

Министерство образования и науки РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«МЭИ»

Прикладная математика и информатика

Кафедра прикладной математики и искусственного интеллекта

Теоретические модели вычисления

Домашнее задание №3
Машины Тьюринга и квантовые вычисления

Преподаватель: Ивлиев С. А.
Студент: Соколова А.С.

Москва 2022

Содержание

1	Машины Тьюринга	3
1.1	Операции с числами	3
1.2	Операции с языками и символами	6
2	Квантовые вычисления	10
2.1	Генерация суперпозиций 1 (1 балл)	10
2.2	Различение состояний 1 (1 балл)	11

1 Машины Тьюринга

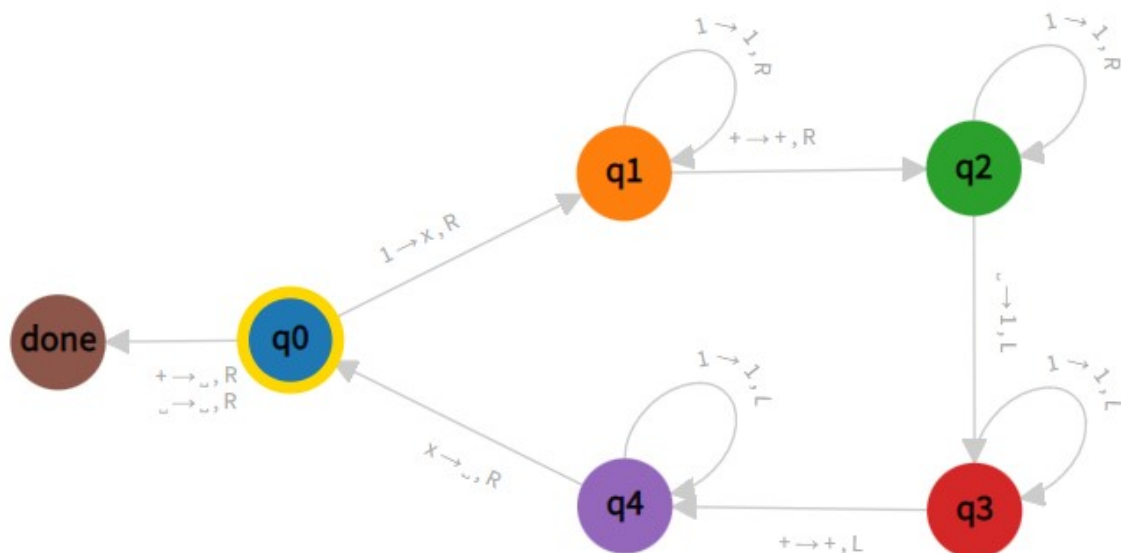
Работу требуется выполнять в системе turingmachine.io.

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)



Используем дополнительный символ x

Считаем первую единицу, заменяем ее на x . Идем до конца, считываем пробел и заменяем его на единицу. Идем влево до x . Считываем x , заменяем его на пробел.

Каждую следующую единицу по очереди заменяем на x , копируем единицу в конец и т.д.

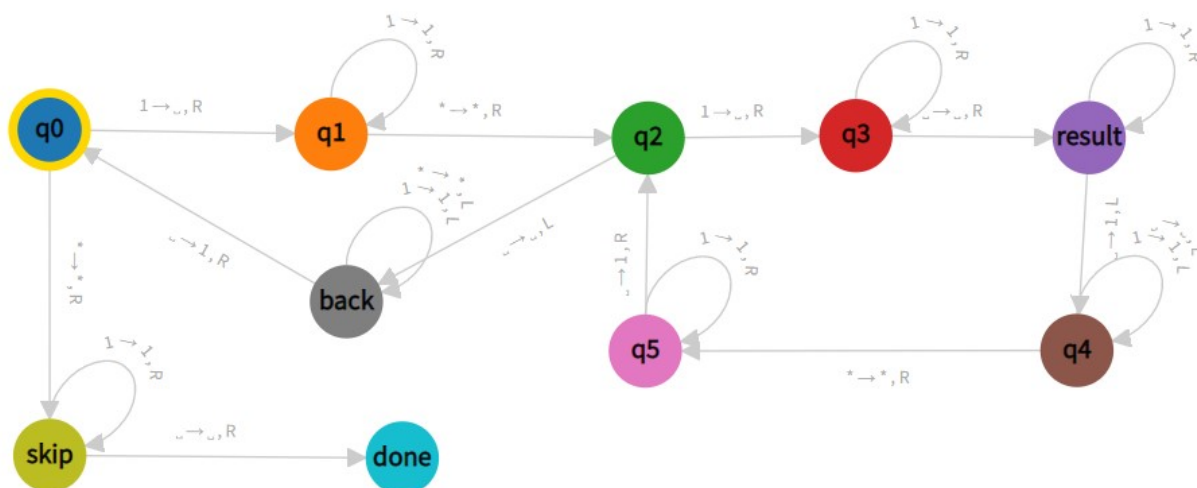
$1+1 \rightarrow 11$

$111+1 \rightarrow 1111$

$1+11111 \rightarrow 111111$

	1	+	x	' '
q0	<q1,x,R>	<done,' ',R>		<done,' ',R>
q1	<q1,1,R>	<q2,+,R>		
q2	<q2,1,R>			<' ',1,L>
q3	<q3,1,R>	<q4,+,L>		
q4	<q4,1,L>		<q0,' ', R>	

