

Теоретические модели вычислений

ДЗ №3: Машины Тьюринга и квантовые вычисления

1 мая 2022 года

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)
Алгоритм унарного сложения:
 1. Удаляем единицу первого числа и заменяем $+$ на 1.
2. Умножение унарных чисел (1 балл)

Алгоритм унарного умножения:

1. Добавляем знак равенства после второго числа и возвращаемся в начало первого числа
2. Заменяем единицу на А (тем самым определяем, какие единицы первого числа уже использованы) и идем до второго числа (после знака x)
3. Заменяем единицу на В (тем самым определяем, какие единицы второго числа уже использованы) и добавляем единицу после знака равенства. Идем обратно ко второму числу, пока не встретим В. Повторяем этот пункт, пока не будут перебраны все единицы второго числа и скопированы за знаком равенства.
4. Заменяем все символы В на единицы, идем обратно к первому числу, пока не встретим символ А, и сдвигаем головку вправо. Если встретим единицу переходим в пункт 2, тем самым пока перебираем все единицы первого числа. Если встретим знак умножения, то удаляем первые два числа, знаки умножения и равенства, оставив только результат.

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)

Для слов, принадлежащих языку, будем выводить символ Т, в противном случае - F.

Алгоритм:

1. Рассматриваем слова, в которых первый символ пустой или 0, иначе слово не принадлежит языку.
2. Заменяем 0 на символ А, переходим в состоянии $replace_1$.
3. В состоянии $replace_1$ если встретим следующие символамы, то слово не принадлежит языку: 2, С, пустой символ. Пропускаем символы 0 и В. Заменяем 1 на символ В, переходим в состояние $replace_2$.
4. В состоянии $replace_2$ если встретим пустой символ или 0, то слово не принадлежит языку. Пропускаем символы 1 и С. Заменяем 2 на символ С. Возвращаемся обратно, до тех пор, пока не встретим символ А.
5. Если символ справа от А это 0 (остались еще нули), то переходим в пункт 2. Если - В, то значит, что все нули закончились, соответственно если слово принадлежит языку, то и 1 и 2 должны закончиться.
6. Начиная самого левого символа удаляем символы А, В, С. Если все символы закончились, значит слово принадлежит языку. Если при чтении очередного символа получили 0, 1 или 2, то значит, слово не принадлежит языку.

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

1. Если встретим закрывающую скобку, то заканчиваем. Иначе ищем открывающую скобку и помечаем эту скобку буквой А и ищем закрывающую скобку этого же типа, двигаясь направо. Если не нашли открывающую и закрывающую скобки, то это значит, что мы перебрали все слово, тогда удаляем все слово и пишем символ Т.
2. Если достигли пустого слова, то слово не принадлежит языку. Иначе помечаем найденную закрывающую скобку буквой А и идем к началу. Переходим к пункту 1.

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

- 1.
- 2.

2 Квантовые вычисления

Для выполнения заданий по квантовым вычислениям требуется QDK. Его можно скачать здесь: <https://docs.microsoft.com/en-us/azure/quantum/install-overview-qdk>.

Но можно использовать любой пакет, типа <https://qiskit.org/>.

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 $|0 \dots 0\rangle$. Также дана некоторая последовательность битов, которое 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:

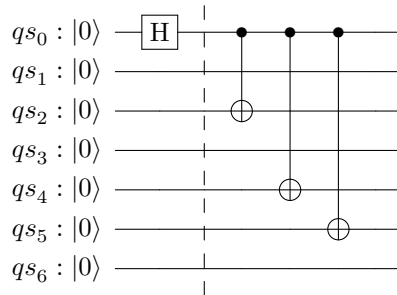
1. Массив кубитов q_s
2. Массив битов $bits$ описывающих некоторое состояние $|\psi\rangle$. Это массив имеет тот же самый размер, что и q_s . Первый элемент этого массива равен 1.

```

1 def Solve1(qs, psi):
2     global circuit
3     # применяем оператор Адамара для первого кубита
4     circuit.h(0)
5     circuit.barrier()
6     # для всех единичных кубитов применяем оператор CNOT с первым
7     for i in range(1, len(psi)):
8         if psi[i] == 1: circuit.cx(qs[0], qs[i])

```

$$\psi = |1010110\rangle$$



2.2 Различение состояний 1 (1 балл)

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

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

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

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

1. Первое состояние: при измерении все кубиты равны 0 или 1.
2. Второе состояние: при измерении только один кубит равен 1, остальные равны 0

При $N = 1$ различить состояния невозможно (в этом случае - $|W\rangle$)

```

1  def Solve2(q):
2      msr = q.measure()[0]
3      one_count = 0
4      for q in msr:
5          if q == '1': one_count += 1
6      if one_count == 1:
7          return 'W'
8      else:
9          return 'GHZ'
10 Solve2(GHZ)

```

2.3 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).

Применяем оператор Адамара для обоих кубитов и измеряем получившиеся состояния.

$$H^2 = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \end{pmatrix}$$

$$H^2 |S_0\rangle = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$H^2 |S_1\rangle = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \times \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$H^2 |S_2\rangle = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$H^2 |S_4\rangle = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \times \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

```

1  def Solve3(q):
2      global circuit
3      circuit.initialize(q)
4      circuit.h(0)
5      circuit.h(1)
6
7      state = Statevector(circuit)
8      msr = state.measure()[0]
9      if msr == '00':
10         return 0
11     elif msr == '01':
12         return 1
13     elif msr == '10':
14         return 2
15     elif msr == '11':
16         return 3
17
18     circuit = QuantumCircuit(2)
19     print(Solve3(S3))

```

2.4 Написание оракула 1 (2 балла)

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

```