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 домашнему заданию

Тема: Конечный автомат, реализующий игру крестики-нолики

Дисциплина: Машинно-зависимые языки и основы компиляции

Студент ИУ6-41Б 01.06.2022 М. А. Кириллов
(Подпись, дата) (И.О. Фамилия)

Преподаватель Ю. И. Бауман
(Подпись, дата) (И.О. Фамилия)

Москва, 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В данной работе представлена реализация конечного автомата «Крестики-нолики».

Для реализации конечного автомата существует 2 способа: аппаратный и программный. Аппаратный способ реализации предусматривает построение устройств памяти для запоминания состояния автомата. Обычно для этого используются триггеры. Программная реализация может быть выполнена на любом языке программирования высокого уровня.

В данной работе используется программный способ реализации конечного автомата. Этот способ дает возможность отладки и тестирования, вариативность реализации.

Цель работы – применив полученные знания, построить конечный автомат в виде таблицы переходов и графа состояний, а также программно реализовать данный автомат и протестировать. Для достижения поставленной цели необходимо выполнить следующие задачи:

- Изучить задание;
- Описать конечный автомат;
- Выбрать один из способов реализации автомата;
- Программно реализовать автомат;
- Протестировать автомат и его состояния.

Задание (вариант 18): игра «Крестики-нолики» для двух игроков через компьютер.

АВТОМАТ КРЕСТИКИ-НОЛИКИ

Спецификация автомата

1) Состояния автомата;
   - S0 – Ожидается ввод крестика;
   - S1 – Ожидается ввод нолика;
   - S2 – Победа крестиков;
   - S3 – Ничья;
   - S4 – Победа ноликов.
2) Входные сигналы;
   - a – Тройка символов не завершена, поле заполнено не полностью;
   - b – Тройка символов не завершена, поле заполнено полностью;
   - c – Тройка символов завершена.
3) Выходные сигналы.
   - 0 – Игра продолжается;
   - 1 – Игра окончена.

Полученный цифровой автомат

После проведения анализа составим таблицу, описывающую составленный конечный автомат (таблица 1).

Таблица 1 – Таблица переходов конечного автомата

Состояние	δ			λ		
	a	b	c	a	b	c
S0	S1	S3	S2	0	1	1
S1	S0	S3	S4	0	1	1
S2	-	-	-	-	-	-
S3	-	-	-	-	-	-
S4	-	-	-	-	-	-

Далее по таблице переходов конечного автомата построим граф состояний (рисунок 1).

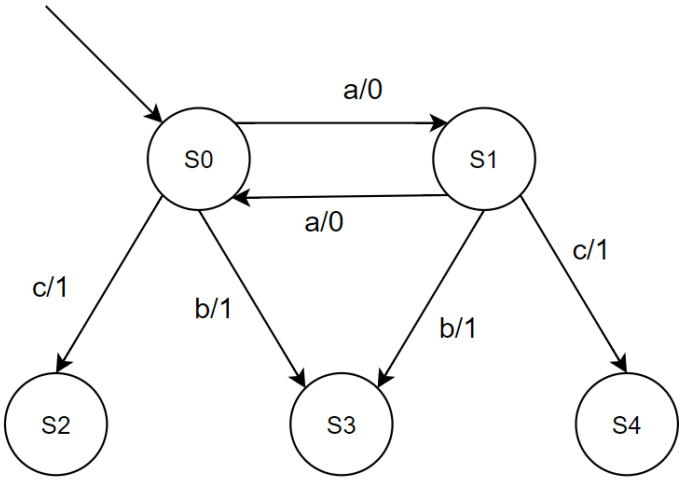

Рисунок 1 – Граф состояний автомата

Реализация конечного автомата

Для реализации автомата разработаем схему алгоритма (рисунок 2) и напишем программу на языке C++.

