Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Уфимский университет науки и технологий

Отчет по лабораторной работе 2

## На тему «Построение дискретно-детерминированных моделей»

По дисциплине «Теория автоматов»

Вариант 3

Выполнил: ст. группы ИВТ-331Б

Имамутдинов Алмаз

Шайхлисламов Валерий

Терегулов Тимур

Исхаков Рафаэль

Денисов Тимофей

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:**

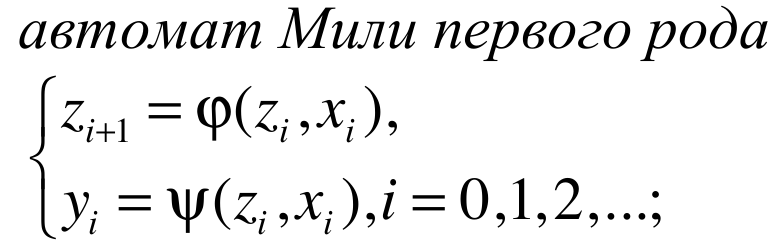
Дискретно-детерминированные модели называются также конечными автоматами (англ. finite automat), или F-схемами. F-схемы характеризуются шестью элементами: конечным множеством Х входных сигналов (входным алфавитом); конечным множеством Y выходных сигналов (выходным алфавитом); конечным множеством Z внутренних 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 |
| 9 | Мили (1-го рода) | 2 | 2 | 2 |



Рассмотрим автомат Мили, предназначенный для управления пешеходным светофором.

Элементы автомата Мили

* Входные **сигналы (2 сигнала):**
  + **z0**​: Таймер зеленого сигнала кончился.
  + **z1**​: Таймер красного сигнала кончился.
* Выходные **сигналы (2 сигнала):**
  + y0​: Включить красный сигнал.
  + y1: Включить зеленый сигнал.
* Состояния **(2 состояния):**
  + a0​: Горит красный сигнал.
  + a1​: Горит зеленый сигнал.

Представим автомат Мили в табличном виде:

Таблица переходов:

|  |  |  |
| --- | --- | --- |
| **Состояния**  **Входы** | **a0** | **a1** |
| **z0** | **a0** | **a0** |
| **z1** | **a1** | **a1** |

Таблица выходов:

|  |  |  |
| --- | --- | --- |
| **Состояния**  **Входы** | **a0** | **a1** |
| **z0** | y0 | y0 |
| **z1** | y1 | y1 |

Совмещенная таблица переходов и выходов автомата Мили.

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

Представим автомат Мили в графическом виде:

y0

a1

y0

z0

z1

z0

a0

y1

y1

z1

Модель автомата Мили:

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

Код:

import matplotlib.pyplot as plt

# Состояния

states = ['a0', 'a1'] # a0: смена на красный, a1: смена на зеленый

# Таблица переходов

# Строки: состояние, столбцы: входные сигналы z0, z1

T = [

[0, 1], # Состояние a0: переход на a0 при z0, на a1 при z1

[1, 0] # Состояние a1: переход на a1 при z0, на a0 при z1

]

# Таблица выходных сигналов

# Строки: состояние, столбцы: входные сигналы z0, z1

outputs = [

['y0', 'y1'], # Состояние a0: y0 при z0, y1 при z1

['y1', 'y0'] # Состояние a1: y1 при z0, y0 при z1

]

# Входная последовательность

input\_sequence = ['z0', 'z1', 'z0', 'z1', 'z0']

def plot\_state\_changes(input\_sequence):

current\_state = 0 # Начальное состояние a0

time\_steps = []

state\_sequence = []

output\_sequence = []

time\_step = 0

for signal in input\_sequence:

# Определяем индекс входного сигнала (z0 -> 0, z1 -> 1)

z = 0 if signal == 'z0' else 1

# Получаем следующее состояние

current\_state = T[current\_state][z]

