

Домашнее задание №3

Сабитов Сергей А-13а-19

1 июня 2022 г.

1 Машины Тьюринга

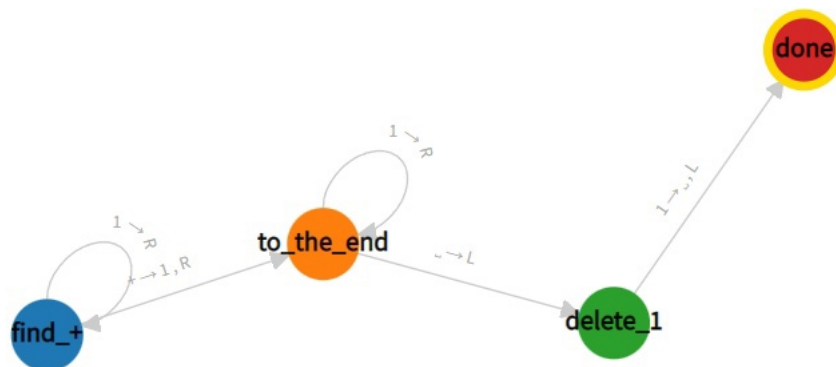
Работу требуется выполнять в системе turingmachine.io.

Для сдачи заданий 1-2 требуется прикрепить файлы YAML с исходным кодом проекта. Каждый файл должен иметь наименование задание_пункт.yml, к примеру 1_1.yml для первой задачи первого задания.

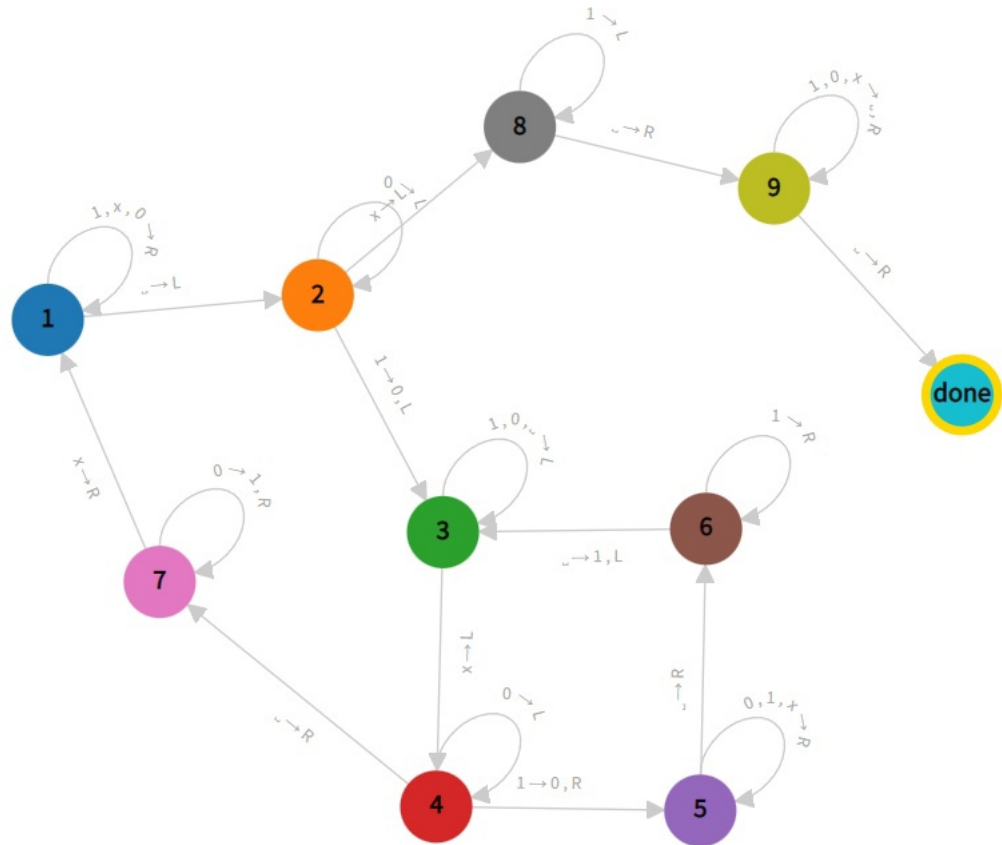
1.1 Операции с числами

Реализуйте машины Тьюринга, которые позволяют выполнять следующие операции:

1. Сложение двух унарных чисел (1 балла)



2. Умножение унарных чисел (1 балл)

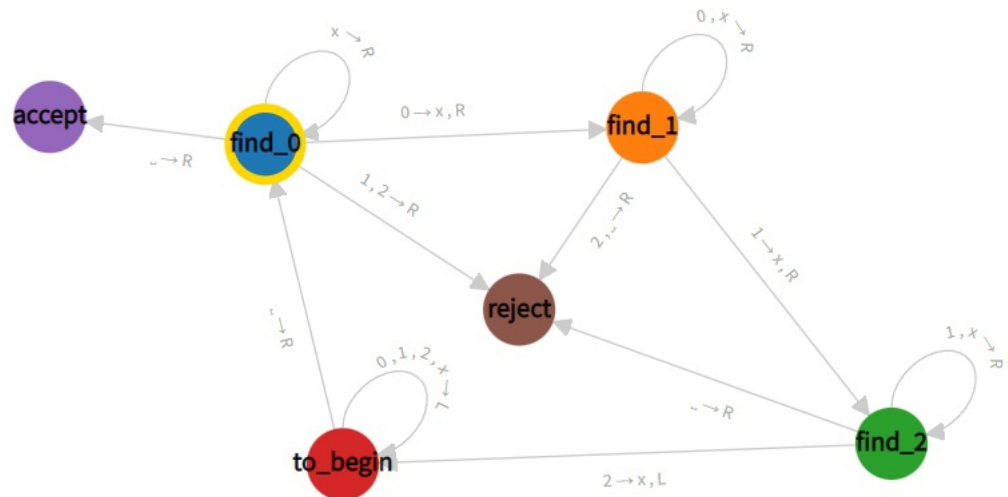


1.2 Операции с языками и символами

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)

