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:**

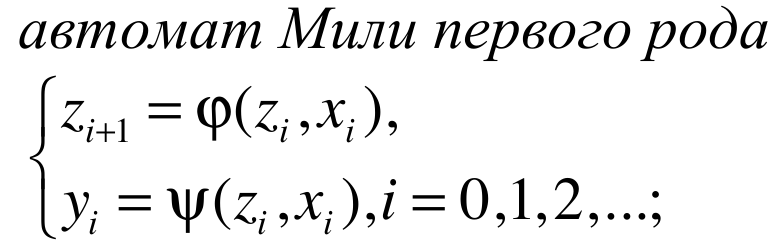
Дискретно-детерминированные модели называются также конечными автоматами (англ. finite automat), или F-схемами. F-схемы характеризуются шестью элементами: конечным множеством Х входных сигналов (входным алфавитом); конечным множеством Y выходных сигналов (выходным алфавитом); конечным множеством Z внутренних 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 хi 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 |



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.
  + y​2: 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.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Представим автомат Мили в графическом виде:

y0

z0

y0

a1

z0

a0

z1

z1

y1

y1

y0

y1

z2

z2

z0

y2

a2

y2

z1

y2

z2

Модель автомата Мили:

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

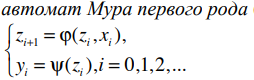
Вход: z2 | Состояние: a2 | Выход: y2 (Зеленый) | Следующее состояние: a1

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 |



Представим автомат Мура в табличном виде:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | y1 | y0 | y1 | y0 |
| a0 | a1 | a2 | a3 |
| z0 | a0 | a0 | a2 | a2 |
| z1 | a1 | a1 | a3 | a3 |
| z2 | a2 | a3 | a2 | a3 |
| z3 | a0 | a1 | a0 | a1 |

Представим автомат Мура в графическом виде:

z0

z1

z1

z3

z3

z0

z2

z3

z2

z2

z2

z3

z1

z0

z0

z1

Модель автомата Мура:

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*

    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")

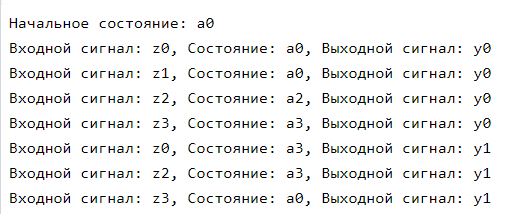
    plt.title("Изменения состояний по времени")

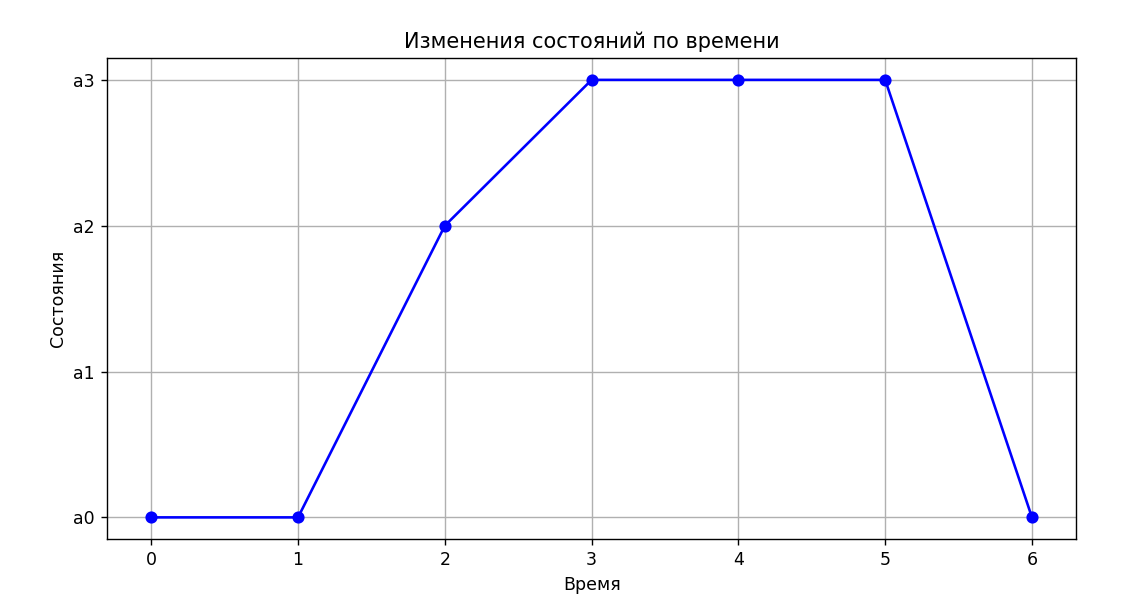
    plt.grid(**True**)

    plt.show()

plot\_state\_changes(input\_sequence)

Результаты:





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