

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное
учреждение высшего образования

**Национальный исследовательский университет
«МЭИ»**

Институт ИВТИ

Кафедра ПМИИ

Теоритические модели вычислений

ДЗ №3: Машины Тьюринга и квантовые вычисления

Выполнил: студент группы А-13а-19

Тулинов А.В.

Преподаватель: Ивлиев С.А.

Москва, 2022 г.

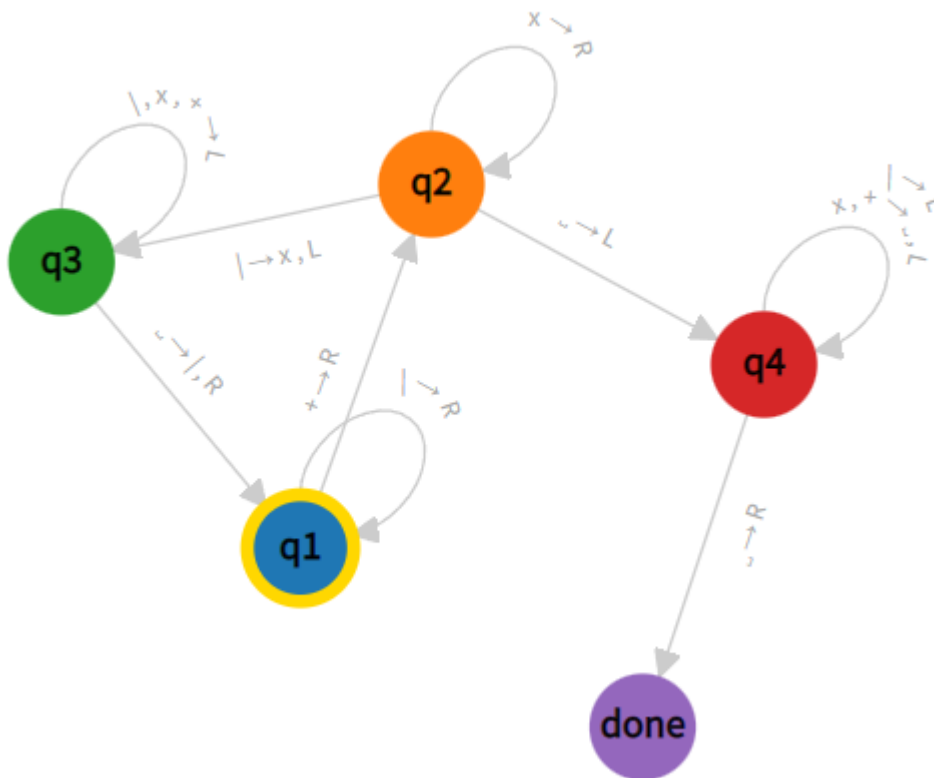
2 Машины Тьюринга

2.1 Операции с числами

Реализуйте машины Тьюринга, которые позволяют выполнять следующие операции:

1. Сложение двух унарных чисел (1 балл)

Решение:



