



«Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ  
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ

## ОТЧЕТ

по домашнему заданию № 1

Дисциплина: Прикладная теория цифровых автоматов

Название: Игра на мальчиках «Один из двух»

Студент гр. ИУ6-41Б

(Подпись, дата)

И. П. Гунин

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Ю. И. Бауман

(И.О. Фамилия)

Москва, 2022

2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
ОСНОВНАЯ ЧАСТЬ .....	4
Матрица игры .....	4
Спецификация автомата .....	4
Полученный цифровой автомат .....	5
Реализация цифрового автомата «Один из двух» .....	6
Тестирование программы .....	8
ЗАКЛЮЧЕНИЕ .....	11

*Источники*

## ВВЕДЕНИЕ

В настоящей работе выполнена реализация цифрового автомата для игры на пальцах «Один из двух».

Существуют программный и аппаратный способы реализации цифровых автоматов. Программный способ состоит в разработке программы на языке высокого уровня, которая симулирует поведение автомата, а аппаратный предусматривает построение устройств памяти для запоминания текущего состояния автомата.

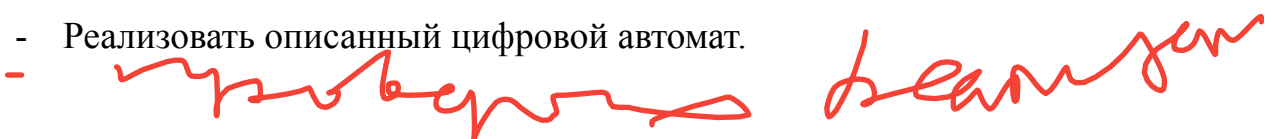
В настоящей работе был выбран программный способ реализации цифрового автомата, так как этот способ предусматривает возможности отладки, тестирования в процессе разработки и расширения функционала в дальнейшем.

Задание (Вариант 15): Каждый из двух игроков показывает другому один или два пальца и одновременно говорит, сколько пальцев покажет партнер. Если оба игрока угадали или не угадали числа пальцев, показанных партнером, то они получают 0 очков. Если угадал один, то он получает 2, 3 или 4 очка – общее число показанных пальцев. (Оптимальная стратегия обоих игроков заключается в следующем: (1,2) следует применять в четырех случаях из семи, (2,1) – в трех из семи, (1,1) и (2,2) – не применять; здесь первая цифра означает число своих показанных пальцев, а вторая – названное число пальцев партнера).

Цель работы - закрепить навыки реализации конечных цифровых автоматов. Для реализации поставленной цели необходимо выполнить следующие задачи.

Задачи:

- Изучить задание в соответствии с вариантом;
- Описать автомат, соответствующий условию задачи;
- Изучить способы реализации цифровых автоматов;
- Выбрать один из способов реализации автоматов;
- Реализовать описанный цифровой автомат.



## ОСНОВНАЯ ЧАСТЬ

### Матрица игры

Проанализируем возможные стратегии игроков, для этого составим матрицу игры (смотреть таблицу 1).

Ходы		Игрок 2			
		1, 1	1, 2	2, 1	2, 2
Игрок 1	1, 1	0, 0	2, 0	0, 3	0, 0
	1, 2	0, 2	0, 0	0, 0	3, 0
	2, 1	3, 0	0, 0	0, 0	0, 4
	2, 2	0, 0	0, 3	4, 0	0, 0

Таблица 1 - Матрица игры «Один из двух»

В парах цифр, выделенных полужирным, первая цифра означает количество пальцев, которые показал соответствующий игрок, а вторая - его предположение, сколько пальцев показал его оппонент. Остальные пары цифр означают количества баллов полученные первым и вторым игроками соответственно.

На основе текста задания и проведенного анализа составим конечный цифровой автомат.

### Спецификация автомата

#### 1. Состояния автомата.

- $q_0$  - начальное состояние автомата;
- $q_1$  - ни один игрок не набирает очков;
- $q_2$  - первый игрок набирает 2 очка;
- $q_3$  - второй игрок набирает 2 очка;
- $q_4$  - первый игрок набирает 3 очка;
- $q_5$  - второй игрок набирает 3 очка;
- $q_6$  - первый игрок набирает 4 очка;
- $q_7$  - второй игрок набирает 4 очка.

