Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Уфимский университет науки и технологий

Отчет по лабораторной работе №3

На тему «Построение дискретно-детерминированных моделей»
По дисциплине «Теория автоматов»

Вариант 1

Выполнил: ст. группы ИВТ-331Б

Еранов И. А.

Зарипов Л. Ф.

Морозов Д. М.

Аюпов А. Б.

Шарипов Р. Ш.

Проверил: доцент каф. ВТиЗИ

Сибагатуллин Р. Р.

Уфа 2024

Цель работы:

Получить представление о дискретно-детерминированных моделей, построении их динамических характеристик, а также проверке их на адекватность.

Теоретическое введение:

Дискретно-детерминированные модели называются также конечными автоматами (англ. finite automat), или F-схемами. F-схемы характеризуются шестью элементами: конечным множеством X входных сигналов (входным алфавитом); конечным множеством Y выходных сигналов (выходным алфавитом); конечным множеством Z внутренних состояний (внутренним алфавитом или алфавитом состояний); начальным состоянием автомата; функцией переходов φ(z, x); функцией выходов ψ(z, x). Работа конечного автомата происходит по следующей схеме: в каждом i-м такте на вход автомата, находящегося в состоянии zi подается некоторый сигнал xi на который он реагирует переходом в новое состояние zi+1 и выдачей некоторого выходного сигнала yi. Существует автомат Мура и Мили.

Исходные данные:

Тип автомата и размерность входного алфавита xn, выходного алфавита ym и состояний zp выбрать согласно варианту:

1. Автомат Мили

№ п/п	Тип автомата	n	m	p
5	Мили (1-го рода)	3	3	3

автомат Мили первого рода

$$\begin{cases} z_{i+1} = \varphi(z_i, x_i), \\ y_i = \psi(z_i, x_i), i = 0,1,2,...; \end{cases}$$

Рассмотрим автомат Мили, предназначенный для управления дорожным светофором.

Элементы автомата Мили

- Входные **сигналы (2 сигнала):**
  - `z0`: Таймер зеленого сигнала кончился.
  - `z1`: Таймер желтого сигнала кончился.
  - `z2`: Таймер красного сигнала кончился.

- Выходные **сигналы (2 сигнала):**
    - y0: Смена на красный сигнал.
    - y1: Смена на желтый сигнал.
    - y2: Смена на зеленый сигнал.
- Состояния **(2 состояния):**
    - a0: Горит красный сигнал.
    - a1: Горит желтый сигнал.
    - a2: Горит зеленый сигнал.

Представим автомат Мили в табличном виде:

Таблица переходов:

Состояния Входы	a0	a1	a2
z0	a0	a0	a0
z1	a1	a1	a1
z2	a2	a2	a2

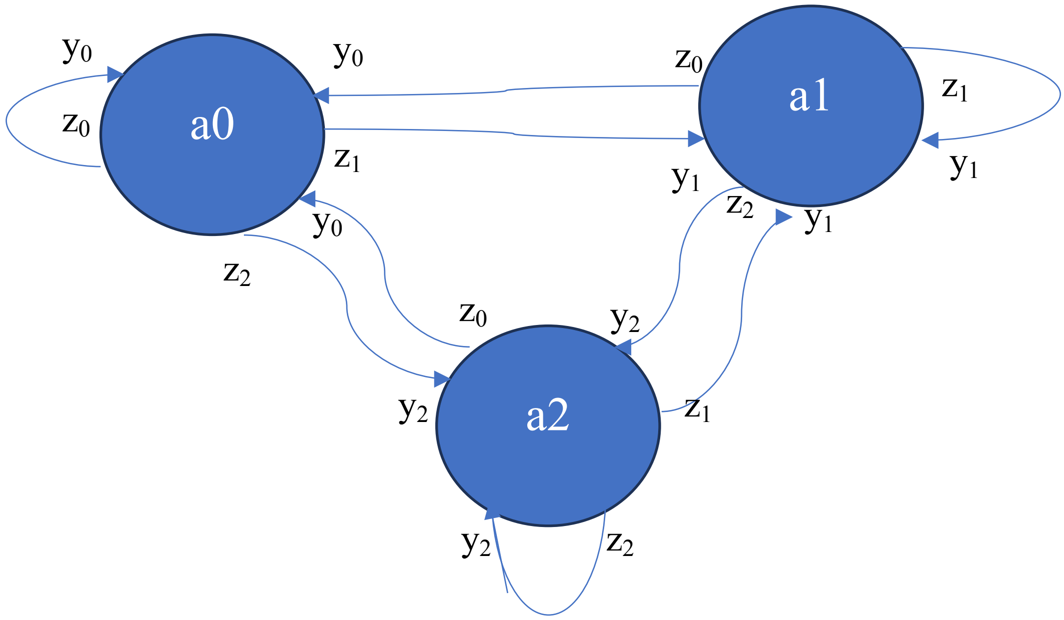
Таблица выходов:

