

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ "МЭИ"

Отчет  
К домашней работе №3  
По теоретическим моделям вычислений  
Машины Тьюринга и квантовые вычисления

Выполнил  
Ушаков Н.А.  
Студент группы А-05-19

Москва 2022

# 1 Машины Тьюринга

## 1.1 Операции с числами

Реализуйте машины Тьюринга, которые позволяют выполнять следующие операции:

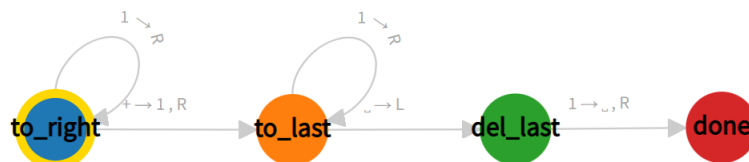
1. Сложение двух унарных чисел.

Реализация: МТ принимает два числа, представляющих последовательность из «1», разделенных знаком «+». МТ проходит вправо по ленте, до знака «+», заменяет его на «1», складывая таким образом числа, и удаляет последнюю лишнюю «1».

Листинг 1\_1.yaml:

```
input: '111+11'
blank: ' '
start state: to_right
table:
  to_right: # Движение вправо
    '1': R
    '+': {write: '1', R: to_last} # Меняем + на 1
  to_last: # Движение в конец
    '1': R
    ' ': {L: del_last}
  del_last: # Удаляем последнюю 1
    '1': {write: ' ', R: done}
  done:
```

Визуализация:



2. Умножение унарных чисел.

Реализация: МТ принимает два числа, представляющих последовательность из «1», разделенных знаком «\*». МТ последовательно уменьшает первое число и копирует второе на каждом шаге.

Листинг 1\_2.yaml:

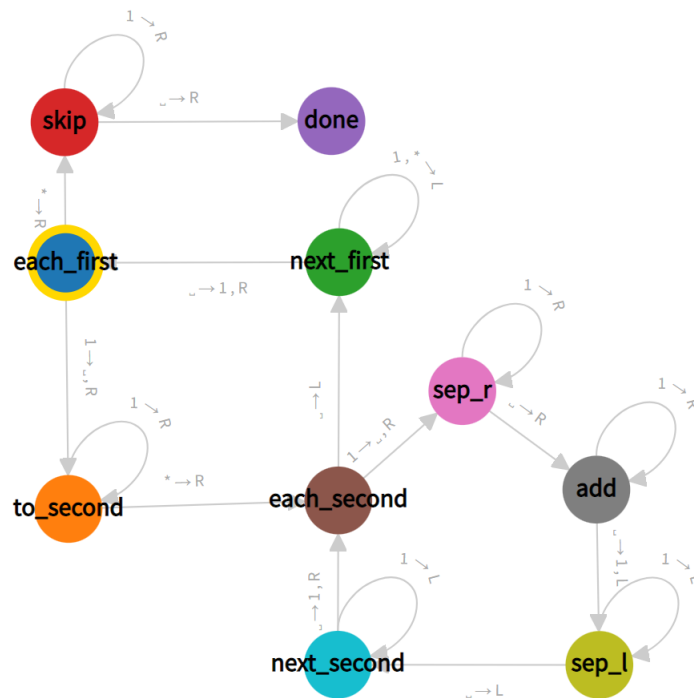
```
input: '111*11'
blank: ' '
start state: each_first
table:
  # Копируем second, first раз, уменьшая first, создавая каждый раз одну копию
  each_first: # Уменьшаем first на единицу
    '1': {write: ' ', R: to_second}
    '*': {R: skip}
```

```

to_second: # Движение к началу second
  '1': R
  '*': {R: each_second} # Копирование
next_first:
  ' ': {write: '1', R: each_first}
  ['1', '*']: L
skip: # Движение к результату
  '1': R
  ' ': {R: done}
done:
  # Копирование second
each_second:
  ' ': {L: next_first}
  '1': {write: ' ', R: sep_r}
sep_r:
  ' ': {R: add}
  '1': R
add:
  ' ': {write: '1', L: sep_l}
  '1': R
sep_l:
  ' ': {L: next_second}
  '1': L
next_second:
  ' ': {write: '1', R: each_second}
  '1': L

```

Визуализация:



## 1.2 Операции с языками и символами

Реализуйте машины Тьюринга, которые позволяют выполнять следующие операции:

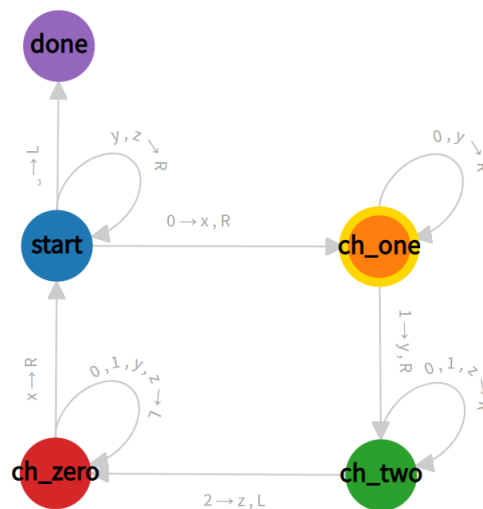
1. Принадлежность языку  $L = \{0^n 1^n 2^n\}, n \geq 0$ .