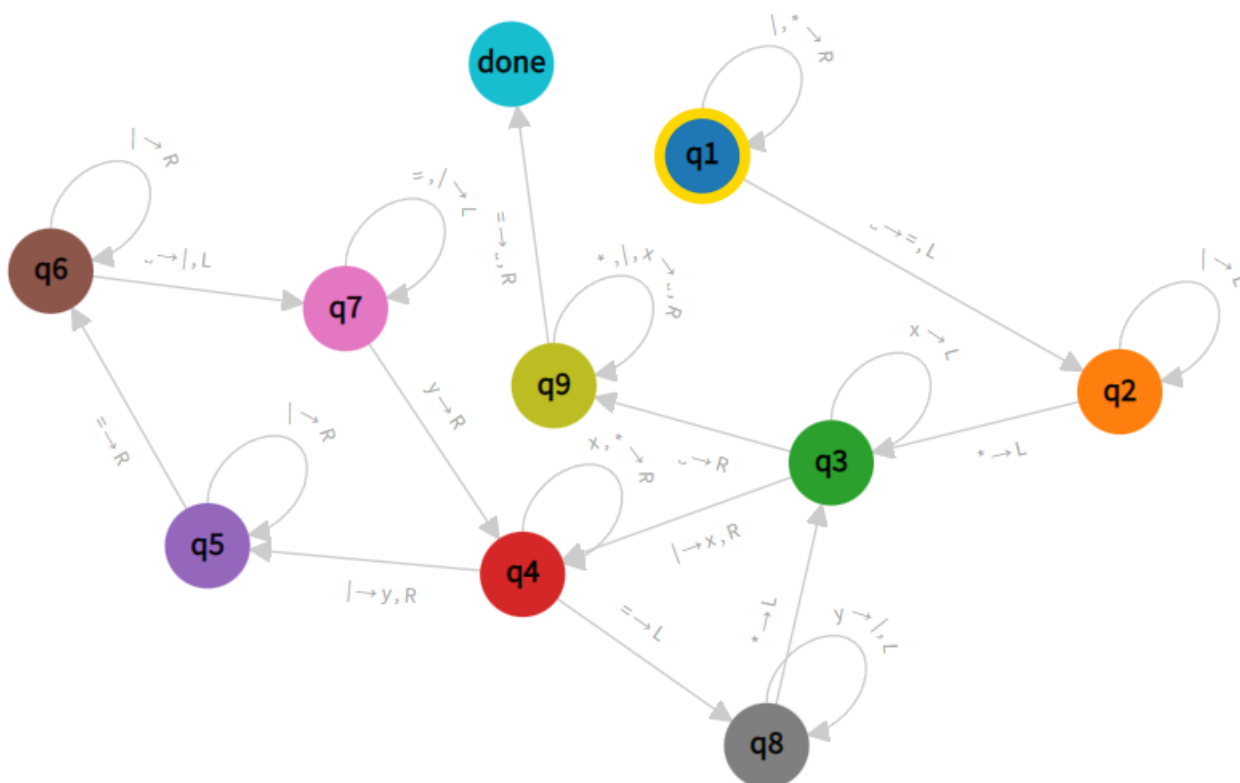
Листинг:

```
1  input: '||+|||'  
2  blank: ' '  
3  start state: q1  
4  table:  
5    q1:  
6      "|": R  
7      '+': {R: q2}  
8    q2:  
9      "|": {write: x, L: q3}  
10     x: R  
11     ' ': {L: q4}  
12    q3:  
13     ["|", "x", '+']: L  
14     ' ': {write: "|", R: q1}  
15    q4:  
16     ["x", '+']: {write: ' ', L}  
17     "|": L  
18     ' ': {R: done}  
19  done:
```

Описание **.yaml** к соответствующей задаче находится в папке MT.

2. Умножение унарных чисел (1 балл)

Решение:



Листинг:

```
1  input: '||*||'
2  blank: ' '
3  start state: q1
4  table:
5    q1:
6      ["|","*"] : R
7      ' ' : {write: '=', L: q2}
8    q2:
9      "|" : L
10     "*" : {L: q3}
11    q3:
12      x: L
13      "|": {write: x, R: q4}
14      ' ': {R: q9}
15    q4:
16      ["x","*"] : R
17      "|": {write: y, R: q5}
18      "=" : {L: q8}
19    q5:
20      "|": R
21      "=" : {R: q6}
22    q6:
23      ' ' : {write: "|",L: q7}
24      "|": R
25    q7:
26      ["=","|"] : L
27      y : {R: q4}
28    q8:
29      y: {write: "|", L}
30      "*" : {L: q3}
31    q9:
32      ["*","|","x"] : {write: ' ',R}
33      "=" : {write: ' ',R: done}
34    done:
```

