

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

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

Студент гр. 0382

Кондратов Ю.А.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

Цель работы.

Смоделировать работу машины Тьюринга на языке Python.

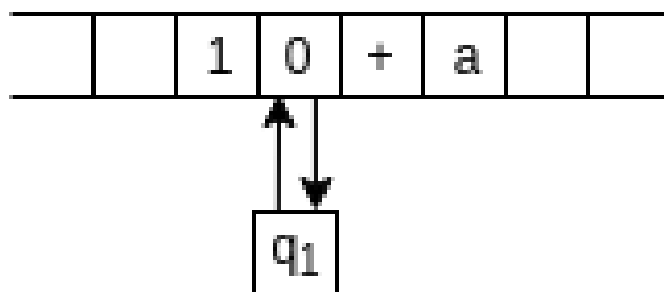
Задание.

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

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» | «+» | «-» | « » |
|----------|-------------------|-------------------|-------------------|----------------------|----------------------|-------------------|
| q_0 | «0», R, « q_1 » | «1», R, « q_1 » | «2», R, « q_1 » | «+», R, « q_1 » | «-», R, « q_1 » | « », R, « q_0 » |
| q_1 | «0», R, « q_1 » | «1», R, « q_1 » | «2», R, « q_1 » | «+», R, « q_2 » | «-», R, « q_5 » | |
| q_2 | «0», L, « q_T » | «1», L, « q_3 » | «2», L, « q_4 » | | | |
| q_3 | «0», N, « q_1 » | «1», N, « q_1 » | «2», L, « q_1 » | «0», L, « q_1 » | | «1», N, « q_T » |
| q_4 | «2», N, « q_T » | «0», L, « q_3 » | «1», L, « q_3 » | «+», L, « q_4 » | | |
| q_5 | «0», L, « q_T » | «1», L, « q_6 » | «2», L, « q_7 » | | | |
| q_6 | «2», L, « q_6 » | «0», L, « q_8 » | «1», N, « q_T » | | «-», L, « q_6 » | |
| q_7 | «1», L, « q_6 » | «2», L, « q_6 » | «0», L, « q_8 » | | «-», L, « q_7 » | |
| q_8 | «0», L, « q_8 » | «1», R, « q_8 » | «2», L, « q_8 » | | | « », R, « q_9 » |
| q_9 | « », R, « q_9 » | «1», N, « q_T » | «2», N, « q_T » | «+», L, « q_{10} » | «-», L, « q_{10} » | |
| q_{10} | | | | | | «0», N, « q_T » |

Описания состояний:

q0 — начальное состояние, автомат находится «в поиске» слова;

q1 — состояние для определения арифметической операции, которую необходимо выполнить;

q2 — состояние для определения цифры, которую необходимо прибавить;

q3 — состояние, в котором автомат прибавляет единицу к числу;

q4 — состояние, в котором автомат прибавляет двойку к числу;

q5 — состояние для определения цифры, которую необходимо вычесть;

q6 — состояние, в котором автомат вычитает единицу;

q7 — состояние, в котором автомат вычитает двойку;

q8 — состояние для перехода к самому левому символу на ленте;

q9 — состояние для удаления незначащих нулей, которые могли появиться при вычитании;

q10 — состояние, предназначенное для замены незначащего нуля пробелом;

- Ввод и инициализация переменных:

- для реализации на Python таблица состояний записывается в двумерный словарь *table*;

- входные данные записываются в переменную *tem*. С помощью функции *list()* входная строка преобразуется в список символов;

- в переменную *state* записано начальное состояние «q0», а переменная *i* выполняет роль индекса.

- Обработка ленты:

Производится при помощи цикла *while* с условием неравенства переменной *state* значению конечного состояния автомата «qT».

В каждой итерации цикла в переменную *sum* сохраняется символ, который автомат должен записать на ленту, в переменную *move* сохраняется

значение, на которое должен измениться индекс (аналог передвижения автомата по ленте), а в переменную *state* сохраняется текущее состояние автомата.

Далее в строке *mem[i] = sym* происходит запись символа на ленту, а в строке *i += move* — изменение индекса, оно же перемещение автомата.

- Вывод:

Изменённая лента выводится на консоль при помощи функции *print()* и метода *join()*

Тестирование.

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

| № п/п | Входные данные | Выходные данные | Комментарии |
|-------|----------------|-----------------|------------------------------|
| 1. | 210-1 | 202-1 | Программа работает правильно |
| 2. | 10000-2 | 2221-2 | Программа работает правильно |
| 3. | 1-1 | 0-1 | Программа работает правильно |
| 4. | 1001+2 | 1010+2 | Программа работает правильно |
| 5. | 2-2 | 0-2 | Программа работает правильно |
| 6. | 101-2 | 22-2 | Программа работает правильно |
| 7. | 111+1 | 112+1 | Программа работает правильно |

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: 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, 'qT'],
                '1': ['1', -1, 'q3'],
                '2': ['2', -1, 'q4']},
         'q3': {'0': ['1', 0, 'qT'],
                '1': ['2', 0, 'qT'],
                '2': ['0', -1, 'q3'],
                '+': ['+', -1, 'q3'],
                ' ': ['1', 0, 'qT']},
         'q4': {'0': ['2', 0, 'qT'],
                '1': ['0', -1, 'q3'],
                '2': ['1', -1, 'q3'],
                '+': ['+', -1, 'q4']},
         'q5': {'0': ['0', -1, 'qT'],
                '1': ['1', -1, 'q6'],
                '2': ['2', -1, 'q7']},
         'q6': {'0': ['2', -1, 'q6'],
                '1': ['0', -1, 'q8'],
                '2': ['1', 0, 'qT'],
                '-': ['-', -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, 'qT'],
                '2': ['2', 0, 'qT'],
                '+': ['+', -1, 'q10'],
                '-': ['-', -1, 'q10']},
         'q10': {' ': ['0', 0, 'qT']}}
}
```

```
mem = list(input())

state = 'q0'
i = 0

while state != 'qT':
    sym, move, state = table[state][mem[i]]
    mem[i] = sym
    i += move

print(''.join(mem))
```