



«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 (ИУ6)
НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ
ТЕХНИКА

О Т Ч Е Т

по домашней работе

Название: Задача о 8 ферзях

Дисциплина: Прикладная теория цифровых автоматов

Студент

ИУ6-41Б
(Группа)

02.06.2022

(Подпись, дата)

К. Д. Наумова

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Ю. И. Бауман

(И.О. Фамилия)

Москва, 2022

СОДЕРЖАНИЕ

Введение.....	3
Основная часть.....	4
Спецификация автомата.....	4
Полученный цифровой автомат.....	5
Реализация цифрового автомата.....	6
Тестирование программы.....	8
Заключение.....	9

ВВЕДЕНИЕ

В настоящей работе выполнена реализация цифрового автомата для размещения 8 ферзей на шахматной доске так, чтобы они не нападали друг на друга.

Существуют 2 способа реализации автомата: программный и аппаратный. Программная реализация выполняется на любом языке высокого уровня. Аппаратная реализация – предусматривает построение устройств памяти для запоминания текущего состояния автомата, в роли которых обычно используются триггеры.

В настоящей работе использован программный способ реализации цифрового автомата, так как этот способ подразумевает вариативность реализации, возможность отладки и тестирования в процессе разработки программы. К программам (в отличие от аппаратной реализации цифровых автоматов) можно добавлять новые функции по мере изменения целей, под которые она разрабатывается.

Задание (вариант 19): Разместить 8 ферзей на шахматной доске так, чтобы они не нападали друг на друга. Подробно описать алгоритм решения.

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

Задачи:

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

Основная часть

Спецификация автомата

Состояния автомата:

- q_1 – начальное состояние;
- q_2 – расставлены две фигуры;
- q_3 – расставлены три фигуры;
- q_4 – расставлены четыре фигуры;
- q_5 – расставлены пять фигур;
- q_6 – расставлена шесть фигур;
- q_7 – расставлена семь фигур;
- q_8 – расставлены все фигуры.

Входные сигналы:

- a – позиция для ферзя, удовлетворяющая условиям задачи. То есть, ферзь на данной позиции не пересекается с другими ферзями ни по горизонтали, ни по вертикали, ни по диагоналям.
- a^* – позиция для ферзя, не удовлетворяющая условиям задачи. То есть ферзь на данной позиции пересекается с другими ферзями по горизонтали или по вертикали, или по одной из диагоналей.
- a_n – верная позиция для ферзя в данном состоянии. В таблице переходов нельзя однозначно считать, что введённая позиция удовлетворяет или не удовлетворяет условиям задачи, так как это зависит от текущего состояния. Поэтому я ввела условное обозначение, чтобы можно было построить таблицу переходов. Однако в графе автомата я использую обозначения a/a^* .

Выходные сигналы:

- b_0 – фигура поставлена верно;
- b_1 – фигура поставлена неверно;
- b_2 – все фигуры расставлены;
- b^* – тупик. Ни один из вариантов установки ферзя не подходит. Идёт отход до предыдущего состояния.

Важно заметить, что в начальном состоянии автомата – q_1 – первый ферзь уже стоит на поле a_1 .

Также в каждом состоянии, при выходном сигнале b^* , происходит возврат к предыдущему состоянию, так как возможно, программа пошла решать задачу неверным путём, и произошёл тупик. Это означает, что на предыдущем шаге необходимо сделать другой выбор.

Полученный цифровой автомат

Составим таблицу, описывающую конечный автомат, составленный по условию задачи в результате проведенного анализа (таблица 1).