Состояния Входы	y0	y1	y2
z0	y0	y0	y0
z1	y1	y1	y1
z2	y2	y2	y2

Совмещенная таблица переходов и выходов автомата Мили.

	a_0	a_1	a_2
z_0	$\frac{a_0}{y_0}$	$\frac{a_0}{y_0}$	$\frac{a_0}{y_0}$
z_1	$\frac{a_1}{y_1}$	$\frac{a_1}{y_1}$	$\frac{a_1}{y_1}$
z_2	$\frac{a_2}{y_2}$	$\frac{a_2}{y_2}$	$\frac{a_2}{y_2}$

Представим автомат Мили в графическом виде:



Модель автомата Мили:

$a_0 = a_0 z_0 \vee a_1 z_0 \vee a_2 z_0$

$a_1 = a_0 z_1 \vee a_1 z_1 \vee a_2 z_1$

$a_2 = a_0 z_2 \vee a_1 z_2 \vee a_2 z_2$

Проверка модели на адекватность (реакция автомата Мили на цепочке входных символов):

Код:

```
import matplotlib.pyplot as plt



states = ["a0 (Красный)", "a1 (Желтый)", "a2 (Зеленый)"]
inputs = ["z0", "z1", "z2"]
transitions = {
    "a0": {"z0": "a1", "z1": "a1", "z2": "a1"},
    "a1": {"z0": "a2", "z1": "a2", "z2": "a0"},
    "a2": {"z0": "a0", "z1": "a0", "z2": "a1"},
}
outputs = {"a0": "y0 (Красный)", "a1": "y1 (Желтый)", "a2": "y2 (Зеленый)"}
```

```
test_inputs = ["z0", "z1", "z2", "z2", "z0"]


current_state = "a0"  # начальное состояние
states_over_time = [current_state]  # для построения графика

print("Проверка автомата Мура на адекватность")
print(f"Начальное состояние: {current_state}")

for signal in test_inputs:
    next_state = transitions[current_state][signal]
    output = outputs[current_state]
    print(f"Вход: {signal} | Состояние: {current_state} | Выход: {output} | Следующее состояние: {next_state}")
    current_state = next_state
    states_over_time.append(current_state)

time = list(range(len(states_over_time)))

state_to_num = {state: i + 1 for i, state in enumerate(states)}
numeric_states = [state_to_num[state] for state in states_over_time]

plt.figure(figsize=(10, 6))
plt.step(time, numeric_states, where="post", label="Состояние автомата")
plt.xticks(time)
plt.yticks(range(1, len(states) + 1), states)
plt.xlabel("Время")
```

```
plt.ylabel("Состояние")
plt.title("Изменение состояния автомата Мура по времени")
plt.legend()
plt.grid(True)
plt.show()
```

Результаты:

Проверка автомата Мура на адекватность

Начальное состояние: a0

Вход: z0 | Состояние: a0 | Вход: y0 (Красный) | Следующее состояние: a1

Вход: z1 | Состояние: a1 | Выход: y1 (Желтый) | Следующее состояние: a2

Вход: z2 | Состояние: a2 | Выход: y2 (Зеленый) | Следующее состояние: a1

Вход: z2 | Состояние: a1 | Выход: y1 (Желтый) | Следующее состояние: a0

Вход: z0 | Состояние: a0 | Выход: y0 (Красный) | Следующее состояние: a1

2. Автомат Мура:

№ п/п	Тип автомата	n	m	p
3	Мура (1-го рода)	4	2	4

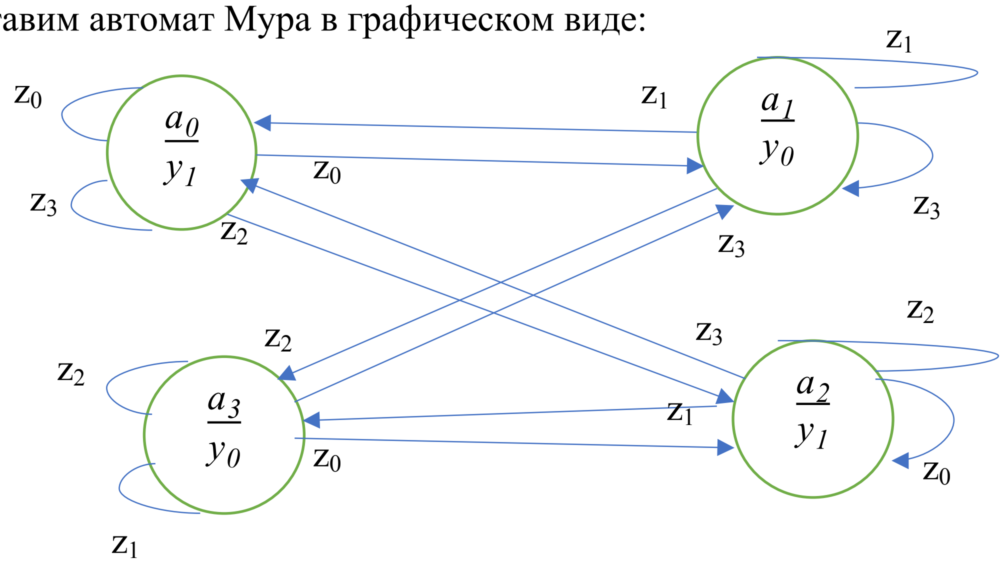
автомат Мура первого рода

$$\begin{cases} z_{i+1} = \varphi(z_i, x_i), \\ y_i = \psi(z_i), i = 0,1,2,\ldots \end{cases}$$

Представим автомат Мура в табличном виде:

	y_1	y_0	y_1	y0
	a_0	a_1	a_2	a_3
z_0	a_0	a_0	a_2	a_2
z_1	a_1	a_1	a_3	a_3
z_2	a_2	a_3	a_2	a_3
z_3	a_0	a_1	a_0	a_1

Представим автомат Мура в графическом виде:



Модель автомата Мура:

$a_0 = a_0z_0 \vee a_0z_3 \vee a_1z_1 \vee a_2z_3$

$a_1 = a_1z_1 \vee a_1z_3 \vee a_0z_0 \vee a_3z_3$

$a_2 = a_2z_2 \vee a_2z_0 \vee a_3z_0 \vee a_0z_2$

$a_3 = a_3z_1 \vee a_3z_2 \vee a_1z_2 \vee a_2z_1$

Проверка модели на адекватность (реакция автомата Мура на цепочке входных символов):

Код программы:

