

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

Мукатанов А.В.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

Цель работы.

Смоделировать работу машины Тьюринга на языке программирования 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.

		1	2	1	+	2			
--	--	---	---	---	---	---	--	--	--

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. Для примера выше лента будет выглядеть так:

		2	0	0	+	2			
--	--	---	---	---	---	---	--	--	--

Ваша программа должна вывести полученную ленту после завершения работы.

Алфавит:

- 0
- 1
- 2
- +
- -
- " " (пробел)

Соглашения:

1. Направление движения автомата может быть одно из R (направо), L (налево), N (неподвижно).
2. Число обязательно начинается с единицы или двойки.
3. Числа и знак операции между ними идут непрерывно.
4. Гарантируется, что в результате операции вычитания не может получиться отрицательного числа.

В отчет включите таблицу состояний. Отдельно кратко опишите каждое состояние, например:

q1 - начальное состояние, которое необходимо, чтобы найти первую цифру первого числа.

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. Если предположить, что видимые символы - весь алфавит ленты из примера выше, то мы имеем следующий алфавит: {1, 0, +, 'a', ""}. Последний символ - пустой, означает пустое содержимое клетки.

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 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: q0, q1, q2 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 :

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

4. input() – считывает вводимые данные Printf() – выводит данные Цикл while – выполняет указания , пока верно условие.

Выполнение работы.

Выполнение работы смотри в табл. 1.

Таблица 1 – Выполнение работы.

	“0”	“1”	“2”	“+”	“-”	“ ”
Q0	'0', 1, 'q1'	'1', 1, 'q1'	'2', 1, 'q1'	'+', 1, 'q1'	'-', 1, 'q1'	' ', 1, 'q0'
Q1	'0', 1, 'q1'	'1', 1, 'q1'	'2', 1, 'q1'	'+', 1, 'q2'	'-', 1, 'q5'	
Q2	'0', -1, 'qA'	'1', -1, 'q3'	'2', -1, 'q4'			
Q3	'1', 0, 'qA'	'2', 0, 'qA'	'0', -1, 'q3'	'+', -1, 'q3'		'1', 0, 'qA'
Q4	'2', 0, 'qA'	'0', -1, 'q3'	'1', -1, 'q3'	'+', -1, 'q4'		

Q5	'0', -1, 'qA'	'1', -1, 'q6'	'2', -1, 'q7'			
Q6	'2', -1, 'q6'	'0', -1, 'q8'	'1', 0, 'qA'		'-', -1, 'q6'	
Q7	'1', -1, 'q6'	'2', -1, 'q6'	'0', -1, 'q8'		'-', -1, 'q7'	
Q8	'0', -1, 'q8'	'1', -1, 'q8'	'2', -1, 'q8'			' ', 1, 'q9'
Q9	' ', 1, 'q9'	'1', 0, 'qA'	'2', 0, 'qA'	'+', -1, 'q10'	'-', -1, 'q10'	
Q10						'0', 0, 'qA'

Q0 – начальное состояние

Q1 – автомат обрабатывает то ,какую операцию ему нужно выполнить и переходит к состоянию q2 или q5

Q2 - автомат обрабатывает то ,какую цифру ему нужно прибавить и переходит к q3 или q4

Q3 - +1

Q4 - +2

Q5 - автомат обрабатывает то ,какую цифру ему нужно вычесть и переходит к q6 или q7

Q6 - -1

Q7- -2

Q8 – переход к самому левому знаку

Q9 – удаление незначащих нулей ,если такие есть то переход к q10

Q10 – замена незначащих нулей пробелом

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 2.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
-------	----------------	-----------------	-------------

1.	111+1	112+1	Программа работает корректно
2.	1-1	0-1	Программа работает корректно

Выводы.

Был изучен принцип работы машины Тьюринга. Этот алгоритм был использован для решения поставленной задачи. Разработана программа, которая создаёт таблицу состояний для машины Тьюринга, считывает исходную строку (ленту), проводит соответствующие таблице операции (сложение или вычитание троичного числа и цифры) и выводит полученную строку.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.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, 'q2'],
                '-': ['-', 1, 'q5'],
                },
         'q2': {'0': ['0', -1, 'qA'],
                '1': ['1', -1, 'q3'],
                '2': ['2', -1, 'q4'],
                },
         'q3': {'0': ['1', 0, 'qA'],
                '1': ['2', 0, 'qA'],
                '2': ['0', -1, 'q3'],
                '+': ['+', -1, 'q3'],
                ' ': ['1', 0, 'qA'],
                },
         'q4': {'0': ['2', 0, 'qA'],
                '1': ['0', -1, 'q3'],
                '2': ['1', -1, 'q3'],
                '+': ['+', -1, 'q4'],
```

```

    },
    'q5': {'0': ['0', -1, 'qA'],
           '1': ['1', -1, 'q6'],
           '2': ['2', -1, 'q7'],
           },
    'q6': {'0': ['2', -1, 'q6'],
           '1': ['0', -1, 'q8'],
           '2': ['1', 0, 'qA'],
           '-': ['-', -1, 'q6'],
           },
    'q7': {'0': ['1', -1, 'q6'],
           '1': ['2', -1, 'q6'],
           '2': ['0', -1, 'q8'],
           '-': ['-', -1, 'q7'],
           },
    'q8': {'0': ['0', -1, 'q8'],
           '1': ['1', -1, 'q8'],
           '2': ['2', -1, 'q8'],
           ' ': [' ', 1, 'q9']},
    'q9': {'0': [' ', 1, 'q9'],
           '1': ['1', 0, 'qA'],
           '2': ['2', 0, 'qA'],
           '+': ['+', -1, 'q10'],
           '-': ['-', -1, 'q10']
           },
    'q10': {' ': ['0', 0, 'qA']
            }
    }

tape = list(input())
state = 'q0'

```



```
i = 0
while state != 'qA':
    sym, step, state = table[state][tape[i]]
    tape[i] = sym
    i += step
print("".join(tape))
```