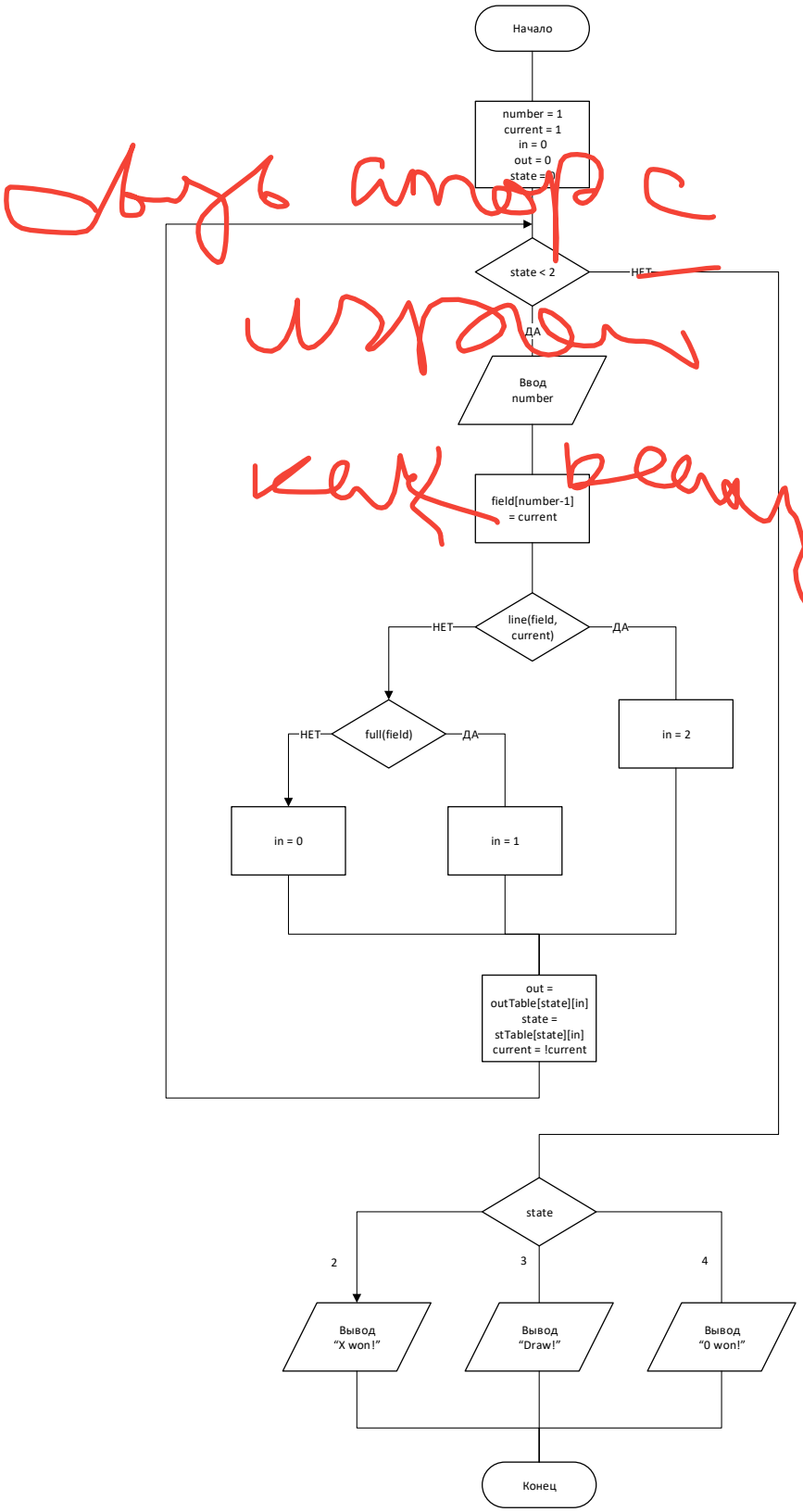

Рисунок 2 – Схема алгоритма

Реализация конечного автомата – программа на языке C++ представлена в Листинге 1.