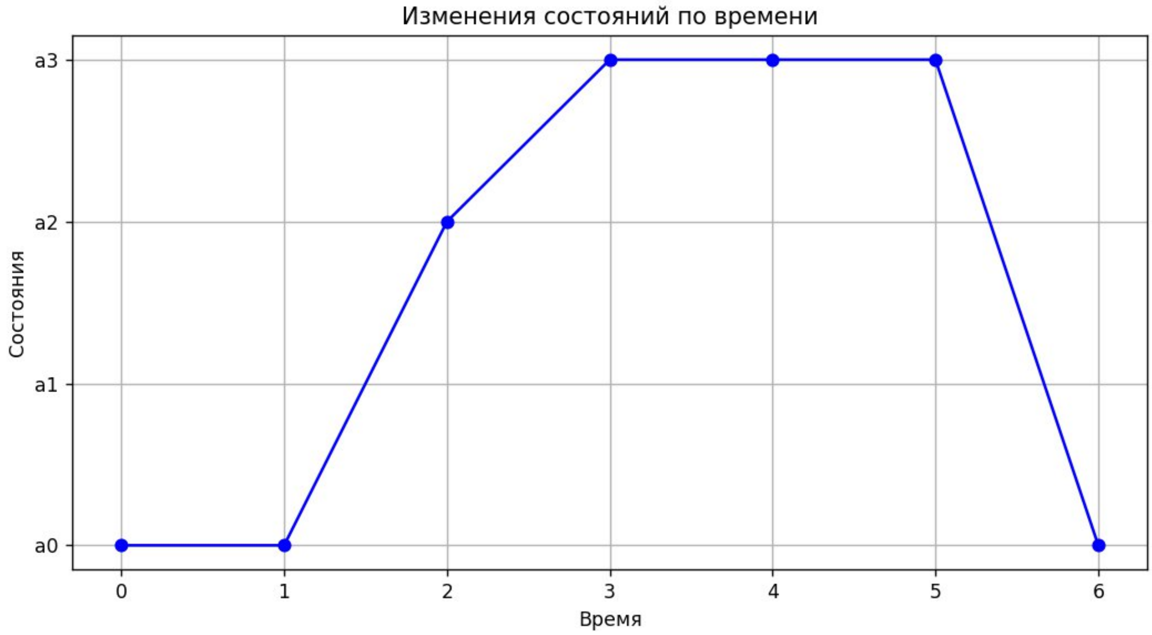
```
import matplotlib.pyplot as plt


z = 0
T = [
    [0, 0, 2, 2],
    [1, 1, 3, 3],
    [2, 2, 2, 3],
    [3, 3, 3, 0]
]
```

```
output = [0, 1, 0, 1]
states = ['a0', 'a1', 'a2', 'a3']
input_sequence = ['z0', 'z1', 'z2', 'z3', 'z0', 'z2', 'z3']

def plot_state_changes(input_sequence):
    current_state = 0
    time_steps = []
    state_sequence = []

    time_step = 0
    for signal in input_sequence:
        if signal == 'z0':
            z = 0
        elif signal == 'z1':
            z = 1
        elif signal == 'z2':
            z = 2
        elif signal == 'z3':
            z = 3
        else:
            print(f"Неизвестный сигнал: {signal}")
            continue

        current_state = T[current_state][z]
        time_steps.append(time_step)
        state_sequence.append(current_state)
        time_step += 1

    # Визуализация
```

```python
    plt.figure(figsize=(10, 5))
    plt.plot(time_steps, state_sequence, marker='o', linestyle='-', color='b')
    plt.yticks([0, 1, 2, 3], states)  # Подписи для состояний
    plt.xticks(time_steps)  # Подписи для времени
    plt.xlabel("Время")
    plt.ylabel("Состояния")
    plt.title("Изменения состояний по времени")
    plt.grid(True)
    plt.show()

plot_state_changes(input_sequence)
```

Результаты:

```
Начальное состояние: a0
Входной сигнал: z0, Состояние: a0, Выходной сигнал: y0
Входной сигнал: z1, Состояние: a0, Выходной сигнал: y0
Входной сигнал: z2, Состояние: a2, Выходной сигнал: y0
Входной сигнал: z3, Состояние: a3, Выходной сигнал: y0
Входной сигнал: z0, Состояние: a3, Выходной сигнал: y1
Входной сигнал: z2, Состояние: a3, Выходной сигнал: y1
Входной сигнал: z3, Состояние: a0, Выходной сигнал: y1
```



Вывод: в ходе лабораторной работы было получено представление о дискретно-детерминированных моделей, построении их динамических характеристик, а также проверке их на адекватность.