Таблица 1 - таблица переходов автомата

ВХОД сост	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
q_1	q_1, b_1	q_2, b_0	q_1, b_1	q_1, b_1	q_1, b_1	q_1, b_1	q_1, b_1	q_1, b_1
q_2	q_2, b_1	q_2, b_1	q_3, b_0	q_2, b_1	q_2, b_1	q_2, b_1	q_2, b_1	q_2, b_1
q_3	q_3, b_1	q_3, b_1	q_3, b_1	q_4, b_0	q_3, b_1	q_3, b_1	q_3, b_1	q_3, b_1
q_4	q_4, b_1	q_4, b_1	q_4, b_1	q_4, b_1	q_5, b_0	q_4, b_1	q_4, b_1	q_4, b_1
q_5	q_5, b_1	q_5, b_1	q_5, b_1	q_5, b_1	q_5, b_1	q_6, b_0	q_5, b_1	q_5, b_1
q_6	q_6, b_1	q_6, b_1	q_6, b_1	q_6, b_1	q_6, b_1	q_6, b_1	q_7, b_0	q_6, b_1
q_7	q_7, b_1	q_7, b_1	q_7, b_1	q_7, b_1	q_7, b_1	q_7, b_1	q_7, b_1	q_8, b_2

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

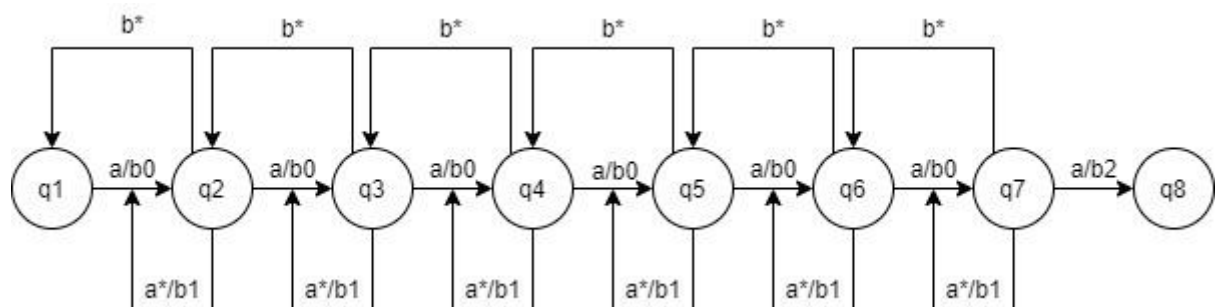


Рисунок 1 – граф переходов цифрового автомата

Реализация цифрового автомата

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

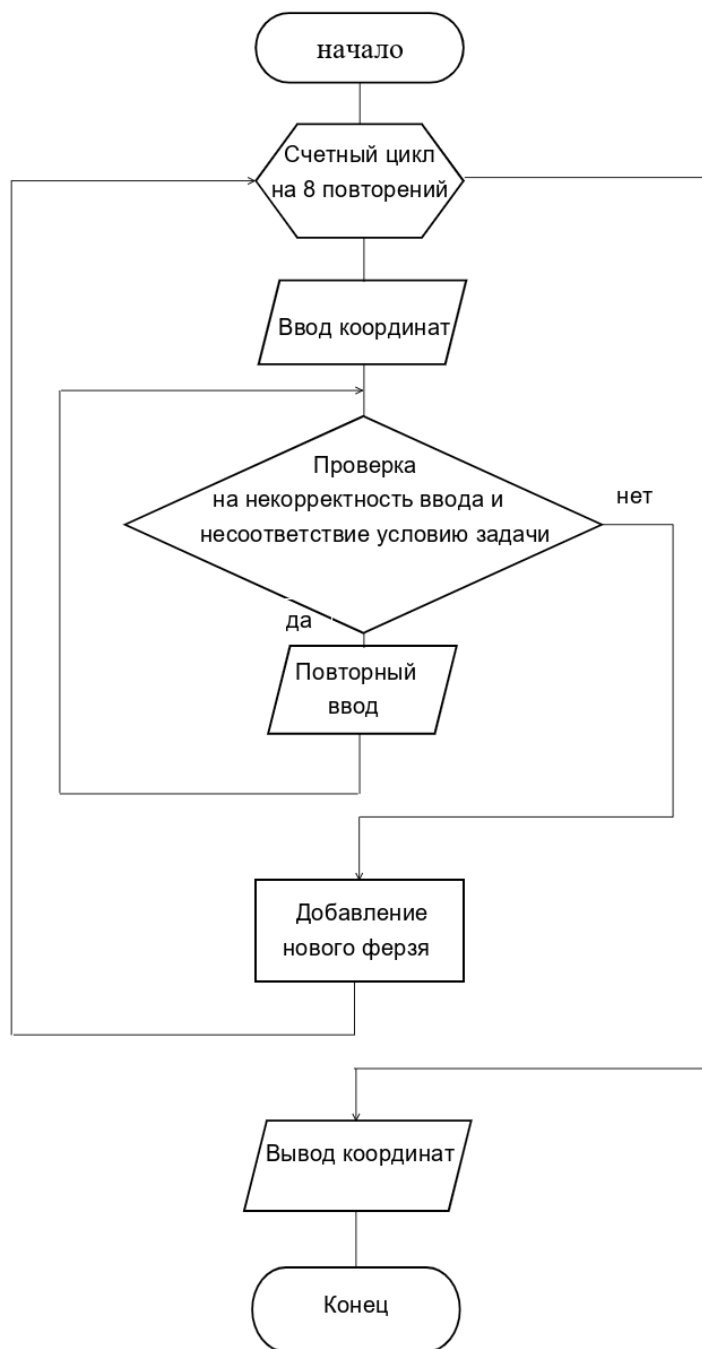


Рисунок 2 – схема алгоритма

Текст программы:

```

1 # frozen_string_literal: true
2
3 def validate_input(queen)
4     queen && (1..8).include?(queen[:x]) && (1..8).include?(queen[:y])
5 end
6
7 def check(queen, new_queen)
8     queen[:x] - new_queen[:x] != 0 &&
9     queen[:y] - new_queen[:y] != 0 &&
10    (queen[:x] - new_queen[:x]).abs != (queen[:y] - new_queen[:y]).abs
11 end
12
13 def input
14     print 'Input two numbers: '
15     gets.split.map(&:to_i).yield_self { |temp| { x: temp[0], y: temp[1] } }
16 end
17
18 queens = []
19
20 8.times do
21     new_queen = input
22     until validate_input(new_queen) && queens.all? { |queen| check(queen, new_queen) }
23         puts 'Wrong coordinates'
24         new_queen = input
25     end
26     queens << new_queen
27     puts 'Queen was added'
28 end