2. Умножение унарных чисел (1 балл)



$1*1 \rightarrow 1$
 $111*1 \rightarrow 111$
 $1*11111 \rightarrow 11111$

	1	*	' '
q0	<q1,' ',R>	<skip,*,R>	
q1	<q1,1,R>	<q2,*,R>	
q2	<q3,' ',R>	<back,' ',L>	
q3	<q3,1,R>		<result,' ',R>
result	<result,1,R>		<q4,1,L>
q4	<q4,1,L>	<q5,*,R>	<q4,' ',L>
q5	<q5,1,R>		<q2,1,R>
back	<back,1,L>	<back,*,L>	<q0,1,R>
skip	<skip,1,R>		<done,' ',R>

Рассмотрим подробнее на простом примере $11*1$

1. Переходим в состояние q0 по 1 и заменяем 1 на пробел и передвигаемся правее. Получаем промежуточный результат:

	1
q0	<q1,' ',R>

1*1

2. Считываем следующую единицу, переходим в состояние q1 по 1. Считываем единицы и передвигаемся правее пока не встретим *. То есть переходим в состояние q2 по *. Промежуточный результат без изменения, только головка передвигается:

	1	*
q1	<q1,1,R>	<q2,*,R>

1*1

3. Далее после знака * встречаем единицу. То есть переходим в состояние q2 по 1 и заменяем 1 на пробел. Далее попадаем в состояние q3:

	1
q2	<q3,' ',R>

1*

4. Считываем еще один пробел в состоянии q3 и переходим в состояние result:

	' '
q3	<result,' ',R>

1*

5. Так как текущая ячейка пробел, то из состояния result по пробелу переходим в состояние q4, заменяя пробел на единицу и передвигая головку левее:

	' '
result	<q4,1,L>

1*

6. Из состояния q4 по пробелу и единице двигаемся левее, пропуская их:

	1	' '
q4	<q4,1,L>	<q4,' ',L>

1*

7. Далее встречаем умножение * и переходим в состояние q5 двигаясь правее. А из состояния q5 встречая пробел, заменяем его на и переходя в состояние q2 правее:

	*
q4	<q5,*,R>
	' '
q5	<q2,1,R>

1*1

8. Из состояния q2 по пробелу переходим в состояние back двигаемся правее. Затем из состояния back по единице и умножению двигаемся левее, переходя к начальному состоянию. Когда встречаем пробел, заменяем его на единицу:

	1	*	' '
back	<back,1,L>	<back,*,L>	<q0,1,R>

11*1_1

9. Следующую единицу снова заменяем на пробел и т д :

