

Теоретические модели вычислений

ДЗ №3: Машины Тьюринга и квантовые вычисления

Игумнов Олег

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: заменяем плюс на 1, и удаляем последний символ

2. Умножение унарных чисел (1 балл)

Алгоритм: сначала ставим в конце флаг для результата (=), затем копируем число справа до тех пор пока не кончатся единицы левого числа

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)

Алгоритм: учитываем случай $n=0$, будем ходить по строке сначала удаляя нули и заменяя единицы на символ а, затем, учитывая, что символов а столько же, сколько было нулей будем удалять символы а и заменять двойки на символ б, при этом будут совершаться проверки на соблюдение порядка следования символов

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

Алгоритм: учитываем случай пустого слова, ищем правую скобку, затем парную ей левую, заменяем обе на символ, возвращаемся в начало опять ищем правую скобку и тд.

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

Алгоритм: поочередно заменяем символы в каждом слове, то слово, после которого будет обнаружен пробел и есть искомое, помечаем его и удаляем лишнее, после возвращаем ему нормальный вид

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.

Частный случай

$$|S\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |101\rangle)$$

см. рис 1. 2.1

код:

```
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]);
            }
        }
    }
}
```

2.2 Различение состояний 1 (1 балл)

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

Пусть $N=3$

$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

$$|W\rangle = \frac{1}{\sqrt{3}}(|100\rangle + |010\rangle + |001\rangle)$$

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

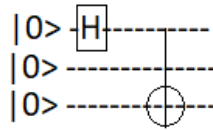
код:

```

operation Solve (qs: Qubit[]) : Int
{
    body
    {
        mutable countOnes = 0;
        for (i in 0..Length(qs) - 1)
        {
            if (M(qs[i]) == One)
            {
                set counter = counter + 1;
            }
        }
        if (counter == 1)
        {
            return 1;
        }
        return 0;
    }
}

```

Рис. 1: 2.1



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).

код:

```

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;
        }
    }
}
}

```

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 ($1 \leq N \leq 8$).

код:

```

operation Solve (x: Qubit[], y: Qubit, b: Int[]) : ()
{
    body
    {
        Reset(y);
        for (i in 0..Length(x) - 1)
        {
            if (b[i] == 1) {
                CNOT(x[i], y);
            }
        }
    }
}

```

схема для входных данных [1 0 1 1]

Рис. 2: 2.4

