

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

Студентка гр. 0382

Здобнова К.Д.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

Цель работы.

Изучить работу машины Тьюринга. Научиться описывать модель работы ленты 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.

| | | | | | | | | | |
|--|--|---|---|---|---|---|--|--|--|
| | | 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.

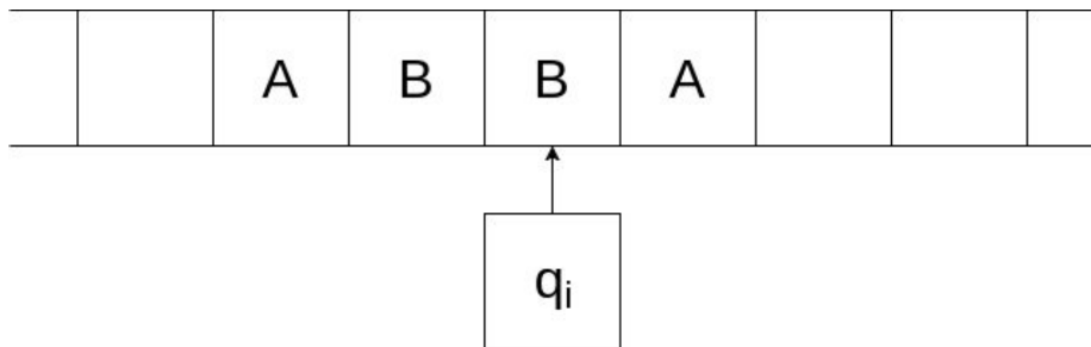
3. Числа и знак операции между ними идут непрерывно.

4. Гарантируется, что в результате операции вычитания не может получиться отрицательного числа.

Основные теоретические положения.

Машина Тьюринга (МТ) состоит из двух частей:

- Неподвижной бесконечной ленты (памяти);
- Автомата (процессора).



Автомат (процессор) – это активная часть Машины Тьюринга. В каждый момент он размещается под одной из клеток ленты и может прочитать ее содержимое; содержимое других клеток автомат не видит.

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 некую операцию. Существует заключительное состояние, в котором автомат останавливается.

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

- Считать видимый символ;
- Записывать в видимую клетку новый символ (в том числе пустой символ);
- Сдвигать на одну клетку влево или вправо;
- Перейти в следующее состояние.

Выполнение работы.

Вводные данные записываются в список *memory*. Таблица состояний (табл.1) хранится в словаре *table*. Переменная *state* указывает на то, в каком состоянии в данный момент находится лента, *index* – на какой элемент списка указывает автомат.

Данные ленты обрабатываются в цикле *while* – пока состояние не станет терминальным. В переменную *sum* записывается символ, который нужно поместить в клетку, на которой находится автомат, переменная *delta* отвечает за направление шага – влево (-1).

Таблица 1 – таблица состояний машины

| | 0 | 1 | 2 | + | - | ' ' |
|----|------------|----------|----------|------------|------------|------------|
| q0 | 0, R, q1 | 1, R, q1 | 2, R, q1 | | | ' ', R, q0 |
| q1 | 0, R, q1 | 1, R, q1 | 2, R, q1 | '+', R, q2 | '-', R, q3 | |
| q2 | 0, L, qt | 1, L, q4 | 2, L, q5 | | | |
| q3 | 0, L, qt | 1, L, q6 | 2, L, q7 | | | |
| q4 | 1, N, qt | 2, N, qt | 0, L, q4 | '+', L, q4 | | '1', N, qt |
| q5 | 2, N, qt | 0, L, q4 | 1, L, q4 | '+', L, q5 | | |
| q6 | 2, L, q6 | 0, L, q8 | 1, L, q8 | | '-', L, q6 | |
| q7 | 1, L, q6 | 2, L, q6 | 0, L, q8 | | '-', L, q7 | |
| q8 | '0', N, qt | 1, N, qt | 2, N, qt | | | ' ', R, q9 |
| q9 | ' ', R, q9 | 1, N, qt | 2, N, qt | '+', L, q9 | '-', L, q9 | '0', N, qt |

q0 – начальное состояние, до первого найденного числа;

q1 – первое число;

q2 – состояние, после которого цифра будет прибавлена к числу;

q3 - состояние, после которого цифра будет отнята от числа;

q4 – состояние, для прибавления 1;

q5 – состояние, для вычитания 1;

q6 – состояние, для прибавления 2;

q7 – состояние, для вычитания 2;

q8 – состояние определяющее, есть ли перед нулем число;

q9 – состояние, вычеркивающее незначащий ноль;

qt – конечное состояние.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

| № п/п | Входные данные | Выходные данные | Комментарии |
|-------|----------------|-----------------|---------------------------------|
| 1. | 111+1 | 112+1 | К числу 111 было прибавлен 1 |
| 2. | 1-1 | 0-1 | Из числа 0 вычли 1 |
| 3. | 10001-2 | 2222-2 | Из числа 10001 вычли 2 |
| 4. | 10122+2 | 10201+2 | К числу 10122 было прибавлено 2 |
| 5. | 2+2 | 11+2 | К числу 2 было прибавлено 2 |

Выводы.

Была изучена работа машины Тьюринга. В лабораторной работе были смоделированы лента и автомат для перехода между состояниями.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb2.py

```
memory = list(input())
index = 0
state = 'q0'
table = {'q0': {'0': ['0', 1, 'q1'],
                '1': ['1', 1, 'q1'],
                '2': ['2', 1, 'q1'],
                ' ': [' ', 1, 'q0']}},
        'q1': {'0': ['0', 1, 'q1'],
                '1': ['1', 1, 'q1'],
                '2': ['2', 1, 'q1'],
                '+': ['+', 1, 'q2'],
                '-': ['-', 1, 'q3']}},
        'q2': {'0': ['0', -1, 'qt'],
                '1': ['1', -1, 'q4'],
                '2': ['2', -1, 'q5']}},
        'q3': {'0': ['0', -1, 'qt'],
                '1': ['1', -1, 'q6'],
                '2': ['2', -1, 'q7']}},
        'q4': {'0': ['1', 0, 'qt'],
                '1': ['2', 0, 'qt'],
                '2': ['0', -1, 'q4'],
                '+': ['+', -1, 'q4'],
                ' ': ['1', 0, 'qt']}},
        'q5': {'0': ['2', 0, 'qt'],
                '1': ['0', -1, 'q4'],
                '2': ['1', -1, 'q4'],
                '+': ['+', -1, 'q5']}},
        'q6': {'0': ['2', -1, 'q6'],
                '1': ['0', -1, 'q8'],
                '2': ['1', -1, 'q8'],
                '-': ['-', -1, 'q6']}},
        'q7': {'0': ['1', -1, 'q6'],
                '1': ['2', -1, 'q6'],
                '2': ['0', -1, 'q8'],
                '-': ['-', -1, 'q7']}},
        'q8': {'0': ['0', 0, 'qt'],
                '1': ['1', 0, 'qt'],
                '2': ['2', 0, 'qt'],
                ' ': [' ', 1, 'q9']}},
        'q9': {'0': [' ', 1, 'q9'],
                '1': ['1', 0, 'qt'],
                '2': ['2', 0, 'qt'],
                '+': ['+', -1, 'q9'],
                '-': ['-', -1, 'q9'],
                ' ': ['0', 0, 'qt']}
        }
}
while state != 'qt':
    sym, delta, state = table[state][memory[index]]
    memory[index] = sym
    index += delta
print(''.join(memory))
```