

Теоретические модели вычислений

ДЗ №3: Машины Тьюринга и квантовые вычисления

Рамазанов Никита, А-13а-19

1 июня 2022

1 Машины Тьюринга

1.1 Операции с числами

Реализуйте машины Тьюринга, которые позволяют выполнять следующие операции:

1. Сложение двух унарных чисел (1 балл).

Алгоритм:

- Ищем знак $+$ и заменяем его на $|$ (элемент унарной системы счисления);
- Сдвигаем головку в конец унарного числа;
- Удаляем последний элемент числа.

Реализация - папка Task1_1, файл 'Subtask1.yml'.

2. Умножение унарных чисел (1 балл).

Алгоритм:

- Берем начало первого числа (первый знак $|$);
- Удаляем его, идем в конец второго, после него указываем разделительный знак (в нашем случае - $*$);
- Полностью копируем второй множитель;
- Возвращаемся в начало и проделываем то же самое для оставшихся цифр в числе;
- В конце алгоритма все, что осталось до разделительного символа, в том числе сам этот символ, удаляется.

Реализация - папка Task1_1, файл 'Subtask2.yml'.

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:

- Ищем в слове последние "0", "1" и "2" и заменяем их на "N";
- Повторяем, пока на ленте не останется цифр;
- Если встречаем элемент, отличный от "N", то очищаем ленту и пишем "N" - то есть слово не принадлежит языку, иначе - очищаем и пишем "Y" - это означает, что слово принадлежит языку.

Реализация - папка Task1_2, файл 'Subtask1.yml'.

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

Алгоритм:

- Ищем первую закрывающую скобку, заменяем ее на "N";

- Возвращаемся на один шаг назад - здесь должна быть открывающая скобка такого же вида, иначе - ошибка, и заменяем ее на "N";
- Идем в начало и повторяем алгоритм: сначала первая закрывающая, затем последняя открывающая перед рядом символов "N";
- Если на каком-то этапе ошибка - очищаем ленту и записываем "N" то есть скобочная последовательность не является правильной.
- Если слева от "N" пустой символ, то очищаем все символы "N" до первого символа, отличного от "N" если этот символ - тоже пустой, то скобочная последовательность правильная - записываем "Y".

Реализация - папка Task1_2, файл 'Subtask2.yml'.

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

Чтобы отличить пробел и пустой символ, будем считать пробелом знак _.

Алгоритм:

- Будем заменять 1 на "a", 0 на "b" сначала первый символ каждого слова, затем второй, и так далее;
- Если на каком-то шаге справа будет пустой символ, это означает, что минимальное слово найдено - ставим справа от него знак *. – Удаляем все что справа, возвращаемся и удаляем все, что слева.

Реализация - папка Task1_2, файл 'Subtask3.yml'.

2 Квантовые вычисления

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.

Решение:

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

```
    }
}
```

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.

Решение:

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

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).

Решение:

```

namespace Solution {
  open Microsoft.Quantum.Primitive;
  open Microsoft.Quantum.Canon;
  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;
        }
      }
    }
  }
}

```