

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

Уфимский университет науки и технологий

Отчёт по лабораторной работе № 2

По дисциплине «Теория автоматов»

Тема: построение дискретно-детерминированных моделей

Вариант: 6

Выполнил ст. гр. ИВТ-ПО-301Б

Кумушбаева Г. А.

Газизова Д. Ф.

Гиндуллин И. С.

Нигматуллин Д. Р.

Проверил:

Сибагатуллин Р. Р.

Уфа 2025

## Содержание

<b>Введение:</b> .....	3
<b>Цель работы:</b> .....	3
<b>Ход работы:</b> .....	4
<b>Вывод:</b> .....	4

## **Введение:**

Конечные автоматы (F-схемы) являются фундаментальными моделями в теории автоматов и используются для описания поведения дискретных систем. Они находят применение в проектировании цифровых устройств, синтаксическом анализе, моделировании процессов управления и других областях. В данной работе рассматривается построение и анализ дискретно-детерминированной модели в виде автомата Мили второго рода.

## **Цель работы:**

Получить практические навыки построения дискретно-детерминированных моделей (конечных автоматов), их анализа и проверки на адекватность с использованием табличного, графического и аналитического способов задания.

## Ход работы:

Исходные данные:

Тип автомата: Мили второго рода

Входной алфавит ( $n=2$ ):  $X = \{x_1, x_2\}$

$x_1$  — Погладить

$x_2$  — Наступить на хвост

Выходной алфавит ( $m=3$ ):  $Y = \{y_1, y_2, y_3\}$

$y_1$  — Стоит

$y_2$  — Виляет хвостом

$y_3$  — Гавкает

Алфавит состояний ( $p=3$ ):  $Z = \{z_1, z_2, z_3\}$

$z_1$  — Нейтральная

$z_2$  — Довольная

$z_3$  — Злая

Начальное состояние:  $z_1$  (Нейтральная)

Входное слово для теста:  $W = (x_1, x_2, x_1)$

Таблица 1. Функция переходов  $\varphi(z_i, x_i)$

Текущее состояние \ Вход	$x_1$ (Погладить)	$x_2$ (Наступить)
$z_1$ (Нейтральная)	$z_2$	$z_3$
$z_2$ (Довольная)	$z_2$	$z_3$
$z_3$ (Злая)	$z_1$	$z_3$

Таблица 2. Функция выходов  $\psi(z_{i+1}, x_i)$

След. состояние \ Вход	$x_1$ (Погладил)	$x_2$ (Наступил)
$z_1$ (Нейтральная)	$y_2$	$y_3$
$z_2$ (Довольная)	$y_2$	$y_3$
$z_3$ (Злая)	$y_1$	$y_3$

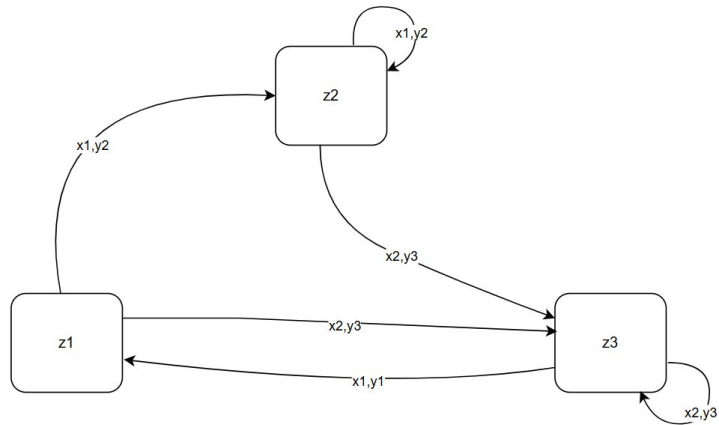


Рисунок 1 – Графический способ задания  $F$ - автомата

### Логические уравнения для состояний:

$$z1 = x1z3$$

$$z2 = x1z1 \vee x1z2$$

$$z3 = x2z1 \vee x2z2 \vee x2z3$$

### Программная реализация:

%% Лабораторная работа №2. Автомат Мили (Поведение собаки)

% Вариант 6

% Выполнили: Кумушбаева Г. А., Газизова Д. Ф., Гиндуллин И. С., Нигматуллин Д. Р.

clear; clc; close all;