	1
q0	<q1,' ',R>

1_ *1_1

10. 1_ * _ _ 1

11. 1_ * _ _ 11

12. 1_ *1_ 11

13. 11*1_ 11

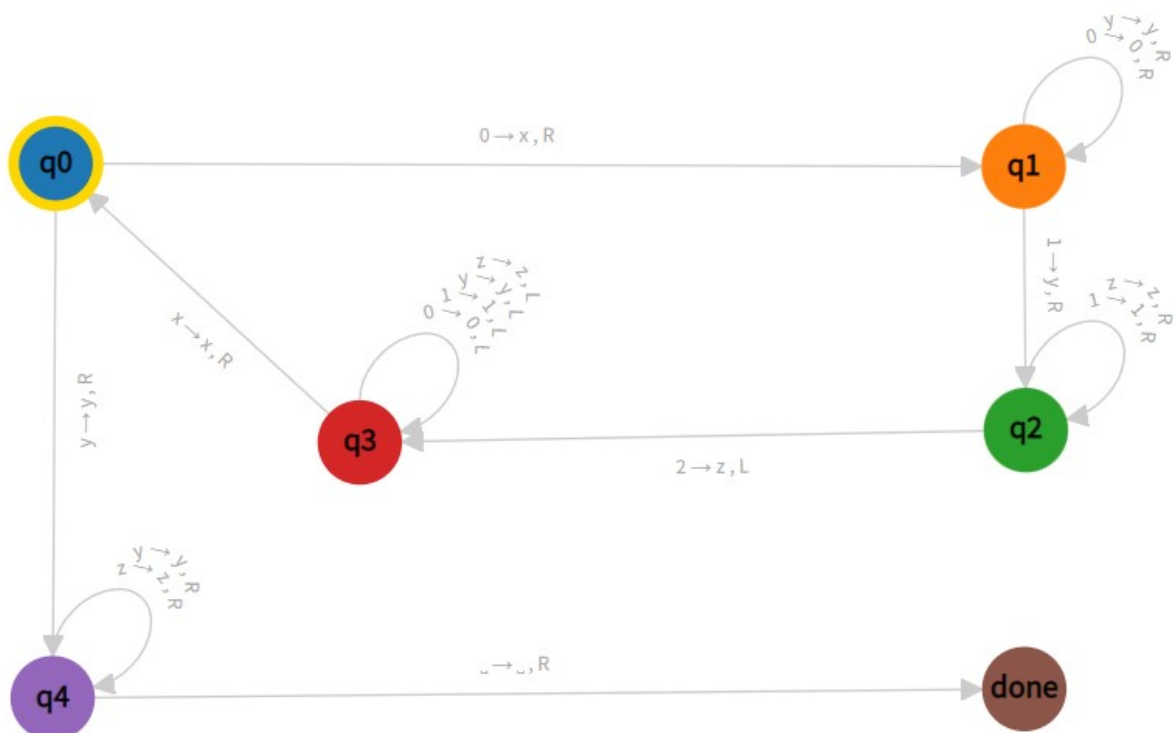
14. Передвигаем головку на начало результата



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)



Примеры данных, при которых МТ корректно завершает работу:

input: '001122'

Как выглядит на ленте завершение программы:

x	x	y	y	z	z		
---	---	---	---	---	---	--	--

input: '012'

input: '001122'

input: '000111222'

input: '000011112222'

Примеры данных, при которых МТ некорректно завершает свою работу:

input: '00011222'

Как выглядит на ленте завершение программы:

x	x	x	y	y	z	z	2
---	---	---	---	---	---	---	---

input: '000122'

input: '221100'

input: '1122'

input: '00122'

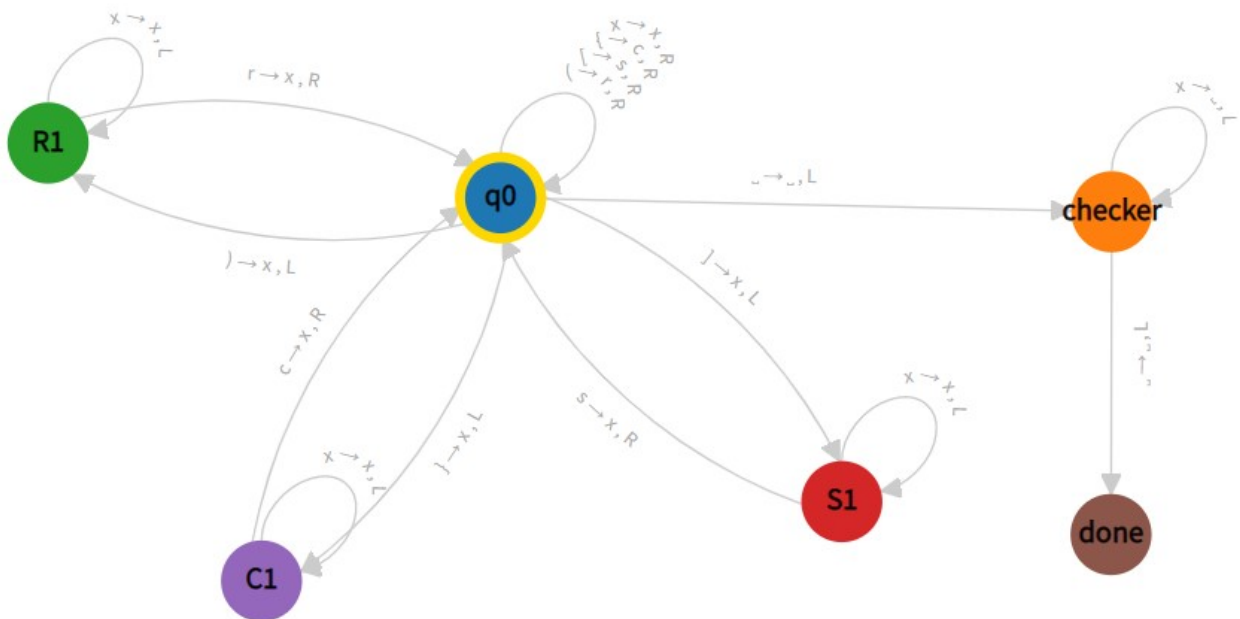
input: '00112'

input: '3'

input: '000111'

input: '01122'

