

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. 0383

Шквиря Е.В.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

Цель работы.

Путём моделирования Машины Тьюринга на языке Python изучить её устройство работы.

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.

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.

Выполнение работы.

Порядок выполнения поставленной задачи:

1. Была составлена таблица состояний, между которыми будет переходить машина Тьюринга. Каждое состояние содержит в себе инструкции для разрешения возможных ситуаций, возникающих при работе программы.
2. Таблица была перенесена на язык программирования Python в форме словаря. Обращение происходит путём поиска у соответствующего состояния инструкцию под конкретный символ.
3. После получения инструкции для обработки символа происходит изменение ленты путём замены символа на другой или тот же(*symb*),

смещением в одну из сторон или сохранения своего расположения(*step*) и переходом в следующее состояние(*chState*).

4. Ответ достигается следующим путём: ищем знак сложения или вычитания; смотрим, какая после него цифра; прибавляем эту цифру к числу или отнимаем её от числа; идём в начало числа, затем в конец, попутно убирая все незначащие нули; если мы дошли до знака + или -, то возвращаемся на прошлую позицию и ставим один ноль(результат арифметической операции равен нулю).

Таблица состояний:

	0	1	2	+	-	" "
q1	0,R,q1	1,R,q1	1,R,q1	+,R,q2	-,R,q3	" ",R,q1
q2	0,L,q4	1,L,q5	2,L,q6			
q3	0,L,q7	1,L,q8	2,L,q9			
q4	0,N,q10	1,N,q10	2,N,q10	+,L,q4	-,L,q4	
q5	1,N,q10	2,N,q10	0,L,q5	+,L,q5	-,L,q5	1,N,q10
q6	2,N,q10	0,L,q5	1,L,q5	+,L,q6	-,L,q6	
q7	0,N,q10	1,N,q10	2,N,q10	+,L,q7	-,L,q7	
q8	2,L,q8	0,N,q10	1,N,q10	+,L,q8	-,L,q8	
q9	1,L,q8	2,L,q8	0,N,q10	+,L,q9	-,L,q9	
q10	0,L,q10	1,L,q10	2,L,q10	+,L,q10	-,L,q10	" ",R,q11
q11	" ",R,q11	1,N,qT	2,N,qT	+,L,q12	-,L,q12	" ",N,qT
q12						0,N,qT

Расшифровка состояний:

q1 — поиск знака.

q2 — если знак +, то смотрим следующую цифру. Далее, в зависимости от цифры, переходим к следующему состоянию.

q3 — если знак -, то смотрим следующую цифру. Далее, в зависимости от цифры, переходим к следующему состоянию.

q4 — складываем число с 0 на соответствующем разряде.

q5 — складываем число с 1 на соответствующем разряде.

q6 — складываем число с 2 на соответствующем разряде.

q7 — вычитаем из числа 0 на соответствующем разряде.

q8 — вычитаем из числа 1 на соответствующем разряде.

q9 — вычитаем из числа 2 на соответствующем разряде.

q10 — после выполнения арифметической операции переходим в левую часть числа.

q11 — убираем незначащие 0 до первого знака 1, 2, + или -. Если встретилась цифра, то завершаем работу. Если встретился знак, то смещаемся назад на один разряд.

q12 — ставим вместо пробела 0, так как результат арифметической операции равен 0.

qT — конечное состояние.

Используемые в программе переменные:

tableOfStates — таблица состояний.

lenta — лента, на которой расположены символы.

state — состояние, в котором сейчас находится машина Тьюринга.

endState — конечное состояние, на котором машина завершает свою работу.

ind — позиция на ленте, которую машина в данный момент обрабатывает.

symb — символ, на который мы заменяем текущий в ленте.

step — шаг, на который мы смещаем позицию после обработки.

chState — состояние, к которому машина переходит после обработки.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	1-1	0-1	Верный ответ
2.	221-0	221-0	Верный ответ
3.	101-2	22-2	Верный ответ
4.	220-2	211-2	Верный ответ
5.	11+2	20+2	Верный ответ
6,	100+0	100+0	Верный ответ
7,	222+1	1000+1	Верный ответ
8,	0+2	2+2	Верный ответ

Выводы.

Были изучено устройство машины Тьюринга путём моделирования её работы на языке Python.

Разработана программа, получающая на вход ленту с троичным числом, арифметическим знаком и троичной цифрой, и в дальнейшем совершающая указанную операцию над числом и цифрой. Для выполнения задачи использовалась таблица состояний, реализованная с помощью словаря.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
tableOfStates = {
    'q1': {'0': ['0', 1, 'q1'], '1': ['1', 1, 'q1'], '2': ['2', 1, 'q1'], '+': ['+', 1, 'q2'], '-':
['-', 1, 'q3'],
        ' ': [' ', 1, 'q1']},
    'q2': {'0': ['0', -1, 'q4'], '1': ['1', -1, 'q5'], '2': ['2', -1, 'q6']},
    'q3': {'0': ['0', -1, 'q7'], '1': ['1', -1, 'q8'], '2': ['2', -1, 'q9']},
    'q4': {'0': ['0', 0, 'q10'], '1': ['1', 0, 'q10'], '2': ['2', 0, 'q10'], '+': ['+', -1,
'q4'],
        '-': ['-', -1, 'q4']},
    'q5': {'0': ['1', 0, 'q10'], '1': ['2', 0, 'q10'], '2': ['0', -1, 'q5'], '+': ['+', -1,
'q5'],
        '-': ['-', -1, 'q5'], ' ': ['1', 0, 'q10']},
    'q6': {'0': ['2', 0, 'q10'], '1': ['0', -1, 'q5'], '2': ['1', -1, 'q5'], '+': ['+', -1, 'q6'],
        '-': ['-', -1, 'q6']},
    'q7': {'0': ['0', 0, 'q10'], '1': ['1', 0, 'q10'], '2': ['2', 0, 'q10'], '+': ['+', -1,
'q7'],
        '-': ['-', -1, 'q7']},
    'q8': {'0': ['2', -1, 'q8'], '1': ['0', 0, 'q10'], '2': ['1', 0, 'q10'], '+': ['+', -1,
'q8'],
        '-': ['-', -1, 'q8']},
    'q9': {'0': ['1', -1, 'q8'], '1': ['2', -1, 'q8'], '2': ['0', 0, 'q10'], '+': ['+', -1, 'q9'],
        '-': ['-', -1, 'q9']},
    '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', 0, 'qT'], '2': ['2', 0, 'qT'], '+': ['+', -1,
'q12'],
        '-': ['-', -1, 'q12'], ' ': [' ', 0, 'qT']},
    'q12': {' ': ['0', 0, 'qT']}
```

```
}
```

```
lenta = list(input())
```

```
state = 'q1'
```

```
endState = 'qT'
```

```
ind = 0
```

```
while state != endState:
```

```
    symb, step, chState = tableOfStates[state][lenta[ind]]
```

```
    lenta[ind] = symb
```

```
    ind += step
```

```
    state = chState
```

```
print(*lenta, sep="")
```