

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

Кузнецов Р.А.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Изучение и реализация базовых принципов работы машины Тьюринга на языке Python.

Задание.

Вариант 3

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, 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: 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.

Выполнение работы

Для работы был создан словарь состояний, который задает поведение машины Тьюринга. Состояния являются ключами к значениям словарей поменьше. Состояние `q_start` ищет первый символ «а» в ленте. При встрече же других символов, курсор передвигается дальше и остается в состоянии `q_start`. Когда символ найден, то машина переходит в состояние `q_after_a`. Это состояние проверяет какой символ находится после «а» и переходит в состояние соответствующее символу. Курсор двигается в начало ленты(`q1.a`, `q1.b`, `q1.c`). Затем курсор двигается, пока не встретится символ «с» (`q2.a`, `q2.b`, `q2.c`). В зависимости от символа, перед «с» ставится соответствующий символ(`q3.a`, `q3.b`, `q3.c`) и машина переходит в состояние `q_end`. Если же символ после «а» это пробел, то машина переходит в состояние `q_void`. Дальше курсор переносится в начало строки, ищется символ «с» и перед ним вставляется символ «а». Машина переходит в состояние `q_end`. Основное тело программы: считывается строка и приводится к списку. Задается начальное состояние и позиция курсора. Выполняется цикл пока состояние не будет `q_end`. В переменные считываются соответствующее значение. На изначальное место символа, ставится символ указанный в словаре. Затем проверяется движение курсора и добавляется к переменной `position`. Потом машина переходит в другое состояние или же остается в текущем. Выводится полученная лента.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	abcabc	abcabc	-
2.	cbbaa	acbbaa	-
3.	bbcbbcaa	bacbbcaa	-

Выводы

Были изучены базовые принципы работы машины Тьюринга. Была реализована машина Тьюринга на языке Python.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb3.py

```
tape = list(input())

machine = {
    #ищем а
    "q_start": {"a": ("a", "R", "q_after_a"), "b": ("b", "R", "q_start"),
    "c": ("c", "R", "q_start"), " ": (" ", "R", "q_start")},
    # символ после а
    "q_after_a": {"a": ("a", "N", "q_a.1"), "b": ("b", "N", "q_b.1"), "c":
    ("c", "N", "q_c.1"), " ": (" ", "L", "q_void")},
    #вставляем а
    "q_a.1": {"a": ("a", "L", "q_a.1"), "b": ("b", "L", "q_a.1"), "c":
    ("c", "L", "q_a.1"), " ": (" ", "R", "q_a.2")},
    "q_a.2": {"a": ("a", "R", "q_a.2"), "b": ("b", "R", "q_a.2"), "c":
    ("c", "L", "q_a.3")},
    "q_a.3": {"a": ("a", "N", "q_end"), "b": ("a", "N", "q_end"), "c":
    ("a", "N", "q_end"), " ": ("a", "N", "q_end")},
    #вставляем б
    "q_b.1": {"a": ("a", "L", "q_b.1"), "b": ("b", "L", "q_b.1"), "c":
    ("c", "L", "q_b.1"), " ": (" ", "R", "q_b.2")},
    "q_b.2": {"a": ("a", "R", "q_b.2"), "b": ("b", "R", "q_b.2"), "c":
    ("c", "L", "q_b.3")},
    "q_b.3": {"a": ("b", "N", "q_end"), "b": ("b", "N", "q_end"), "c":
    ("b", "N", "q_end"), " ": ("b", "N", "q_end")},
    #вставляем с
    "q_c.1": {"a": ("a", "L", "q_c.1"), "b": ("b", "L", "q_c.1"), "c":
    ("c", "L", "q_c.1"), " ": (" ", "R", "q_c.2")},
    "q_c.2": {"a": ("a", "R", "q_c.2"), "b": ("b", "R", "q_c.2"), "c":
    ("c", "L", "q_c.3")},
    "q_c.3": {"a": ("c", "N", "q_end"), "b": ("c", "N", "q_end"), "c":
    ("c", "N", "q_end"), " ": ("c", "N", "q_end")},
    #если пробел
    "q_void": {"a": ("a", "L", "q_void"), "b": ("b", "L", "q_void"), "c":
    ("c", "L", "q_void"), " ": (" ", "R", "q_void.2")},
    "q_void.2": {"a": ("a", "R", "q_void.2"), "b": ("b", "R", "q_void.2"),
    "c": ("c", "L", "q_void.3")},
    "q_void.3": {"a": ("a", "N", "q_end"), "b": ("a", "N", "q_end"), "c":
    ("a", "N", "q_end"), " ": ("a", "N", "q_end")}
}

state = "q_start"
position = 0

while state != "q_end":
    insert_symbol, move, next_state = machine[state][tape[position]]
    tape[position] = insert_symbol

    position += 1 if move == "R" else -1 if move == "L" else 0
    state = next_state
print("".join(tape))
```