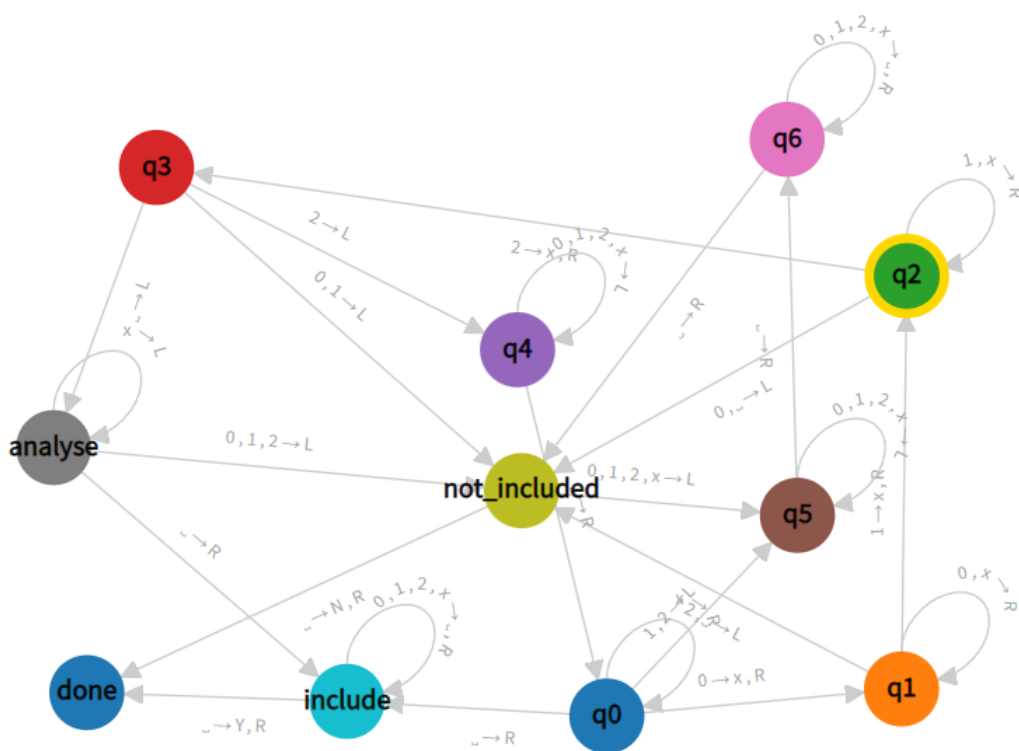
Описание **.yaml** к соответствующей задаче находится в папке МТ.

2.2 Операции с языками и символами

Реализуйте машины Тьюринга, которые позволяют выполнять следующие операции:

1. Принадлежность к языку $L = \{0^n 1^n 2^n\}, n \geq 0$ (0.5 балла)

Решение:



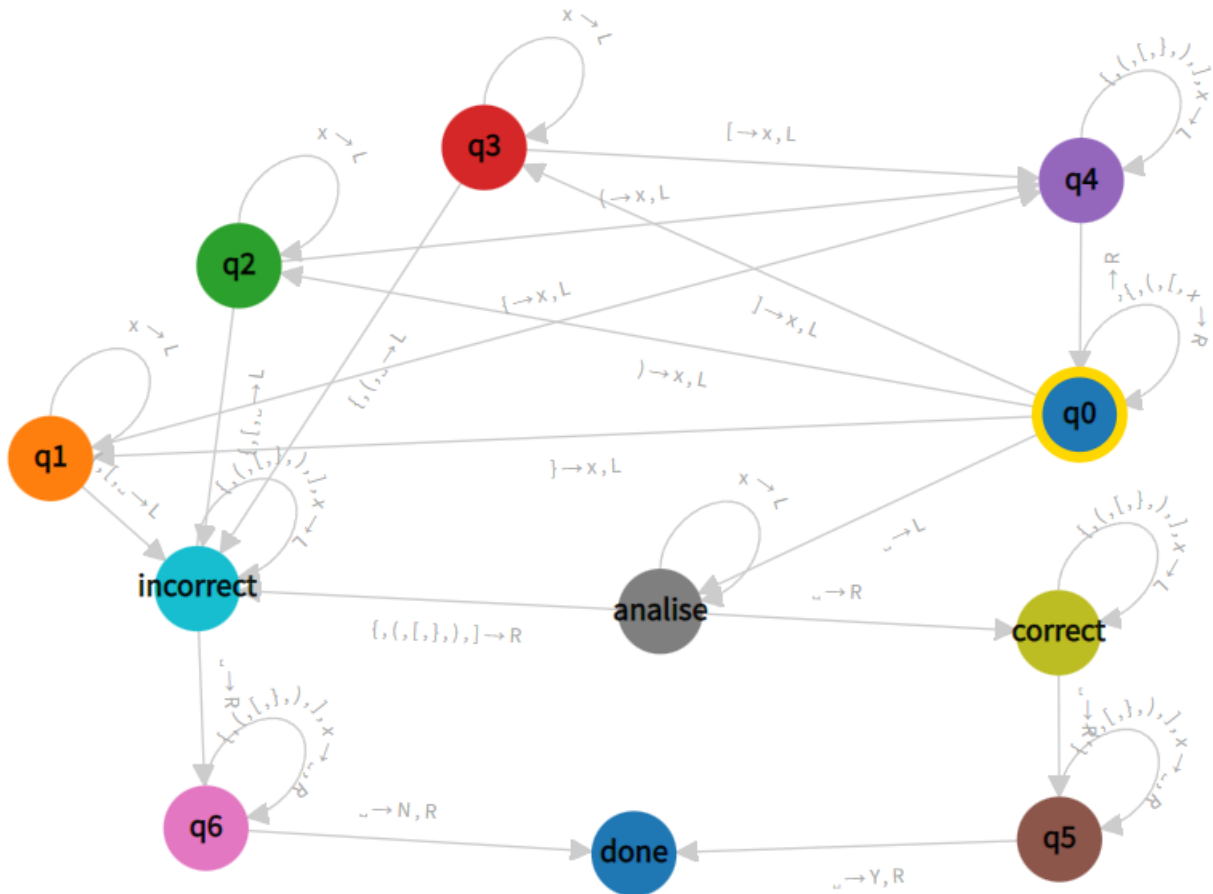
Листинг:

```
1  input: '001122'
2  blank: ' '
3  start state: q0
4
5  table:
6    q0:
7      'x': R
8      0: {write: 'x', R: q1}
9      [1,2]: {L: q5}
10     ' ': {R: include}
11    q1:
12      [0,'x']: R
13      1: {write: 'x', R: q2}
14      [2,' ']: {L: not_included}
15    q2:
16      [1,'x']: R
17      2: {write: 'x', R: q3}
18      [0,' ']: {L: not_included}
19    q3:
20      [0,1]: {L: not_included}
21      [2]: {L: q4}
22      [' ']: {L: analyse}
23    q4: # repeat q1-q3
24      [0,1,2,'x']: L
25      ' ': {R: q0}
26    q5:
27      [0,1,2,'x']: L
28      ' ': {R: q6}
29    q6:
30      [0,1,2,'x']: {write: ' ', R}
31      ' ': {R: not_included}
32    analyse:
33      'x': L
34      [0,1,2]: {L: not_included}
35      [' ']: {R: include}
36    not_included:
37      [0,1,2,'x']: {L: q5}
38      ' ': {write: N, R: done}
39    include:
40      [0,1,2,'x']: {write: ' ', R}
41      ' ': {write: Y, R: done}
42    done:
```