%% Исходные данные автомата Мили 2-го рода

% Алфавит состояний  $Z = \{z1, z2, z3\}$

states = {'z1 (Нейтральная)', 'z2 (Довольная)', 'z3 (Злая)'};

num\_states = length(states);

% Входной алфавит  $X = \{x1, x2\}$

inputs = {'x1 (Погладить)', 'x2 (Наступить)'};

num\_inputs = length(inputs);

```
% Выходной алфавит Y = {y1, y2, y3}
```

```
outputs = {'y1 (Стоит)', 'y2 (Виляет хвостом)', 'y3 (Гавкает)'};
```

```
num_outputs = length(outputs);
```

```
% Функция переходов (next state) - Таблица 1
```

```
% transitions(current_state, input) = next_state
```

```
% Индексация: z1=1, z2=2, z3=3; x1=1, x2=2
```

```
transition_table = [
```

```
    2, 3; % z1: x1->z2, x2->z3
```

```
    2, 3; % z2: x1->z2, x2->z3
```

```
    1, 3; % z3: x1->z1, x2->z3
```

```
];
```

```
% Функция выходов (output) - Таблица 2
```

```
% output_table(next_state, input) = output
```

```
% Индексация состояний и входов как выше, выходов: y1=1, y2=2, y3=3
```

```
output_table = [
```

```
    2, 3; % Если след. состояние z1: после x1->y2, после x2->y3
```

```
    2, 3; % Если след. состояние z2: после x1->y2, после x2->y3
```

```
    1, 3; % Если след. состояние z3: после x1->y1, после x2->y3
```

```
];
```

```
%% Моделирование работы автомата для входного слова W
```

```
% W = (x1, x2, x1)
```

```
input_sequence = {'x1', 'x2', 'x1'};
```

```
% Начальное состояние
```

```
current_state = 1; % z1 (Нейтральная)
```

```
% Массивы для хранения истории (для построения графиков)
```

```
time_steps = 0:length(input_sequence); % Временные такты (0 - начальное состояние)
```

```
state_history = zeros(1, length(time_steps)); % История состояний
```

```
output_history = cell(1, length(time_steps)); % История выходных сигналов
```

```

% Инициализация

state_history(1) = current_state;

output_history{1} = 'Начальное состояние';


fprintf('Проверка автомата Мили на адекватность\n');
fprintf('Начальное состояние: %s\n\n', states{current_state});


% Проход по входному слову
for step = 1:length(input_sequence)

    % Определяем код входного сигнала
    current_input = input_sequence{step};

    if strcmp(current_input, 'x1')
        input_code = 1;
    elseif strcmp(current_input, 'x2')
        input_code = 2;
    else
        error('Неизвестный входной сигнал: %s', current_input);
    end

    % Определяем следующее состояние по функции переходов
    next_state = transition_table(current_state, input_code);

    % Определяем выходной сигнал по функции выходов
    % Для автомата Мили выход зависит от текущего состояния и входа
    % Но у вас задана таблица  $\psi(z_{next}, x)$ , что соответствует 2-му роду
    current_output = output_table(next_state, input_code);

    % Логика для вывода: для автомата Мили выход формируется на переходе
    % У нас уже есть  $z_i$  (current_state) и  $x_i$ , мы вычислили  $z_{i+1}$  (next_state) и  $y_i$ 
    fprintf('Шаг %d:\n', step);
    fprintf(' Текущее состояние: %s\n', states{current_state});
    fprintf(' Входной сигнал:  %s\n', inputs{input_code});

```

```

fprintf(' Выходной сигнал: %s\n', outputs{current_output});

fprintf(' След. состояние: %s\n\n', states{next_state});


% Сохраняем данные для графика
% Выходной сигнал y_i соответствует переходу по входу x_i на шаге i
% На графике состояния отображаются в конце такта
state_history(step + 1) = next_state;
output_history{step + 1} = outputs{current_output};


% Переходим к следующему состоянию
current_state = next_state;
end


%% Построение временных диаграмм (графиков)


figure('Position', [100, 100, 900, 600]);
subplot(2,1,1);


% График состояний
stairs(time_steps, state_history, 'LineWidth', 2, 'Marker', 'o');
grid on;
title('Временная диаграмма: Изменение состояний автомата', 'FontSize', 14);
xlabel('Такт (время)', 'FontSize', 12);
ylabel('Состояние', 'FontSize', 12);
yticks(1:num_states);
yticklabels(states);
xlim([0, length(input_sequence)]);
xticks(time_steps);


% Добавление подписей выходов на график состояний
for i = 1:length(time_steps)
    if i == 1
        text(time_steps(i), state_history(i), ' Нач.', 'VerticalAlignment', 'bottom');
    end
end