МТ строится аналогично той, что мы строили на лекции

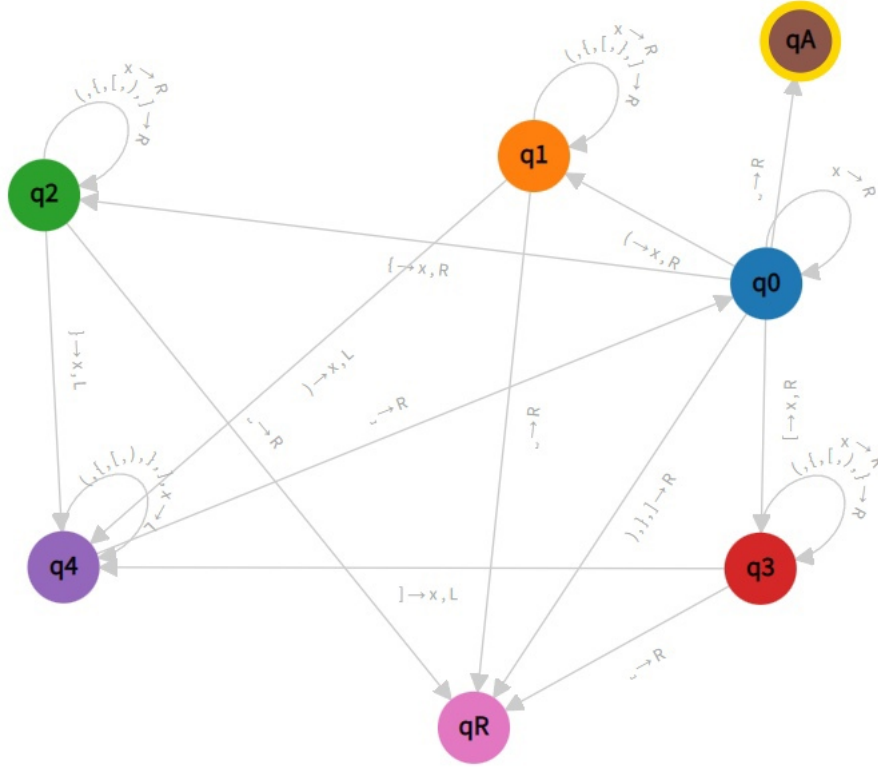


2. Проверка соблюдения правильности скобок в строке (минимум 3 вида скобок) (0.5 балла)

МТ принимает последовательность скобок и в конце мы смотрим на конечное состояние: если это q_A - то скобочная последовательность правильная; если q_R - неправильная.

Алгоритм заключается в том, что

- мы считываем открывающую скобку и помечаем ячейку
- ищем соответствующую закрывающую и так же помечаем ячейку
- идем в начало до пустого символа и начинаем с 1 пункта



3. Поиск минимального по длине слова в строке (слова состоят из символов 1 и 0 и разделены пробелом) (1 балл)

2 Квантовые вычисления

В качестве решения задачи надо предоставить схему алгоритма для частного случая при фиксированном количестве кубитов и фиксированных состояниях.

2.1 Генерация суперпозиций 1 (1 балл)

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

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

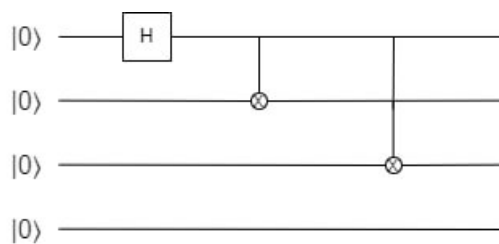
То есть требуется реализовать операцию, которая принимает на вход:

1. Массив кубитов q_s
2. Массив битов $bits$ описывающих некоторое состояние ψ . Это массив имеет тот же самый размер, что и q_s . Первый элемент этого массива равен 1.

Код:

```
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]) {
                    CNOT(qs[0], qs[i]);
                }
            }
        }
    }
}
```

Схема для случая $|0000\rangle, |1110\rangle$:



2.2 Различение состояний 1 (1 балл)

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

$$GHZ = \frac{1}{\sqrt{2}}(0 \dots 0 + 1 \dots 1)$$

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

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

Код:

```
namespace Solution {
    open Microsoft.Quantum.Primitive;
    open Microsoft.Quantum.Canon;
    operation Solve (qs : Qubit[]) : Int
    {
        body
        {
            mutable ones = 0;
            for (i in 0..Length(qs) - 1) {
                if (M(qs[i]) == One) {
                    set ones = ones + 1;
                }
            }
            if (ones == Length(qs) or ones == 0)
            {
                return 0;
            }
            return 1;
        }
    }
}
```

2.3 Различение состояний 2 (2 балла)

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

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

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

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

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

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

Заготовка для кода:

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

            return

        }
    }
}
```

2.4 Написание оракула 1 (2 балла)

Требуется реализовать квантовый оракул на N кубитах ($1 \leq N \leq 8$), который реализует следующую функцию: $f(\mathbf{x}) = (\mathbf{bx}) \bmod 2$, где $\mathbf{b} \in \{0, 1\}^N$ вектор битов и \mathbf{x} вектор кубитов. Выход функции записать в кубит \mathbf{y} . Количество кубитов N ($1 \leq N \leq 8$).

Заготовка для кода:

```
namespace Solution {
    open Microsoft.Quantum.Primitive;
    open Microsoft.Quantum.Canon;
    operation Solve (x : Qubit[], y : Qubit, b : Int[]) : ()
    {
        body
        {

        }
    }
}
```