## 2. Входные сигналы.

a - оба игрока угадывают или не угадывают количество пальцев;

b - угадывает только первый игрок, общее количество показанных пальцев равно 2;

c - угадывает только второй игрок, общее количество показанных пальцев равно 2;

d - угадывает только первый игрок, общее количество показанных пальцев равно 3;

e - угадывает только второй игрок, общее количество показанных пальцев равно 3;

f - угадывает только первый игрок, общее количество показанных пальцев равно 4;

g - угадывает только второй игрок, общее количество показанных пальцев равно 4;

h - один из игроков показывает или предполагает количество пальцев, не соответствующее условию.

## 3. Выходные сигналы.

0 - невозможно вычислить количество очков, набранных игроками;

1 - ни один игрок не набирает очков;

2 - первый игрок набирает 2 очка;

3 - второй игрок набирает 2 очка;

4 - первый игрок набирает 3 очка;

5 - второй игрок набирает 3 очка;

6 - первый игрок набирает 4 очка;

7 - второй игрок набирает 4 очка.

## Полученный цифровой автомат

Составим таблицу полученного цифрового автомата (смотреть таблицу 2)

Рисунок 1

Состояние	$\delta$								$\lambda$							
	a	b	c	d	e	f	g	h	a	b	c	d	e	f	g	h
$q_0$	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$	$q_6$	$q_7$	$q_0$	1	2	3	4	5	6	7	0
$q_1$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
$q_2$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
$q_3$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
$q_4$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
$q_5$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
$q_6$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
$q_7$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Таблица 2 - Таблица переходов полученного автомата

Теперь представим описанный автомат в виде графа переходов (смотреть рисунок 1).

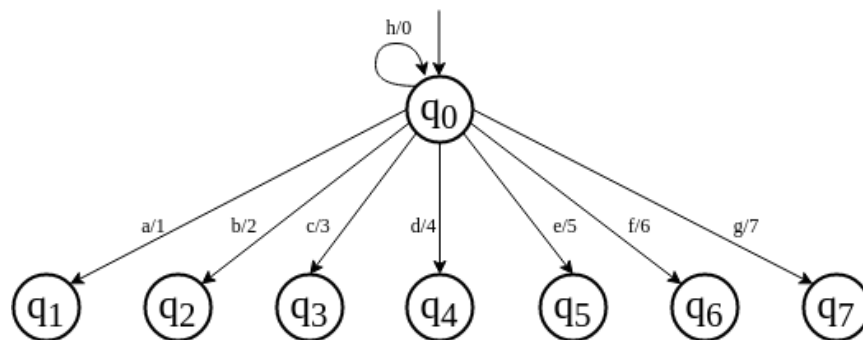


Рисунок 1 - Граф переходов автомата

### Реализация цифрового автомата «Один из двух»

Для реализации цифрового автомата была разработана схема алгоритма (смотреть рисунок 2) и написана программа на языке Ruby.



Рисунок 2 - Схема алгоритма

Код программы:

```
# frozen_string_literal: true

def valid?(player_move)
  player_move.values.all? { |value| [1, 2].include? value }
end

def correct?(first, second)
  first[:assumes] == second[:shows]
end

def sum(first, second)
  first[:shows] + second[:shows]
end

def game_result(first, second)
  {
    first: correct?(first, second) && !correct?(second, first) ? sum(first, second) : 0,
    second: !correct?(first, second) && correct?(second, first) ? sum(first, second) : 0
  }
end

def input
  first = {}
  second = {}
  print 'Количества пальцев, показанные игроками (через пробел): '
  first[:shows], second[:shows] = gets.split.map(&:to_i)
  print 'Количества пальцев, названные игроками (через пробел): '
  first[:assumes], second[:assumes] = gets.split.map(&:to_i)
  [first, second]
end

first, second = input
until valid?(first) && valid?(second)
  puts 'Ошибка: оба игрока должны показать и назвать один или два пальца, повторите ввод'
  first, second = input
end
puts "Количества очков, набранные игроками: #{game_result(first, second).values.map(&:to_s).join ' '}"
```

## Тестирование программы

Протестируем написанную программу (смотреть рисунки 3-6).