```



```

else

    % Подписываем выход, который привел к этому состоянию в предыдущем такте

    text(time_steps(i), state_history(i), [' ' output_history{i}], 'VerticalAlignment', 'bottom', 'FontSize', 8);

end

end

subplot(2,1,2);

% График выходных сигналов (в числовом виде для наглядности)

output_numeric = zeros(1, length(output_history));

for i = 1:length(output_history)

    switch output_history{i}

        case 'y1 (Стоит)'

            output_numeric(i) = 1;

        case 'y2 (Виляет хвостом)'

            output_numeric(i) = 2;

        case 'y3 (Гавкает)'

            output_numeric(i) = 3;

        otherwise

            output_numeric(i) = 0; % Начальное состояние

    end

end

stairs(time_steps, output_numeric, 'r', 'LineWidth', 2, 'Marker', 's');

grid on;

title('Временная диаграмма: Выходные сигналы автомата', 'FontSize', 14);

xlabel('Такт (время)', 'FontSize', 12);

ylabel('Выходной сигнал', 'FontSize', 12);

yticks(0:3);

yticklabels({'Нач.', 'y1 (Стоит)', 'y2 (Виляет)', 'y3 (Гавкает)'});

xlim([0, length(input_sequence)]);

xticks(time_steps);

% Общий вывод

```

```

fprintf('\n--- Итоговые последовательности ---\n');

fprintf('Входное слово W: ');

fprintf('%s ', input_sequence{:});

fprintf('\n');

fprintf('Последовательность состояний: ');

state_seq = arrayfun(@(x) states{x}, state_history, 'UniformOutput', false);

fprintf('%s ', state_seq{:});

fprintf('\n');

fprintf('Выходная последовательность: ');

% Выходная последовательность - это выходы на каждом шаге перехода

output_seq = output_history(2:end); % Игнорируем начальное "состояние" выхода

fprintf('%s ', output_seq{:});

fprintf('\n');

```

## Результат:

```

Проверка автомата Мили на адекватность
Начальное состояние: z1 (Нейтральная)

Шаг 1:
Текущее состояние: z1 (Нейтральная)
Входной сигнал:    x1 (Погладить)
Выходной сигнал:   y2 (Виляет хвостом)
След. состояние:   z2 (Довольная)

Шаг 2:
Текущее состояние: z2 (Довольная)
Входной сигнал:    x2 (Наступить)
Выходной сигнал:   y3 (Гавкает)
След. состояние:   z3 (Злая)

Шаг 3:
Текущее состояние: z3 (Злая)
Входной сигнал:    x1 (Погладить)
Выходной сигнал:   y2 (Виляет хвостом)
След. состояние:   z1 (Нейтральная)

--- Итоговые последовательности ---
Входное слово W:  x1 x2 x1
Последовательность состояний: z1 (Нейтральная) z2 (Довольная) z3 (Злая) z1 (Нейтральная)
Выходная последовательность: y2 (Виляет хвостом) y3 (Гавкает) y2 (Виляет хвостом)
>>

