

ДЗ №3: Машины Тьюринга и квантовые вычисления

Николай Малахов

June 2022

1 Задание 2.1.1

Реализуйте машину Тьюринга, которая позволяет выполнить сложение двух унарных чисел.

Алгоритм

На вход принимается строка вида " $n + m =$ "

Машина начинает в состоянии `record_start`. Считывается единица (при её наличии), удаляется из ячейки и переносится в конец, справа от последней единицы за знаком "=" (или самим знаком "=" в тривиальном случае). Бегунок возвращается в начало, на первом встретившемся пробеле восстанавливает "украденную" единицу, после чего повторяет алгоритм, начиная со следующей ячейки.

Знак "+" пропускается, после перехода через знак "=" в состоянии поиска единицы, программа завершается.

Скрипт решения: `2_1_1.yaml`

2 Задание 2.1.2

Реализуйте машину Тьюринга, которая позволяет выполнить умножение двух унарных чисел.

Алгоритм

На вход принимается строка вида " $n * m$ "

Машина начинает в состоянии `loop`. Считывается единица (при её наличии), удаляется из ячейки и переносится в начало следующего множителя, откуда так же "забирает" единицу. Бегунок переносится в конец, пропуская один пробел (предполагаемый знак `=`, который было лень отдельно обрабатывать), где записывает единицу нового числа. После этого машина возвращает "украденную" единицу из второго множителя, и переходит на

следующую, входя в цикл обработки второго множителя. Повторяя описанные выше действия, машина обрабатывает умножение одного разряда первого множителя, а затем возвращается в начало строки.

Описанный выше алгоритм циклически выполняется, пока все единицы первого множителя не будут считаны (считанные ранее единицы восстанавливаются, аналогично обработке второго множителя).

Обработка заканчивается, когда все разряды первого множителя были считаны и обработаны.

Скрипт решения: 2_1_2.yaml

3 Задание 2.2.1

Реализуйте машину Тьюринга, которая позволяет проверить принадлежность языку

$$L = \{0^n, 1^n, 2^n\}, n \geq 0$$

Алгоритм

Машина начинает в состоянии find0.

На вход принимается строка из символов 0,1,2.

Производится поиск первого символа 0. В случае обнаружения любого другого символа (за исключением спецсимвола s), МТ переходит в состояние gej и останавливает работу. Бегунок движется влево при поиске, т.к. предполагается что исходная строка начинается с 0 (иначе она пустая или неправильная), после чего поиск нуля всегда вызывается из ячеек с двойками, поэтому и при поиске движемся влево).

Найденный 0 заменяется на спецсимвол, после чего переходит к поиску символа 1.

Поиск символа 1 допускает обнаружение символа 0 и спецсимвола s, однако останавливает работу в состоянии gej при обнаружении 2 (т.к. на каждый найденный 0 должна быть 1, и только потом 2). Действия производятся аналогичные (проход в право и замена первого найденного символа 1 спецсимволом s), после чего МТ переходит к поиску символа 2.

Поиск символа 2 осуществляется полностью аналогично, запрещая появление символа 0 или пробела, а единицы и спецсимволы пропуская осознанно. После успешно найденного символа 2, МТ возвращается к поиску 0, двигаясь влево.

В случае, если программа не смогла обнаружить в очередной раз символ 0, она достигнет пробела слева и начнет проверку строки на наличие

лишних символов 1 и 2. Это будет означать, что исходная строка не соответствует языку, т.к. в таком случае все 1 и 2 заменены спецсимволом.

Пройдя полностью от левого до правого пробела в состоянии проверки, МТ завершит работу в состоянии асс, означающем принадлежность исходной строки языку. (Можно было бы реализовать обратную расшифровку, но как-то я не подумал заранее)

Скрипт решения: 2_2_1.yaml

4 Задание 2.2.2

Реализуйте машину Тьюринга, которая позволяет осуществить проверку соблюдения правильности скобок в строке (три вида скобок)

Алгоритм.

Машина начинает в состоянии `search_close`. Ищем первую закрывающую скобку в строке. Затем меняем состояние и переходим в движение налево в поиске первой открывающейся строки. Закрывающаяся скобка нам встретиться не может, т.к. мы взяли самую первую (левую) из них. Если первая открывающая скобка перед данной не соответствует найденной закрытой, МТ останавливается в состоянии `rej`. Иначе, обе скобки заменяются спецсимволом `p` (опять же, можно было добавить индивидуальные символы и менять на скобки обратно после обработки, но в ТЗ этого нет). Цикл повторяется, пока вся строка не будет состоять из спецсимволов (тогда МТ закончит в состоянии `асс`), или пока МТ не остановится в состоянии `rej` на каком-то из этапов обработки.

Скрипт решения: 2_2_2.yaml

5 Задание 2.2.3

Реализуйте машину Тьюринга, которая позволяет осуществить поиск минимального по длине слова в строке (слова состоят из символов 1 и 0 и разделены пробелом)

Алгоритм.

В данной МТ пустым символом является символ `e` (аналог ε). Проходим по строке слева направо, заменяя в каждом слове первый встреченный символ 0 или 1 на `o` и `x` соответственно. Заменяв символ, МТ переходит в другое состояние, в котором пропускает все оставшиеся символы

слова до пробела. Пробел сбрасывает состояние обратно в поиск и перезаписывает первого символа 0/1. Достигнув правой границы, обработка запускается зеркально до левой.

Как только при поиске новой "жертвы" встречается символ пробела или эpsilon, начинается обратный перевод найденного минимального слова в исходное состояние (перевод осуществляется в направлении, противоположном тому, в котором двигался бегунок во время поиска нового символа для замены).

МТ останавливается в состоянии done, на ячейке, принадлежащей найденному слову (первому или последнему символу, в зависимости от направления перевода).

Скрипт решения: 2_2_3.yaml

6 Задание 3.1. Генерация суперпозиций

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.

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

То есть требуется реализовать операцию, которая принимает на вход:

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

Первые кубиты векторов различны, поэтому используем оператор Адамара к первому кубиту Все кубиты qs равны 0 \Rightarrow путаем 1-ый кубит, если $bits[i] = 1$

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

}

7 Задание 3.2 Различение состояний

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.

Пояснения: Чтобы измерить состояние системы надо измерить кубиты 1-ое состояние - $N > 1$: N нулей, либо N единиц 2-ое состояние - 1 единица При $N = 1$ состояния неразличимы

```
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) { // measurement
                    set ones += 1;
                }
            }
            if (ones == 1) {
                return 1;
            }
            return 0;
        }
    }
}
```