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



Примеры данных, при которых МТ корректно завершает работу:

input: '([[]])'

input: '() []'

input: '() [] { }'

input: '() [] {}'

Примеры данных, при которых МТ некорректно завершает свою работу:

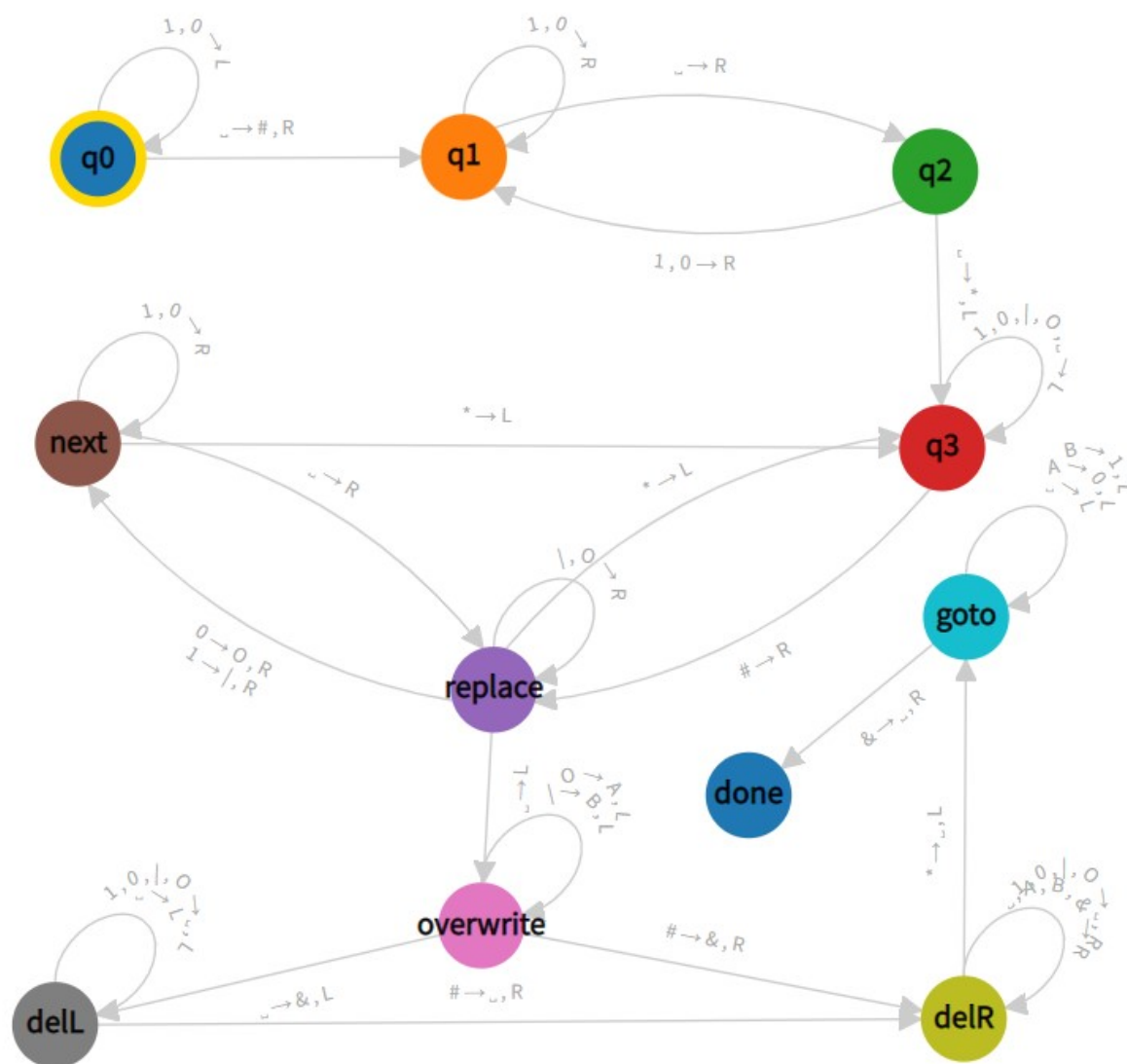
input: '([[])'

input: '() ({ [] } ([])'

input: '() (())'

input: '('

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



Тесты:

input: '10101 101 100'

Результат:

1	0	1
---	---	---

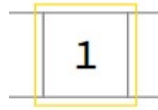
input: '11 01 10'

Результат:

1	1
---	---

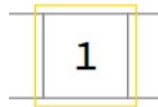
input: '1 101 110'

Результат:



input: '1'

Результат:



2 Квантовые вычисления

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.

Заготовка для кода:

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

```

```

    }
  }
}

```

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.

Заготовка для кода:

```

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

            return

        }
    }
}

```