Описание **.yaml** к соответствующей задаче находится в папке МТ.

2. Проверка соблюдения правильности скобок в строке (минимум 3 вида скобок) (0.5 балла)

Решение:



Листинг:

```
1  input: '{()[]}'
2  blank: ' '
3  start state: q0
4
5  table:
6    q0:
7      ' ': {L: analyse}
8      ['{','(','[' , 'x']: R
9      '}' : {write: 'x', L: q1}
10     ')' : {write: 'x', L: q2}
11     ']' : {write: 'x', L: q3}
12    q1:
13      ['(', '[', ' ']: {L: incorrect}
14      '{': {write: 'x', L: q4}
15      'x': L
16    q2:
17      '(' : {write: 'x', L: q4}
18      'x': L
19      ['{','[' , ' ']: {L: incorrect}
20    q3:
21      '[' : {write: 'x', L: q4}
22      'x': L
23      ['(', '(', ' ']: {L: incorrect}
24    q4: #repeat q1-q3
25      ['{','(','[' , '}', ')', ']', 'x']: L
26      ' ': {R: q0}
27    q5:
28      ['{','(','[' , '}', ')', ']', 'x']: {write: ' ', R}
29      ' ': {write: Y, R: done}
30    q6:
31      ['{','(','[' , '}', ')', ']', 'x']: {write: ' ', R}
32      ' ': {write: N, R: done}
33    analyse:
34      'x': L
35      ['{','(','[' , '}', ')', ']' ]: {R: incorrect}
36      ' ': {R: correct}
37    correct:
38      ['{','(','[' , '}', ')', ']', 'x']: L
39      ' ': {R: q5}
40    incorrect:
41      ['{','(','[' , '}', ')', ']', 'x']: L
42      ' ': {R: q6}
43    done:
```

Описание **.yaml** к соответствующей задаче находится в папке МТ.

