Студентка гр. 3344

Якимова Ю.А.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Получить навык составления таблиц для машины Тьюринга, научиться работать с машиной Тьюринга.

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.

Выполнение работы

Перед началом работы машины Тьюринга была считана входная строка, разбита на список и занесена в переменную *s*. Также было задано состояние *state="q1"*, начальный индекс *index=0*. Был создан словарь состояний *states*. Ключами в словаре являются состояния, а значениями - словари с действиями состояния для каждого символа алфавита.

Состояния:

q1 – нахождение начала строки, начальное состояние

q2 – нахождение конца строки

q3 – нахождение первого с конца символа а

q4 – нахождение повтора символов а

q5 – считывание символа слева от повтора

q6 – замена символа справа от “аа” на символ слева, если символ слева - “а”

q7 – замена символа справа от “аа” на символ слева, если символ слева - “б”

q8 – замена символа справа от “аа” на символ слева, если символ слева - “с”

qT – конечное состояние

Был реализован цикл *while*, который работал, пока состояние не равнялось конечному. Были инициализированы переменные *symbol* - символ, на который нужно заменить текущий; *step* - шаг, на который должен увеличиться или уменьшиться индекс ленты; *state* - состояние, в которое перейдёт машина Тьюринга. Значения этим переменным были присвоены согласно правилам словаря *states*. На каждой итерации цикла проверялось: если состояние равнялось 4, то обновлялся *symbol* на значение слева от повтора, если состояние равнялось 5, то происходило обновление символа справа от повтора на символ слева от повтора. После цикла отредактированный список выводился на экран с помощью *print(“”.join(s))*. Таблица состояний представлена в табл. 1

Таблица 1 — Таблица состояний

	“a”	“b”	“c”	“ “
q1	“a”, 1, q2	“b”, 1, q2	“c”, 1, q2	“ “, 1, q1
q2	“a”, 1, q2	“b”, 1, q2	“c”, 1, q2	“ “, -1, q3
q3	“a”, -1, q4	“b”, -1, q3	“c”, -1, q3	
q4	“a”, -1, q5	“b”, -1, q3	“c”, -1, q3	
q5	“a”, 1, q6	“b”, 1, q7	“c”, 1, q8	
q6	“a”, 1, q6	“a”, 0, qT	“a”, 0, qT	“a”, 0, qT
q7	“a”, 1, q7	“b”, 0, qT	“b”, 0, qT	“b”, 0, qT
q8	“a”, 1, q8	“c”, 0, qT	“c”, 0, qT	“c”, 0, qT

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	abcaabc	abcaacc	-
2.	abcabcbaa	abcabcbaab	-

Выводы

Был получен навык составления таблиц для машины Тьюринга. Было получено знание о работе машины Тьюринга.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Yakimova_Yuliya_lb3.py

```
s = list(input())
state = "q1"
index = 0

states = {
    "q1" : {
        "a" : ["a", 1, "q2"],
        "b" : ["b", 1, "q2"],
        "c" : ["c", 1, "q2"],
        " " : [" ", 1, "q1"]
    },
    "q2" : {
        "a" : ["a", 1, "q2"],
        "b" : ["b", 1, "q2"],
        "c" : ["c", 1, "q2"],
        " " : [" ", -1, "q3"]
    },
    "q3" : {
        "a" : ["a", -1, "q4"],
        "b" : ["b", -1, "q3"],
        "c" : ["c", -1, "q3"]
    },
    "q4" : {
        "a" : ["a", -1, "q5"],
        "b" : ["b", -1, "q3"],
        "c" : ["c", -1, "q3"]
    },
    "q5" : {
        "a" : ["a", 1, "q6"],
        "b" : ["b", 1, "q7"],
        "c" : ["c", 1, "q8"]
    },
    "q6" : {
        "a" : ["a", 1, "q6"],
        "b" : ["a", 0, "qT"],
        "c" : ["a", 0, "qT"],
        " " : ["a", 0, "qT"]
    },
    "q7" : {
        "a" : ["a", 1, "q7"],
        "b" : ["b", 0, "qT"],
        "c" : ["b", 0, "qT"],
        " " : ["b", 0, "qT"]
    },
    "q8" : {
        "a" : ["a", 1, "q8"],
        "b" : ["c", 0, "qT"],
        "c" : ["c", 0, "qT"],
        " " : ["c", 0, "qT"]
    }
}
```

```
}

while state != "qT":
    symbol, step, state = states[state][s[index]]
    s[index]=symbol
    index += step

print("".join(s))
```