Листинг 1 – Программная реализация конечного автомата крестики-нолики

```
#include <iostream>

bool line(int* field, int cur) {

     if (field[0] == cur && field[1] == cur && field[2] == cur)
          return true;
     if (field[3] == cur && field[4] == cur && field[5] == cur)
          return true;
     if (field[6] == cur && field[7] == cur && field[8] == cur)
          return true;

     if (field[0] == cur && field[3] == cur && field[6] == cur)
          return true;
     if (field[1] == cur && field[4] == cur && field[7] == cur)
          return true;
     if (field[2] == cur && field[5] == cur && field[8] == cur)
          return true;

     if (field[0] == cur && field[4] == cur && field[8] == cur)
          return true;
     if (field[2] == cur && field[4] == cur && field[6] == cur)
          return true;

     return false;
}

bool full(int* field) {

     for (int i = 0; i < 9; i++)
          if (field[i] < 0)
               return false;

     return true;
}

void draw(int* field) {

     for (int i = 1; i < 10; i++) {
          if (field[i-1] == 1)
               printf("x ");
          else if (field[i-1] == 0)
               printf("o ");
          else
               printf("  ");

          if (!(i % 3))
               printf("\n");
     }
```

```
     printf("\n");
}

int main()
{
     int stTable[2][3] = { {1, 3, 2}, {0, 3, 4} };
     int outTable[2][3] = { {0, 1, 1}, {0, 1, 1} };

     int field[9];
     for (int i = 0; i < 9; i++)
               field[i] = -1;

     printf("Welcome to Tic-Tac-Toe game!\n\n");
     printf("Board with each cage number:\n");
     for (int i = 1; i < 10; i++) {
          printf("%d ", i);
          if (!(i % 3))
               printf("\n");
     }

     int number = 1;
     int current = 1;
     int in = 0;
     int out = 0;
     int state = 0;

     while (state < 2) {

          if (current)
               printf("\nInput cage number to put x: ");
          else
               printf("\nInput cage number to put o: ");

          scanf("%d", &number);

          if (number > 9 || number < 1)
               printf("\nWrong cage number. Try again!\n");
          else if (field[number - 1] >= 0)
               printf("\nCage is already occupied! Try
again!\n");

          else {
               field[number - 1] = current;
               if (line(field, current))
                    in = 2;
               else if (full(field))
                    in = 1;
               else
                    in = 0;

               out = outTable[state][in];
               state = stTable[state][in];
               draw(field);
```

```
                current = !current;
            }
        }

        switch (state) {
            case 2: printf("\nX won!\n"); break;
            case 3: printf("\nDraw!\n"); break;
            case 4: printf("\n0 won!\n"); break;
        }

        return 0;
}
```

Тестирование

Протестируем автомат и его состояния (рисунки 3 – 5). Для этого разделим входные значения на следующие классы эквивалентности:

1) Победа ноликов (входная последовательность a-a-a-a-a-c);
2) Победа крестиков (входная последовательность a-a-a-a-c);
3) Ничья (входная последовательность a-a-a-a-a-a-a-a-b).

```
Welcome to Tic-Tac-Toe game!

Board with each cage number:
1 2 3
4 5 6
7 8 9

Input cage number to put x: 5

  x



Input cage number to put o: 1
o
  x



Input cage number to put x: 3
o   x
  x



Input cage number to put o: 2
o o x
  x



Input cage number to put x: 7
o o x
  x
x


X won!
```

