

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и кибербезопасности  
Высшая школа программной инженерии

**ЛАБОРАТОРНАЯ РАБОТА №1**  
по дисциплине «Теория автоматов и формальных языков»

Выполнил  
студент гр. 5130904/30108

Ребдев П.А.

Проверил Тышкевич А.И.

Санкт-Петербург  
2026

## **1. Формулировка задания**

Проверить эквивалентность двух заданных конечных автоматов с 4-5 состояниями. Выходной алфавит двоичный. Необходимо предоставить протокол решения с пошаговыми вычислениями и интерпретацию результата.

## 2. Выбранные автоматы

Автомат А (4 состояния)

- Входной алфавит: {0, 1}
- Выходной алфавит: {0, 1}
- Состояния: {0, 1, 2, 3}
- Начальное состояние: 0

Функции переходов ( $\delta$ ) и выходов ( $\lambda$ ):

Состояние	$\delta(0)$	$\delta(1)$	$\lambda(0)$	$\lambda(1)$
0	1	2	0	1
1	3	0	1	0
2	0	3	1	0
3	2	1	0	1

Автомат В (4 состояния, избыточный вариант автомата А)

- Входной алфавит: {0, 1}
- Выходной алфавит: {0, 1}
- Состояния: {A, B, C, D} (в программе: 0,1,2,3)
- Начальное состояние: A (0)

Функции переходов ( $\delta$ ) и выходов ( $\lambda$ ):

Состояние	$\delta(0)$	$\delta(1)$	$\lambda(0)$	$\lambda(1)$
A (0)	C (2)	B (1)	0	1
B (1)	D (3)	A (0)	1	0
C (2)	A (0)	D (3)	1	0
D (3)	B (1)	C (2)	1	0

*Примечание:* Автомат В является избыточным вариантом автомата А, где состояния А и С эквивалентны, а состояния В и D эквивалентны. Ожидается, что автоматы эквивалентны.

### **3. Алгоритм решения**

1. Для проверки эквивалентности используется алгоритм на основе прямого произведения автоматов:
2. Строится прямое произведение двух автоматов
3. Выполняется обход в ширину (BFS) достижимых состояний произведения
4. Для каждого достижимого состояния проверяется совпадение выходов по всем входным символам
5. Если найдено различие - автоматы не эквивалентны
6. Если все достижимые состояния обработаны без различий - автоматы эквивалентны

## 4. Программное решение

Для решения задачи была реализована программа на языке программирования C++. Ввод характеристик автоматов производиться из консоли

main.cpp:

```
#include <iostream>
#include <vector>
#include <queue>

struct Automaton
{
    int statesCount;
    int inputAlphabetSize;
    int outputAlphabetSize;
    int initialState;
    std::vector< std::vector< int > > transition;
    std::vector< std::vector< int > > output;
};

bool areEquivalent(const Automaton & a1, const Automaton & a2)
{
    if (a1.inputAlphabetSize != a2.inputAlphabetSize)
    {
        std::cout << "Входные алфавиты не совпадают!\n";
        return false;
    }
    if (a1.outputAlphabetSize != a2.outputAlphabetSize)
    {
        std::cout << "Выходные алфавиты не совпадают!\n";
        return false;
    }

    int n1 = a1.statesCount;
    int n2 = a2.statesCount;
    int m = a1.inputAlphabetSize;

    std::vector< std::vector< bool > > visited(n1, std::vector< bool >(n2, false));
    std::queue< std::pair< int, int > > q;

    q.push(std::make_pair(a1.initialState, a2.initialState));
    visited[a1.initialState][a2.initialState] = true;

    int step = 0;
    while (!q.empty())
    {
        auto current = q.front();
        int s1 = current.first;
        int s2 = current.second;
        q.pop();
        ++step;

        std::cout << "Шаг " << step << ": состояние (" << s1 << ", " << s2 << ")\n";

        for (int a = 0; a < m; ++a)
        {
            int out1 = a1.output[s1][a];
            int out2 = a2.output[s2][a];
```

```

    std::cout << " Вход " << a << ": выходы " << out1 << " и " << out2;

    if (out1 != out2)
    {
        std::cout << " -> РАЗЛИЧАЮТСЯ!\n";
        std::cout << "Автоматы не эквивалентны.\n";
        return false;
    }
    std::cout << " -> совпадают.\n";

    int next1 = a1.transition[s1][a];
    int next2 = a2.transition[s2][a];

    if (!visited[next1][next2])
    {
        visited[next1][next2] = true;
        q.push(std::make_pair(next1, next2));
        std::cout << " Добавлено состояние (" << next1 << ", " << next2 << ")\n";
    }
}
}

std::cout << "Все достижимые состояния обработаны, различий не найдено.\n";
return true;
}

int main()
{
    Automaton a1, a2;

    std::cout << "==== Ввод автомата A ====\n";
    std::cout << "Количество состояний: ";
    std::cin >> a1.statesCount;
    std::cout << "Размер входного алфавита: ";
    std::cin >> a1.inputAlphabetSize;
    std::cout << "Размер выходного алфавита: ";
    std::cin >> a1.outputAlphabetSize;
    std::cout << "Начальное состояние: ";
    std::cin >> a1.initialState;

    a1.transition.resize(a1.statesCount, std::vector< int >(a1.inputAlphabetSize));
    a1.output.resize(a1.statesCount, std::vector< int >(a1.inputAlphabetSize));

    std::cout << "Таблица переходов (по строкам состояния, для каждого входа 0, 1, ...):\n";
    for (int i = 0; i < a1.statesCount; ++i)
    {
        for (int j = 0; j < a1.inputAlphabetSize; ++j)
        {
            std::cin >> a1.transition[i][j];
        }
    }

    std::cout << "Таблица выходов (по строкам состояния, для каждого входа 0, 1, ...):\n";
    for (int i = 0; i < a1.statesCount; ++i)
    {
        for (int j = 0; j < a1.inputAlphabetSize; ++j)
        {
            std::cin >> a1.output[i][j];
        }
    }
}

```

```

std::cout << "\n==== Ввод автомата B ====\n";
std::cout << "Количество состояний: ";
std::cin >> a2.statesCount;
std::cout << "Размер входного алфавита: ";
std::cin >> a2.inputAlphabetSize;
std::cout << "Размер выходного алфавита: ";
std::cin >> a2.outputAlphabetSize;
std::cout << "Начальное состояние: ";
std::cin >> a2.initialState;

a2.transition.resize(a2.statesCount, std::vector< int >(a2.inputAlphabetSize));
a2.output.resize(a2.statesCount, std::vector< int >(a2.inputAlphabetSize));

std::cout << "Таблица переходов:\n";
for (int i = 0; i < a2.statesCount; ++i)
{
    for (int j = 0; j < a2.inputAlphabetSize; ++j)
    {
        std::cin >> a2.transition[i][j];
    }
}

std::cout << "Таблица выходов:\n";
for (int i = 0; i < a2.statesCount; ++i)
{
    for (int j = 0; j < a2.inputAlphabetSize; ++j)
    {
        std::cin >> a2.output[i][j];
    }
}

std::cout << "\n==== Проверка эквивалентности ====\n";
if (areEquivalent(a1, a2))
{
    std::cout << "Результат: автоматы ЭКВИВАЛЕНТНЫ.\n";
}
else
{
    std::cout << "Результат: автоматы НЕ ЭКВИВАЛЕНТНЫ.\n";
}

return 0;
}

```

## 5. Протокол выполнения

```
==== Автомат 1 ====
Состояний: 4
Входных символов: 2
Выходных символов: 2
Начальное состояние: 0
Таблица переходов:
1 2
3 0
0 3
2 1
Таблица выходов:
0 1
1 0
1 0|
0 1

==== Автомат 2 ====
Состояний: 4
Входных символов: 2
Выходных символов: 2
Начальное состояние: 0
Таблица переходов:
2 1
3 0
0 3
1 2
Таблица выходов:
0 1
1 0
1 0
0 1

==== Проверка эквивалентности ====
Шаг 1: состояние (0, 0)
    Вход 0: выходы 0 и 0 -> Совпадают.
    Добавлено состояние (1, 2)
    Вход 1: выходы 1 и 1 -> Совпадают.
    Добавлено состояние (2, 1)
Шаг 2: состояние (1, 2)
    Вход 0: выходы 1 и 1 -> Совпадают.
    Добавлено состояние (3, 0)
    Вход 1: выходы 0 и 0 -> Совпадают.
    Добавлено состояние (0, 3)
Шаг 3: состояние (2, 1)
    Вход 0: выходы 1 и 1 -> Совпадают.
    Вход 1: выходы 0 и 0 -> Совпадают.
Шаг 4: состояние (3, 0)
    Вход 0: выходы 0 и 0 -> Совпадают.
    Добавлено состояние (2, 2)
    Вход 1: выходы 1 и 1 -> Совпадают.
    Добавлено состояние (1, 1)
Шаг 5: состояние (0, 3)
    Вход 0: выходы 0 и 0 -> Совпадают.
    Вход 1: выходы 1 и 1 -> Совпадают.
Шаг 6: состояние (2, 2)
    Вход 0: выходы 1 и 1 -> Совпадают.
    Вход 1: выходы 0 и 0 -> Совпадают.
    Добавлено состояние (3, 3)
Шаг 7: состояние (1, 1)
    Вход 0: выходы 1 и 1 -> Совпадают.
    Вход 1: выходы 0 и 0 -> Совпадают.
Шаг 8: состояние (3, 3)
    Вход 0: выходы 0 и 0 -> Совпадают.
    Вход 1: выходы 1 и 1 -> Совпадают.
Все состояния обработаны, различий не найдено.
Автоматы эквивалентны.
```

## **6. Интерпретация результата**

Автоматы эквивалентны. Это означает, что для любого входного слова оба автомата выдают одинаковые выходные последовательности. Проверка всех достижимых состояний прямого произведения не выявила различий в выходных функциях.

Достижимые состояния прямого произведения:  $(0,0)$ ,  $(1,2)$ ,  $(2,1)$ ,  $(3,0)$ ,  $(0,3)$ ,  $(2,2)$ ,  $(1,1)$ ,  $(3,3)$ . Во всех этих состояниях выходы по всем входным символам совпадают.

## **7. Выводы**

1. Разработана программа на C++ для проверки эквивалентности конечных автоматов
2. Алгоритм основан на построении прямого произведения и обходе достижимых состояний
3. Для выбранных автоматов с 4 состояниями каждый подтвержден эквивалентность
4. Программа предоставляет подробный протокол проверки с пошаговым описанием
5. Метод проверки через прямое произведение является корректным и полным для детерминированных конечных автоматов

Программа может быть использована для проверки эквивалентности любых детерминированных конечных автоматов с совпадающими входными и выходными алфавитами.