29
30 puts 'All coordinates are correct. Coordinates are:'
31 queens.each { |queen| puts "#{queen[:x]} #{queen[:y]}" } |

```

Рисунок 2 – текст программы на Ruby

Тестирование программы

Протестируем написанную программу (рисунки 3-4):

- 1) введем координаты 8 ферзей такие, чтобы фигуры не нападали друг на друга (рисунок 3);
- 2) попытаемся ввести координаты, нарушающие условие задачи (рисунок 4).

```
ksenia@ksenia-VirtualBox:~/dz$ ruby main.rb
Input two coordinates: 1 1
Queen was added
Input two coordinates: 2 5
Queen was added
Input two coordinates: 3 8
Queen was added
Input two coordinates: 4 6
Queen was added
Input two coordinates: 5 3
Queen was added
Input two coordinates: 6 7
Queen was added
Input two coordinates: 7 2
Queen was added
Input two coordinates: 8 4
Queen was added
All coordinates are correct. Coordinates are:
1 1
2 5
3 8
4 6
5 3
6 7
7 2
8 4
```

Рисунок 3 – Введены верные координаты

```
ksenia@ksenia-VirtualBox:~/dz$ ruby main.rb
Input two coordinates: 1 1
Queen was added
Input two coordinates: 2 5
Queen was added
Input two coordinates: 3 1
Wrong coordinates
Input two coordinates: 3 8
Queen was added
Input two coordinates: 4 6
Queen was added
Input two coordinates: 5 3
Queen was added
Input two coordinates: 6 7
Queen was added
Input two coordinates: 7 2
Queen was added
Input two coordinates: 8 4
Queen was added
All coordinates are correct. Coordinates are:
1 1
2 5
3 8
4 6
5 3
6 7
7 2
8 4
```

Рисунок 4 – Попытка ввести неверные координаты

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 задачи о 8 ферзях. Создана программная реализация автомата на языке Ruby.
- 3) Закреплены навыки подготовки и оформления отчета по проделанной работе с учетом требований ГОСТ 7.32. 2017