

Теоретические модели вычислений

ДЗ №3: Машины Тьюринга и квантовые вычисления

Дмитрий Пугачёв А-05-19

May 2022

1 Машины Тьюринга

1.1 Операции с числами

Реализуйте машины Тьюринга, которые позволяют выполнять следующие операции:

1. Сложение двух унарных чисел

Алгоритм для решения этой задачи предельно прост:

Для ее решения понадобилось 3 ключевых состояния, первое для поиска знака "+" и замены его на 1 (элемент унарной системы счисления), затем состояние, сдвигающее головку до конца слова и завершающее состояние, которое удаляет последний элемент, так как он "переносится" на место знака сложения.

Код находится по адресу "Task1/1_1.yml".

2. Умножение унарных чисел

Алгоритм для этой задачи работает следующим образом:

Сперва мы совершаем проход по первому числу, пока не встретим 1 (как и в прошлой задаче это элемент унарной системы). Затем для каждой 1 во втором числе мы делаем ее копию после исходного выражения. Когда второе число закончится (встретим пробельный символ) - возвращаемся в начало и проделываем то же самое для оставшихся цифр в числе. В конце всего алгоритма исходное выражение удаляется.

Код находится по адресу "Task1/1_2.yml".

1.2 Операции с языками и символами

Реализуйте машины Тьюринга, которые позволяют выполнять следующие операции:

1. Принадлежность к языку $L = \{0^n 1^n 2^n\}, n \geq 0$

Ключевыми компонентами алгоритма для этой задачи являются 3 состояния, которые поочередно проверяют наличие одинакового количества 0, 1 и 2. Достигается это путем замкнутого прохода по всему слову, причем при каждом проходе первые символы алфавита заменяются на a, b, c соответственно. Цикл завершает свою работу, если во время выполнения какой-либо вершины встречаем неожиданный элемент. Результатом программы является напечатанная буква "F" или "T" означающая отсутствие принадлежности или ее присутствие.

Код находится по адресу "Task2/2_2.yml".

2. Проверка соблюдения правильности скобок в строке (минимум 3 вида скобок) Следующий алгоритм реализует похожую идею, но абсолютно в другом порядке:

Происходит движение головки в сторону конца входного слова, при этом - если встречается какая либо закрывающая скобка, то алгоритм ставит на ее место символ, говорящий о том, что эта скобка уже рассматривалась, и затем головка меняет направление для поиска пары для нужной скобочки. Если все скобки обладают парами, то программа выведет "Т иначе "F".

Код находится по адресу "Task2/2_2.yml".

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

Идея этого алгоритма основана на рассуждениях из прошлой задачи. Имеется набор слов разной длины, если поочередно у каждого слова заменять символ (0 или 1) на букву (а или b) начиная с первого и заканчивая номером последнего символа в минимальном слове, то мы получим предложение, где только одно слово полностью состоит из букв. Такой результат достигается постепенной заменой всех цифр на буквы и при обратном ходе головки, проверками последнего элемента в слове. Если последний элемент буква, то буквы заменяются обратно на цифры и головка устанавливается на начало этого слова. Особенностью этого алгоритма является установка головки на последнее минимальное слово в предложении.

Код находится по адресу "Task2/2_3.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;
      }
    }
  }
}

```