Реализация: МТ принимает слово, являющееся посл-тью вида «0..1..2» либо пустое слово. Проходясь по слову, МТ заменяет цифры «0,1,2» на «x,y,z» соответственно, что позволяет понять, рассматривалась ли уже данная цифра. Если, по окончании работы МТ, каретка оказалась на пустом символе, то совершается переход в состояние «done», то есть слово принадлежит языку, иначе не принадлежит.

Листинг 2\_1.yaml:

```
input: '001122'
blank: ' '
start state: start
table:
  start:
    '0': {write: 'x', R: ch_one} # 0 на x
    ['y', 'z']: R
    ' ': {L: done}
  ch_one:
    ['0', 'y']: R
    '1': {write: 'y', R: ch_two} # 1 на y
  ch_two:
    ['0', '1', 'z']: R
    '2': {write: 'z', L: ch_zero} # 2 на z
  ch_zero:
    ['0', '1', 'y', 'z']: L
    'x': {R: start}
  done:
```

Визуализация:



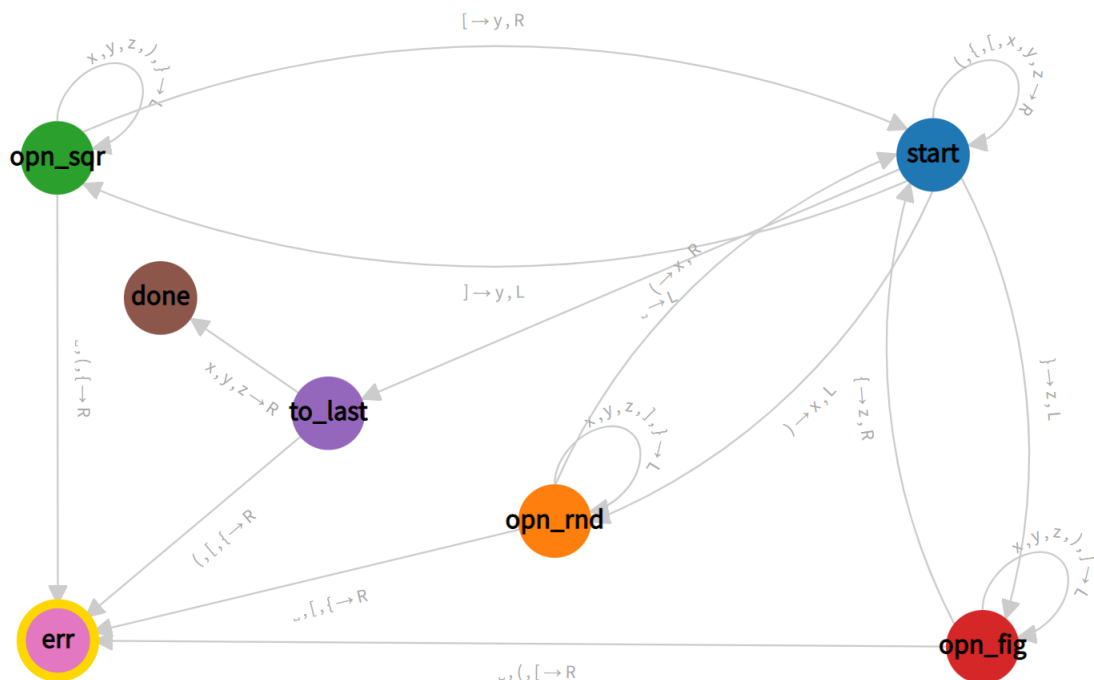
## 2. Проверка соблюдения правильности скобок в строке (минимум 3 вида скобок).

Реализация: МТ принимает произвольную скобочную последовательность из элементов типа «(, {, [, ]». Каретка МТ движется вправо, до первой закрывающей скобки и, при встрече ее, заменяет «(, {, [» на «x, y, z» соответственно. После, осуществляется обратное движение влево, пока не будет найдена, соответствующая открывающая скобка. Если встречена открывающая скобка другого типа или пустой символ, то скобочная последовательность неверная, совершается переход в состояние «err». Алгоритм повторяется. Если, по окончании работы МТ, каретка оказалась на пустом символе, то совершается переход в «done», последовательность была верной.

Листинг 2\_2.yaml:

```
input: '()[]{}(){}'
blank: ' '
start state: start
table:
  start: # Движение вправо, до открытой скобки
    ['(', '[', '{', 'x', 'y', 'z']: R
    ')': {write: 'x', L: opn_rnd}
    ']': {write: 'y', L: opn_sqr}
    '}': {write: 'z', L: opn_fig}
    ' ': {L: to_last}
  opn_rnd: # Ищем открывающую круглую скобку
    [' ', '[', '{']: {R: err}
    ['x', 'y', 'z', ']', '}']: L
    '(': {write: 'x', R: start}
  opn_sqr: # Ищем открывающую квадратную скобку
    [' ', '(', '{']: {R: err}
    ['x', 'y', 'z', ')', '}']: L
    '[': {write: 'y', R: start}
  opn_fig: # Ищем открывающую фигурную скобку
    [' ', '(', '{']: {R: err}
    ['x', 'y', 'z', ')', '}']: L
    '{': {write: 'z', R: start}
  to_last:
    ['(', '[', '{']: {R: err}
    ['x', 'y', 'z']: {R: done}
  done:
  err:
```

Визуализация:



## 2 Квантовые вычисления