```

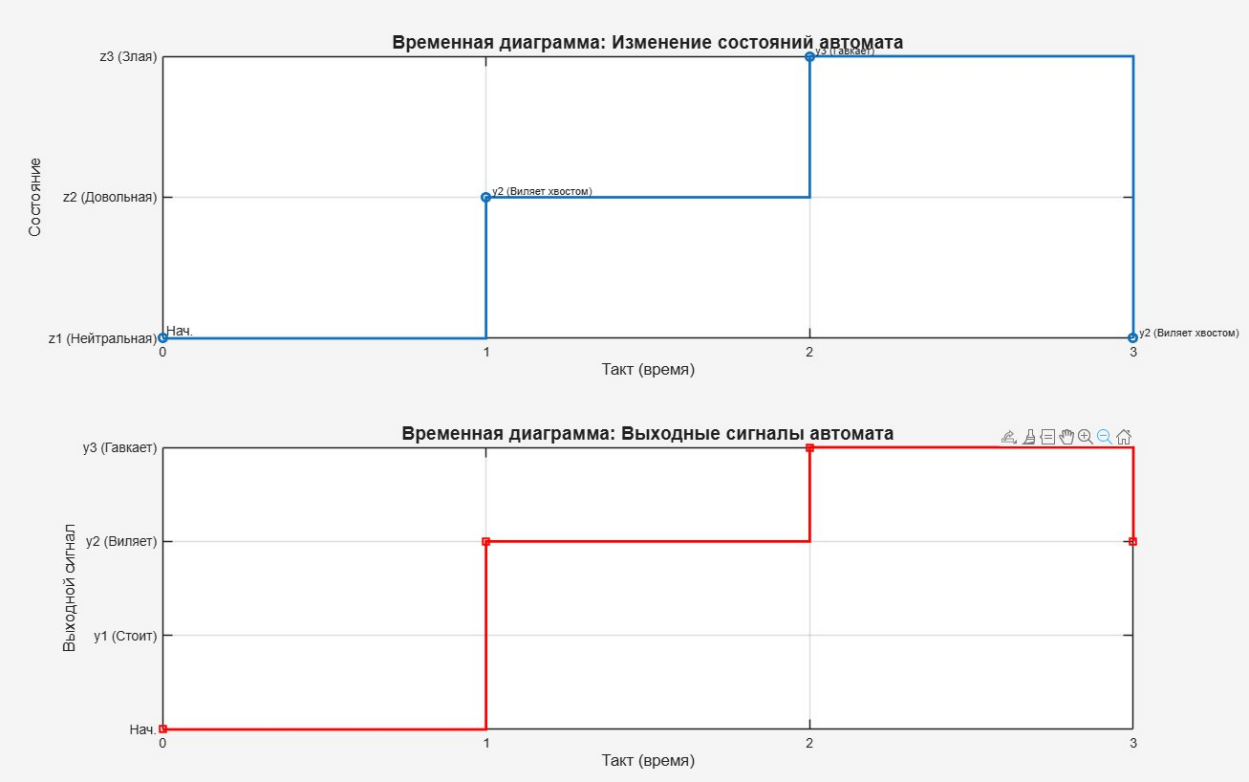
Рисунок 2 – Результат программы

## Проверка на адекватность:

Таблица 3 – Цепочка переходов и выходов автомата

Шаг	Текущее состояние	Вход	Следующее состояние	Выход	Логика поведения
-----	-------------------	------	---------------------	-------	------------------

1	z1 (Нейтральная)	x1 (Погладить)	z2 (Довольная)	y2 (Виляет хвостом)	Собака довольна, виляет хвостом
2	z2 (Довольная)	x2 (Наступить на хвост)	z3 (Злая)	y3 (Гавкает)	Собака рассержена, гавкает
3	z3 (Злая)	x1 (Погладить)	z1 (Нейтральная)	y2 (Виляет хвостом)	Собака успокоилась, виляет хвостом



## **Вывод:**

Был разработан F-автомат Мили второго рода для моделирования поведения собаки в ответ на два входных сигнала: «Погладить» и «Наступить на хвост». Были определены три состояния: нейтральная, довольная и злая, а также три выхода: «Стоит», «Виляет хвостом», «Гавкает». Начальное состояние автомата выбрано как нейтральное.

Функция переходов  $\varphi(z, x)$  и функция выходов  $\psi(z\_next, x)$  были оформлены в виде таблиц. Также построен граф состояний, где вершины отражают состояния собаки, а дуги — переходы между ними в зависимости от входного сигнала с указанием соответствующего выхода. Такая структурная схема позволяет наглядно проследить динамику работы автомата.

Для проверки был использован входной набор  $W = (x1, x2, x1)$ . Последовательность работы автомата показала логичное поведение собаки. Выходная последовательность  $(y2, y3, y2)$  полностью соответствует ожидаемой реакции собаки, что подтверждает адекватность модели.

Таким образом, разработанная модель полностью соответствует требованиям F-автомата Мили второго рода и может использоваться для дальнейшего анализа и визуализации поведения объекта.