

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Информатика»
Тема: Моделирование работы Машины Тьюринга

Студент гр. 0382

Крючков А.М.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

Цель работы.

Смоделировать работу машины Тьюринга на языке Python

Задание.

На вход программе подается строка неизвестной длины. Каждый элемент является значением в ячейке памяти ленты Машины Тьюринга. На ленте находится троичное число, знак (плюс или минус) и троичная цифра.

Напишите программу, которая выполнит арифметическую операцию. Указатель на текущее состояние Машины Тьюринга изначально находится слева от числа (но не на первом его символе). По обе стороны от числа находятся пробелы. Результат арифметической операции запишите на месте первого числа. Ваша программа должна вывести полученную ленту после завершения работы.

Основные теоретические положения.

Машина Тьюринга (МТ) состоит из двух частей: неподвижной бесконечной ленты (памяти) и автомата (процессора).

1) Лента используется для хранения информации. Она бесконечна в обе стороны и разбита на клетки, которые никак не нумеруются и не именуются. В каждой клетке может быть записан один символ или ничего не записано. Память пассивна: она ничего не делает, просто хранит данные.

2) Алфавит ленты - конечное множество всех возможных символов ленты $\{0, 1, 2, +, -, '\}$.

3) Автомат – это активная часть Машины Тьюринга. В каждый момент он размещается под одной из клеток ленты и видит её содержимое; это видимая клетка, а находящийся в ней символ – видимый символ; содержимое же 3 соседних и других клеток автомат не видит. Кроме того, в каждый момент автомат находится в одном из состояний, которые обычно обозначаются буквой q с номерами: q_0, q_1, q_2 и т.д. Существует конечное число таких состояний.

В каждом из состояний автомат выполняет какую-то конкретную операцию. Существует заключительное состояние, в котором автомат останавливается.

Автомат за один такт (шаг) может выполнить следующие действия:

1. считать видимый символ;
2. записывать в видимую клетку новый символ (в том числе пустой символ);
3. сдвигаться на одну клетку влево или вправо («перепрыгивать» сразу через несколько клеток автомат не может);
4. перейти в следующее состояние.

Выполнение работы.

Составим таблицу состояний (таблица 1).

Таблица 1 — Таблица состояний

	0	1	2	+	-	“ “
q0	0;1;q1	1;1;q1	2;1;q1	+;1;q1	-;1;q1	„1“;1q0
q1	0; 1; q1	1; 1; q1	2; 1; q1	+; 1; q1	-; 1; q1	“ “; -1; q2
q2	0; -1; q3	1; -1; q4	2; -1; q5			
q3	0; 0; q3	1; 0; q3	2; 0; q3	+; 0; q3	-; 0; q3	“ “; 0; q3
q4				+;-1;q6	-;-1;q7	
q5				+;-1;q8	-;-1;q9	
q6	1; 0; q10	2; 0; q10	0; -1; q6			1;1;q10
q7	2; -1; q7	1; 0; q10	1; 0; q10			
q8	2; 0; q3	1; -1; q6	1; -1; q6			
q9	1; -1; q7	2; -1; q7	0; 0; q10			
q10	0;-1;q10	1;-1;q10	2;-1;q10	+;-1;q10	-;-1;q10	„ „; 1;q11
q11	„ „;1;q11	1;1;q3	2;1;q3	+;-1;q12	-;-1;q12	

q1						0;0;q3
2						

Состояния

q0 – состояние, для перехода к первой цифре выражения.

q1 – состояние, для перехода к последней цифре выражения.

q2 – состояние, для определения, какую цифру надо прибавить или вычесть.

q3 – состояние, в котором автомат останавливается.

q4 – состояние, в котором определяется надо вычитать или прибавлять единицу.

q5 – состояние, в котором определяется надо вычитать или прибавлять единицу двойку.

q6 – состояние, в котором прибавляется единица.

q7 – состояние, в котором вычитается единица.

q8 – состояние, в котором прибавляется двойка.

q9 – состояние, в котором вычитается двойка.

q10 – состояние, в котором автомат переходит к началу выражения.

q11 – состояние, в котором автомат удаляет незначащие нули.

q12 — состояние, в котором автомат добавляет значащий ноль, если полученное число 0.

Описание используемых переменных

table — словарь, содержит в себе таблицу состояний

position — число, позиция автомата на ленте

state — строка, текущее состояние

tape — список, содержащий ленту

sym — символ, который запишет автомат

move – шаг автомата

Тестирование.

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	111+1	112+1	Верно
2.	111-1	110-1	Верно
3.	22+1	100+1	Верно

Выводы.

В ходе работы был изучен принцип работы машины Тьюринга и, в соответствии с заданием, была смоделирована работа машины Тьюринга на языке Python.

Разработана программа, в которой таблица состояний хранится в двумерном словаре, а обработка входных данных происходит при помощи цикла while.

Программа выполняет сложение или вычитание троичного числа (слева от знака) и троичной цифры (справа от знака), изменяет ленту памяти и выводит изменённую ленту на экран.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb2.py

```
table = {
    "q0": {"0": ["0", 1, "q1"], "1": ["1", 1, "q1"], "2": ["2", 1,
"q1"], "+" : ["+", 1, "q1"] , "-" : ["-", 1, "q1"], " " : [" ", 1,
"q0"]},
    "q1": {"0": ["0", 1, "q1"], "1": ["1", 1, "q1"], "2": ["2", 1,
"q1"], "+" : ["+", 1, "q1"] , "-" : ["-", 1, "q1"], " " : [" ", -1,
"q2"]},
    "q2": {"0": ["0", -1, "q10"], "1": ["1", -1, "q4"], "2": ["2", -
1, "q5"]},
    "q3": {"0": ["0", 0, "q3"], "1": ["1", 0, "q3"], "2": ["2", 0,
"q3"], "+" : ["+", 0, "q3"] , "-" : ["-", 0, "q3"], " " : [" ", 0,
"q3"]},
    "q4": {"+" : ["+", -1, "q6"] , "-" : ["-", -1, "q7"]},
    "q5": {"+" : ["+", -1, "q8"] , "-" : ["-", -1, "q9"]},
    "q6": {"0": ["1", 0, "q10"], "1": ["2", 0, "q10"], "2": ["0", -
1, "q6"], " " : ["1", 1, "q10"]}, # +1
    "q7": {"0": ["2", -1, "q7"], "1": ["0", 0, "q10"], "2": ["1", 0,
"q10"]}, # -1
    "q8": {"0": ["2", 0, "q10"], "1": ["0", -1, "q6"], "2": ["1", -
1, "q6"]}, # +2
    "q9": {"0": ["1", -1, "q7"], "1": ["2", -1, "q7"], "2": ["0", 0,
"q10"]}, # -2
    "q10": {"0": ["0", -1, "q10"], "1": ["1", -1, "q10"], "2": ["2",
-1, "q10"], "+" : ["+", -1, "q10"] , "-" : ["-", -1, "q10"], " " : [" ",
1, "q11"]},
    "q11": {"0": [" ", 1, "q11"], "1": ["1", 1, "q3"], "2": ["2", 1,
"q3"], "+" : ["+", -1, "q12"] , "-" : ["-", -1, "q12"]},
    "q12": {" " : ["0", 0, "q3"]}
}

position = 0
state = "q0"
tape = list(input())

while state!="q3":
    sym, move, state = table[state][tape[position]]
    tape[position] = sym
    position += move

print("".join(tape))
```