Для успешного тестирования необходимо проверить все ветви условий, в которые может переходить программа.



- 1) Введено две пары чисел - количества пальцев, показанные игроками и количества пальцев, ожидаемые игроками от оппонента. Все введенные числа соответствуют условию  $n \in [1, 2] \cap N$ . Предположения обоих игроков совпадают с количествами пальцев, показанных обоими игроками, следовательно, ожидаемый вывод - ни один из игроков не набирает очков (смотреть рисунок 3).
- 2) Введено две пары чисел - количества пальцев, показанные игроками и количества пальцев, ожидаемые игроками от оппонента. Все введенные числа соответствуют условию  $n \in [1, 2] \cap N$ . Предположение первого игрока совпадает с количеством пальцев, показанным вторым игроком, предположение второго игрока ошибочно. Ожидаемый вывод - очки набирает только первый игрок (смотреть рисунок 4).
- 3) Введено две пары чисел - количества пальцев, показанные игроками и количества пальцев, ожидаемые игроками от оппонента. Все введенные числа соответствуют условию  $n \in [1, 2] \cap N$ . Предположение первого игрока ошибочно, предположение второго игрока верно. Ожидаемый вывод - очки набирает только второй игрок (смотреть рисунок 5).
- 4) Введено две пары чисел - количества пальцев, показанные игроками и количества пальцев, ожидаемые игроками от оппонента. Введенные числа не соответствуют условию  $n \in [1, 2] \cap N$ . Программа должна сообщить об ошибке во входных данных и ожидать повторного ввода значений. Во второй раз вводятся значения, соответствующие условию, такие что предположения обоих игроков ошибочны. Ожидаемый вывод - ни один из игроков не набирает очков (смотреть рисунок 6).

```
pyro at sabertooth in ~/P/automata> ruby automaton.rb
Количества пальцев, показанные игроками (через пробел): 2 2
Количества пальцев, названные игроками (через пробел): 2 2
Количества очков, набранные игроками: 0 0
pyro at sabertooth in ~/P/automata> █
```

Рисунок 3 - Оба предположения верны, никто не набирает очки

```
pyro at sabertooth in ~/P/automata> ruby automaton.rb
Количества пальцев, показанные игроками (через пробел): 2 2
Количества пальцев, названные игроками (через пробел): 2 1
Количества очков, набранные игроками: 4 0
pyro at sabertooth in ~/P/automata>
```

Рисунок 4 - Только предположение первого игрока верно, первый игрок набирает очки

```
pyro at sabertooth in ~/P/automata> ruby automaton.rb
Количества пальцев, показанные игроками (через пробел): 2 2
Количества пальцев, названные игроками (через пробел): 1 2
Количества очков, набранные игроками: 0 4
pyro at sabertooth in ~/P/automata>
```

Рисунок 5 - Только предположение второго игрока верно, второй игрок набирает очки

```
pyro at sabertooth in ~/P/automata> ruby automaton.rb
Количества пальцев, показанные игроками (через пробел): 0 0
Количества пальцев, названные игроками (через пробел): 0 0
Ошибка: оба игрока должны показать и назвать один или два пальца, повторите ввод
Количества пальцев, показанные игроками (через пробел): 1 2
Количества пальцев, названные игроками (через пробел): 1 2
Количества очков, набранные игроками: 0 0
pyro at sabertooth in ~/P/automata> █
```

Рисунок 6 - Первый ввод не соответствует условию, во втором оба предположения ошибочны, никто не набирает очки

## ЗАКЛЮЧЕНИЕ

- 1) При выполнении домашнего задания изучен программный способ реализации конечных цифровых автоматов.
- 2) В ходе выполнения данного домашнего задания спроектирован и реализован конечный автомат для игры на пальцах «Один из двух». Создана программная реализация автомата на языке Ruby (для интерпретатора Ruby версии 3.1.2).
- 3) Закреплены навыки подготовки и оформления отчета по проделанной работе с учетом требований ГОСТ 7.32.

Исходники

НЕТ

Омне