Рисунок 3 – Результат первого теста

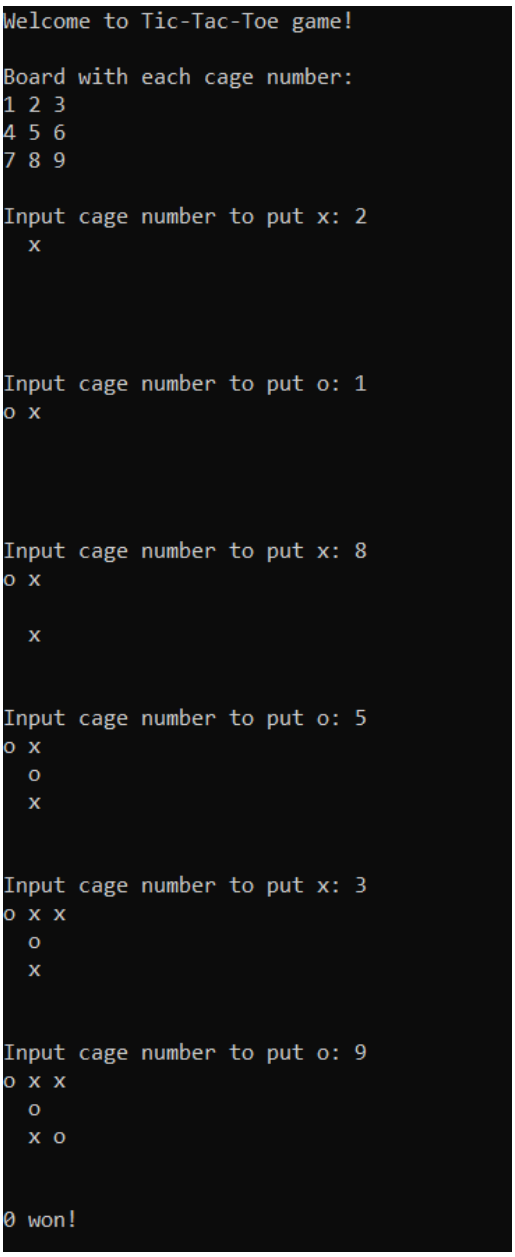

Рисунок 4 – Результат второго теста

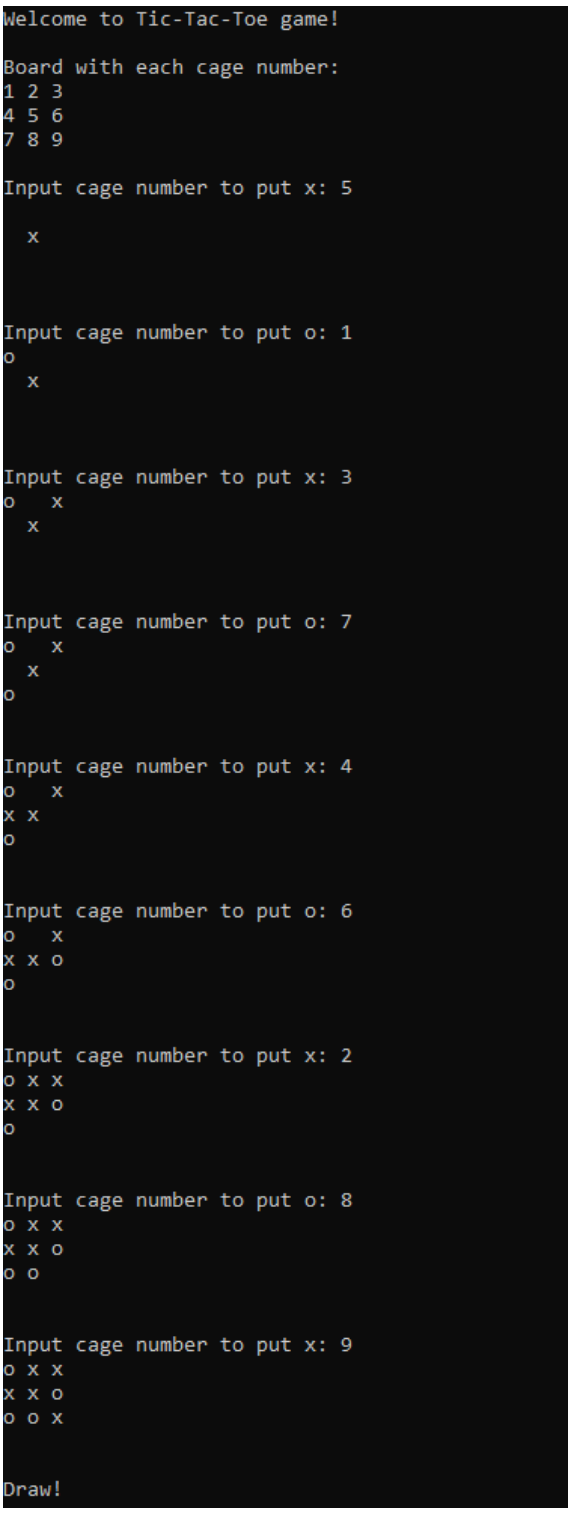

Рисунок 5 – Результат третьего теста

ЗАКЛЮЧЕНИЕ

1) В ходе выполнения домашнего задания изучен способ программной реализации конечных автоматов;
2) При выполнении задания спроектирован и реализован конечный автомат «Крестики-нолики». Разработана программа на языке C++, реализующая созданный автомат;
3) Закреплены навыки подготовки и оформления отчета по проделанной работе с учетом требований ГОСТ 7.32-2017.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. А. А. Городов, А. Л. Мыльников, А. М. Попов Теория автоматов: метод. указания к выполнению курсовых работ. – Красноярск: 2014. -52 с.
2. Ю. Г. Карпов Теория автоматов. – Спб: 2003. -205 с.
3. ГОСТ 7.32-2017 Система стандартов информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе.