



«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
НАПРАВЛЕНИЕ ПОДГОТОВКИ: 09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

ОТЧЕТ

по домашнему заданию № 1

Дисциплина: Прикладная теория цифровых автоматов

Название: Эквивалентность автоматов

Студент гр. ИУ6-44Б

(Подпись, дата)

А.А. Куценко
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Ю.И. Бауман
(И.О. Фамилия)

Москва, 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
ОСНОВНАЯ ЧАСТЬ.....	4
Описание алгоритма проверки автоматов на эквивалентность	4
Спецификация автомата	4
Полученный цифровой автомат.....	4
Реализация цифрового автомата.....	6
Тестирование программы.....	11
ЗАКЛЮЧЕНИЕ	12

ВВЕДЕНИЕ

В настоящей работе выполнена реализация цифрового автомата для проверки эквивалентности двух автоматов.

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 5): Эквивалентность автоматов.

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

Описание алгоритма проверки автоматов на эквивалентность

Алгоритм определения того, эквивалентны два исходных автомата или нет, включает в себя 3 шага:

- строится прямое произведение (тоже автомат) этих автоматов;
- из полученного автомата удаляются недостижимые состояния;
- к «тому, что осталось» применяется теорема Мура.

Спецификация автомата

1. Состояния автомата.

- q_0 - начальное состояние автомата;
- q_1 - построена таблица автомата $A*B$;
- q_2 - автоматы эквиваленты;
- q_4 - автоматы не эквиваленты.

2. Входные сигналы.

- a - хотя бы для одного автомата введены некорректные данные;
- b - все введенные данные корректны;
- c - в таблице переходов автомата $A*B$ все пары выходных сигналов одноименные;
- d - в таблице переходов автомата $A*B$ есть разноименные пары выходных сигналов.

3. Выходные сигналы.

- 0 - невозможно построить таблицу переходов автомата $A*B$;
- 1 - построена таблица переходов автомата $A*B$;
- 2 - автоматы A и B эквивалентны;
- 3 - автоматы A и B не эквивалентны.

Полученный цифровой автомат

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

Таблица 1 - таблица переходов автомата “Один из трех”

Состояние	δ				λ			
	a	b	c	d	a	b	c	d
q₀	q ₀	q ₁	-	-	0	1	-	-
q₁	-	-	q ₂	q ₃	-	-	2	3
q₂	-	-	-	-	-	-	-	-
q₃	-	-	-	-	-	-	-	-

Представим описанный автомат в виде графа переходов (рисунок 1).

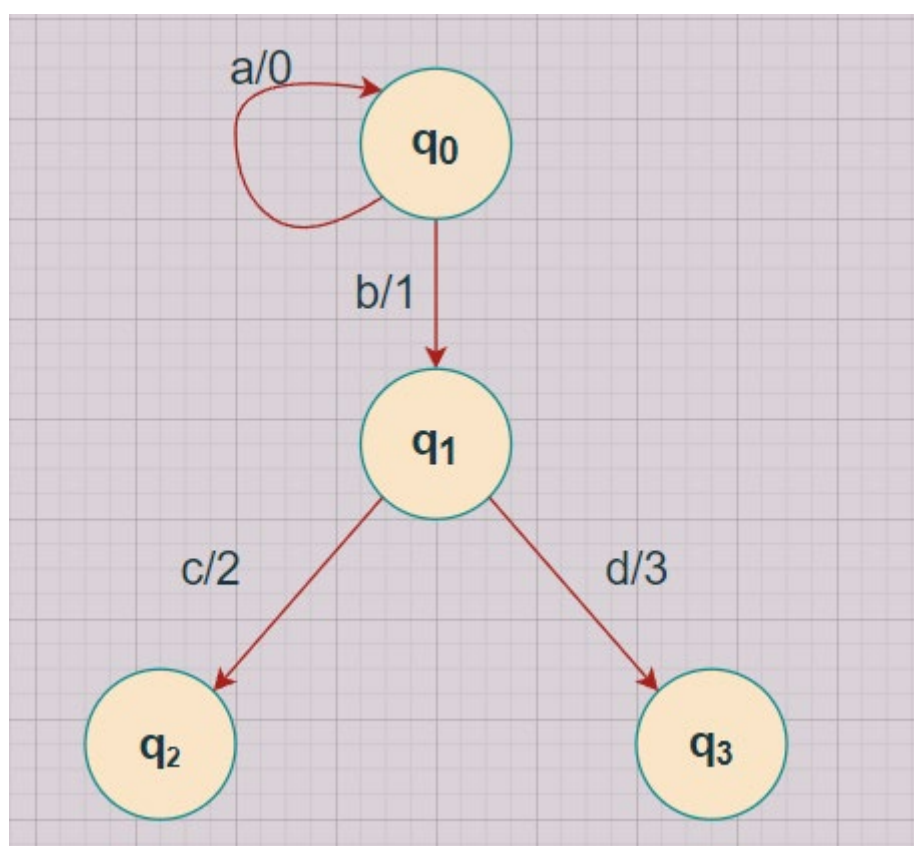


Рисунок 1 - граф переходов цифрового автомата

Реализация цифрового автомата

Для реализации описанного цифрового автомата была разработана схема алгоритма (рисунки 2 - 4) и написана программа на языке Pascal.

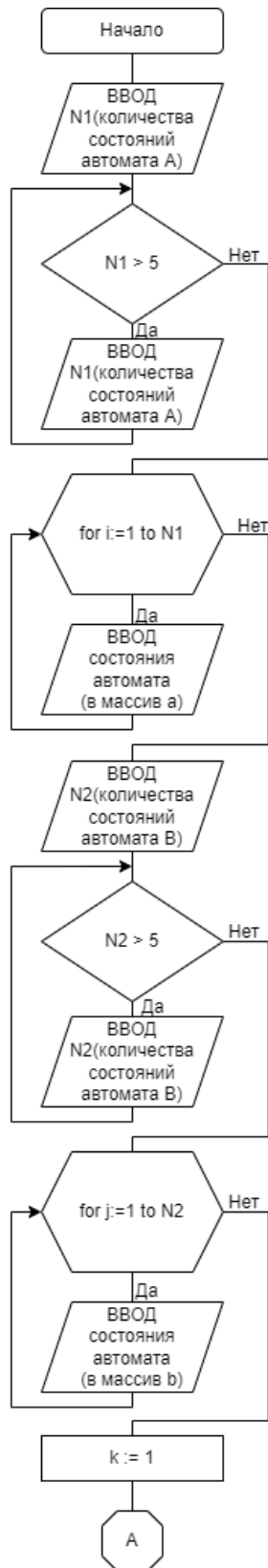


Рисунок 2 - схема алгоритма

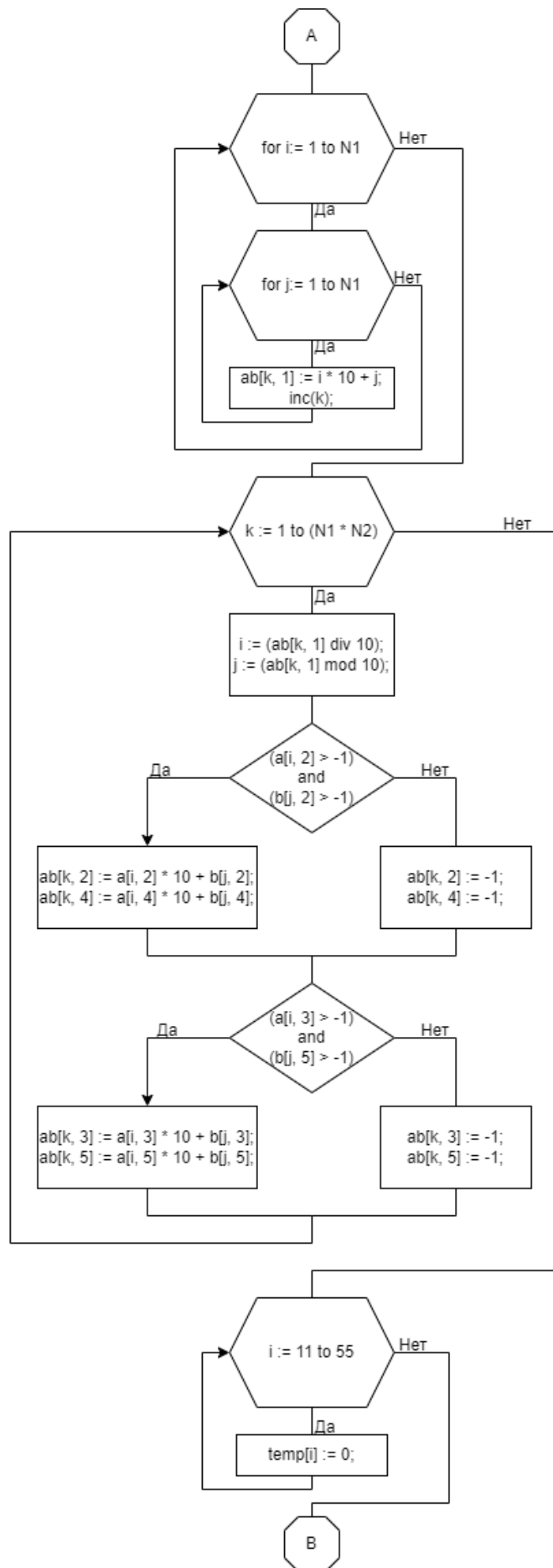


Рисунок 3 – схема алгоритма (продолжение)

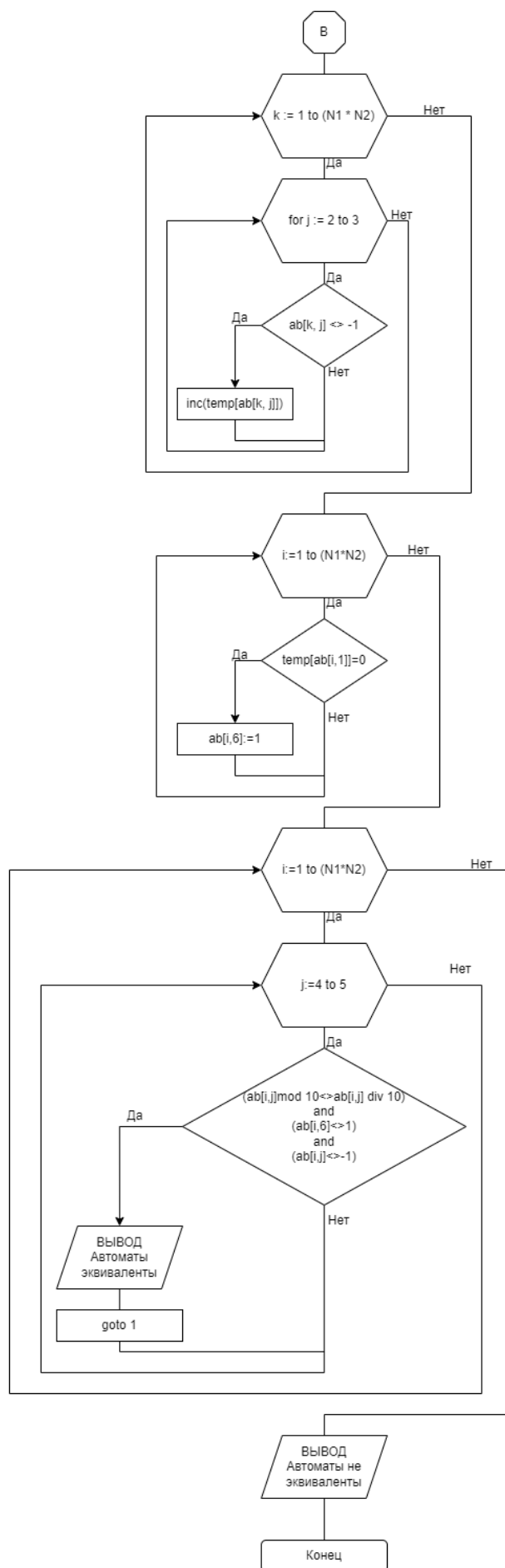


Рисунок 4 – схема алгоритма (окончание)

Текст программы представлен в листинге 1.

Листинг 1 – исходный код программы

```
//в программе рассматриваются автоматы не более чем с 5
состояниями
//состояния имеют номера от 1 до (максимум) 5
var
  a: array[1..5, 1..5] of integer;
  b: array[1..5, 1..5] of integer;
  i, N1, N2, j, k: integer;
  //этот массив нужен для записи результатов построения прямого
произведения автоматов
  ab: array[1..25, 1..6] of integer;
  //вспомогательный массив для удаления недостижимых вершин
  temp: array[11..55] of integer;

label 1, 2;
begin
  writeln('Введите количество состояний исходного автомата A');
  readln(N1);
  while N1 > 5 do
    begin
      writeln('Автомат должен содержать не больше 10 состояний.
Введите количество состояний исходного автомата');
      readln(N1)
    end;
    writeln('если автомат не имеет данного перехода, заполните
значение ячейки таблицы числом -1');
    writeln('Введите для автомата A строки в формате: <переход
автомата для сигнала a><переход автомата для сигнала b><выходной
сигнал для сигнала a><выходной сигнал для сигнала b> для каждого
состояния автомата ');
    for i := 1 to N1 do
      begin
        a[i, 1] := i;
        readln(a[i, 2], a[i, 3], a[i, 4], a[i, 5]);
      end;
    writeln('Введите количество состояний исходного автомата B');
    readln(N2);
    while N2 > 5 do
      begin
        writeln('Автомат должен содержать не больше 10 состояний.
Введите количество состояний исходного автомата');
        readln(N2)
      end;
      writeln('если автомат не имеет данного перехода, заполните
значение ячейки таблицы числом -1');
      writeln('Введите для автомата B строки в формате: <переход
автомата для сигнала a><переход автомата для сигнала b><выходной
сигнал для сигнала a><выходной сигнал для сигнала b> для каждого
состояния автомата ');
      for j := 1 to N2 do
        begin
```

```

    b[j, 1] := j;
    readln(b[j, 2], b[j, 3], b[j, 4], b[j, 5]);
end;
//построим таблицу автомата - прямое произведение A*B
k := 1;
for i := 1 to N1 do
begin
    for j := 1 to N2 do
    begin
        ab[k, 1] := i * 10 + j;
        inc(k);
    end;
end;

for k := 1 to (N1 * N2) do
begin
    i := (ab[k, 1] div 10);
    j := (ab[k, 1] mod 10);
    if (a[i, 2] > -1) and (b[j, 2] > -1) then begin
        ab[k, 2] := a[i, 2] * 10 + b[j, 2];
        ab[k, 4] := a[i, 4] * 10 + b[j, 4];
    end
    else begin
        ab[k, 2] := -1;
        ab[k, 4] := -1;
    end;
    if (a[i, 3] > -1) and (b[j, 5] > -1) then begin
        ab[k, 3] := a[i, 3] * 10 + b[j, 3];
        ab[k, 5] := a[i, 5] * 10 + b[j, 5];
    end
    else begin
        ab[k, 3] := -1;
        ab[k, 5] := -1;
    end;
end;
end;
// удаляем недостижимые вершины
for i := 11 to 55 do
    temp[i] := 0;
for k := 1 to (N1 * N2) do
begin
    for j := 2 to 3 do
    begin
        if ab[k, j] <> -1 then
            inc(temp[ab[k, j]]);
    end;
end;
end.

```

Тестирование программы

Протестируем написанную программу (рисунки 5 - 6).

```
Введите количество состояний исходного автомата A
1
если автомат не имеет данного перехода, заполните значение ячейки таблицы числом -1
Введите для автомата A строки в формате: <переход автомата для сигнала а><переход автомата для сигнала б><выходной сигнал для сигнала а><выходной сигнал для сигнала б> для
каждого состояния автомата
1 -1 1 -1
Введите количество состояний исходного автомата B
2
если автомат не имеет данного перехода, заполните значение ячейки таблицы числом -1
Введите для автомата B строки в формате: <переход автомата для сигнала а><переход автомата для сигнала б><выходной сигнал для сигнала а><выходной сигнал для сигнала б> для
каждого состояния автомата
-1 1 -1 1
2 1 1 2
автоматы эквивалентны
```

Рисунок 5 — результаты верны (автоматы эквивалентны, т.к.
единственная достижимая вершина автомата $A*B$ имеет выходной сигнал
1,1)

```
Введите количество состояний исходного автомата A
1
если автомат не имеет данного перехода, заполните значение ячейки таблицы числом -1
Введите для автомата A строки в формате: <переход автомата для сигнала а><переход автомата для сигнала б><выходной сигнал для сигнала а><выходной сигнал для сигнала б> для
каждого состояния автомата
1 -1 1 -1
Введите количество состояний исходного автомата B
2
если автомат не имеет данного перехода, заполните значение ячейки таблицы числом -1
Введите для автомата B строки в формате: <переход автомата для сигнала а><переход автомата для сигнала б><выходной сигнал для сигнала а><выходной сигнал для сигнала б> для
каждого состояния автомата
-1 1 -1 1
2 1 2 2
автоматы не эквивалентны
```

Рисунок 6 — результаты верны (автоматы неэквивалентны, т.к.
единственная достижимая вершина автомата $A*B$ имеет выходной сигнал
1,2)

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 проверки эквивалентности автоматов. Создана программная реализация автомата в среде разработки Lazarus 1.8.2 на языке Pascal.
- 3) Закреплены навыки подготовки и оформления отчета по проделанной работе с учетом требований ГОСТ 7.32. 