Решение:



Листинг:

```
1  input: '101 11 1111'
2  blank: '*'
3  start state: q0
4
5  table:
6    q0:
7      1: {write: 'a', R: q1}
8      0: {write: 'b', R: q1}
9      ['a','b']: R
10     [' ','*']: {write: '#',R: q3}
11    q1:
12      [0,1]: R
13      ' ': {R: q0}
14      '*': {L: q2}
15    q2: # to begin
16      [0,1,'a','b',' ']: L
17      '*': {R: q0}
18    q3:
19      [0,1,'a','b',' ']: R
20      '*': {L: q4}
21    q4:
22      [0,1,' ','a','b']: {write: '*', L}
23      '#': {write: '*', L: q5}
24    q5:
25      a: {write: 1, L}
26      b: {write: 0, L}
27      ' ': {write: '*', L: q6}
28      '*': {R: done}
29    q6:
30      [0,1,'a','b',' ']: {write: '*', L}
31      '*': {R: q7}
32    q7:
33      '*': R
34      [0,1]: {L: done}
35    done:
```

Описание **.yaml** к соответствующей задаче находится в папке МТ.

3 Квантовые вычисления

3.1 Генерация суперпозиций (1 балл)

Дано N кубитов ($1 \leq N \leq 8$) в нулевом состоянии $|0\dots 0\rangle$. Также дана некоторая последовательность битов, которая задаёт ненулевое базисное состояние размера N . Задача получить суперпозицию нулевого состояния и заданного.

$$|S\rangle = \frac{1}{\sqrt{2}} (|0\dots 0\rangle + |\psi\rangle)$$

То есть требуется реализовать операцию, которая принимает на вход:

1. Массив кубитов q_s
2. Массив битов *bits* описывающих некоторое состояние $|\psi\rangle$. Этот массив имеет тот же самый размер, что и q_s .

Первый элемент этого массива равен 1.

Листинг:

```
namespace Solution {  
    open Microsoft.Quantum.Primitive;  
    open Microsoft.Quantum.Canon;  
    operation Solve(qs: Qubit[], bits: Bool[]) : ()  
    {  
        body  
        {  
            H(qs[0]);  
            for (i in 1..Length(qs) - 1)  
            {  
                if (bits[i])
```

$$\{ \text{CNOT}(\text{qs}[0], \text{qs}[i]);$$

3.2 Различение состояний 1 (1 балл)

Дано N кубитов ($1 \leq N \leq 8$), которые могут быть в одном из двух состояний:

$$|GHZ\rangle = \frac{1}{\sqrt{2}} (|0...0\rangle + |1...1\rangle)$$

$$|W\rangle = \frac{1}{\sqrt{N}} (|10\dots 00\rangle + |01\dots 00\rangle + \dots + |00\dots 01\rangle)$$

Требуется выполнить необходимые преобразования, чтобы точно различить эти два состояния. Возвращать 0, если первое состояние и 1, если второе.

Листинг:

```
namespace Solution {
  open Microsoft.Quantum.Primitive;
  open Microsoft.Quantum.Canon;
  operation Solve(qs: Qubit[]) : Int
  {
    body
    {
      mutable one = 0;
```

```

        for (q in qs)
            {
                if (M(q) == One)
                    {
                        set one = one + 1;
                    }
            }
        if (one == 1)
            {
                return 1;
            }
        else
            {
                return 0;
            }
    }
}

```

3.3 Различение состояний 2 (2 балла)

Дано 2 кубита, которые могут быть в одном из двух состояний:

$$|S_0\rangle = \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

$$|S_1\rangle = \frac{1}{2} (|00\rangle - |01\rangle + |10\rangle - |11\rangle)$$

$$|S_2\rangle = \frac{1}{2} (|00\rangle + |01\rangle - |10\rangle - |11\rangle)$$

$$|S_3\rangle = \frac{1}{2} (|00\rangle - |01\rangle - |10\rangle + |11\rangle)$$

Требуется выполнить необходимые преобразования, чтобы точно различить эти четыре состояния. Возвращать требуется индекс состояния (от 0 до 3).

Листинг:

```
namespace Solution {
    open Microsoft.Quantum.Canon;
    open Microsoft.Quantum.Primitive;
    operation Solve (qs: Qubit[]) : Int
    {
        body
        {
            H(qs[0]);
            H(qs[1]);
            if (M(qs[0]) == Zero)
            {
                if (M(qs[1]) == Zero)
                {
                    return 0;
                }
                else
                {
                    return 1;
                }
            }
            else
            {
                if (M(qs[1]) == Zero)
                {
                    return 2;
                }
                else
            }
        }
    }
}
```

```

    {
        return 3;
    }
}
}
}
}

```