# Получаем выходной сигнал для текущего состояния и входного сигнала

y = outputs[current\_state][z]

# Добавляем данные для графика

time\_steps.append(time\_step)

state\_sequence.append(current\_state)

output\_sequence.append(y)

time\_step += 1

# Визуализация

plt.figure(figsize=(10, 5))

# График состояний

plt.subplot(2, 1, 1)

plt.plot(time\_steps, state\_sequence, marker='o', linestyle='-', color='b')

plt.yticks([0, 1], states) # Подписи для состояний

plt.xticks(time\_steps) # Подписи для времени

plt.xlabel("Время")

plt.ylabel("Состояния")

plt.title("Изменения состояний автомата Мили по времени")

plt.grid(True)

# График выходных сигналов

plt.subplot(2, 1, 2)

plt.plot(time\_steps, output\_sequence, marker='o', linestyle='-', color='r')

plt.yticks(['y0', 'y1'], ['y0', 'y1']) # Подписи для выходных сигналов

plt.xticks(time\_steps) # Подписи для времени

plt.xlabel("Время")

plt.ylabel("Выход")

plt.title("Выходные сигналы автомата Мили по времени")

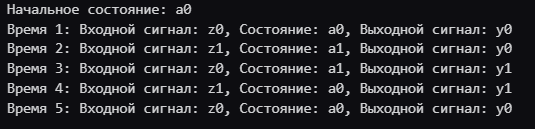
plt.grid(True)

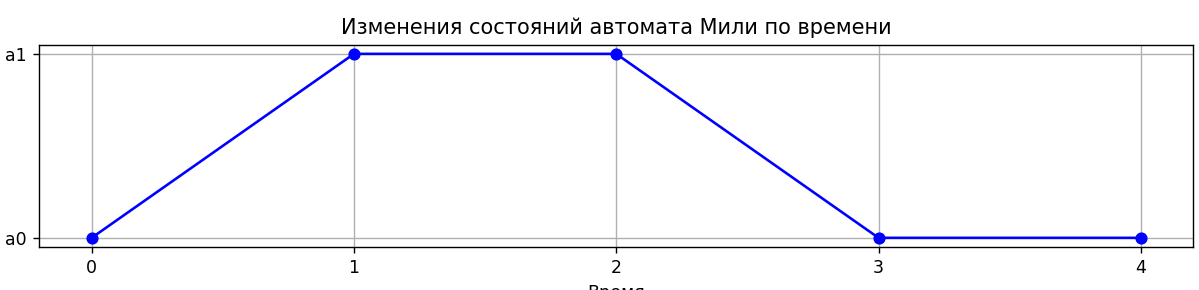
plt.tight\_layout()

plt.show()

plot\_state\_changes(input\_sequence)

Результаты:





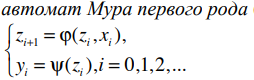
2. Автомат Мура:

Рассмотрим автомат Мили, предназначенный для управления пешеходным светофором.

Элементы автомата Мили

* Входные сигналы:
  + z0 - Кнопка закрыть
  + z1 - Кнопка открыть
  + z2 - Датчик дня
  + z3 - Датчик ночи
* Выходные сигналы:
  + w0 - Открыть шторы
  + w1 - Закрыть шторы
* Состояния:
  + a0 - Ночь, закрыты
  + a1 - Ночь, открыты
  + a2 - День, закрыты
  + a3 - День, Открыты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № 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:

z1

z1

z0

z2

z0

z3

z3

z3

z3

z2

z2

z2

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")

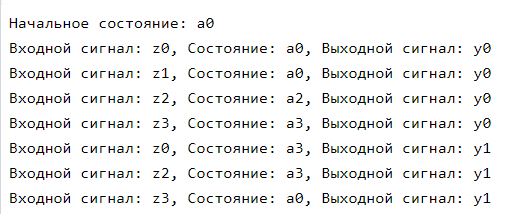
    plt.title("Изменения состояний по времени")

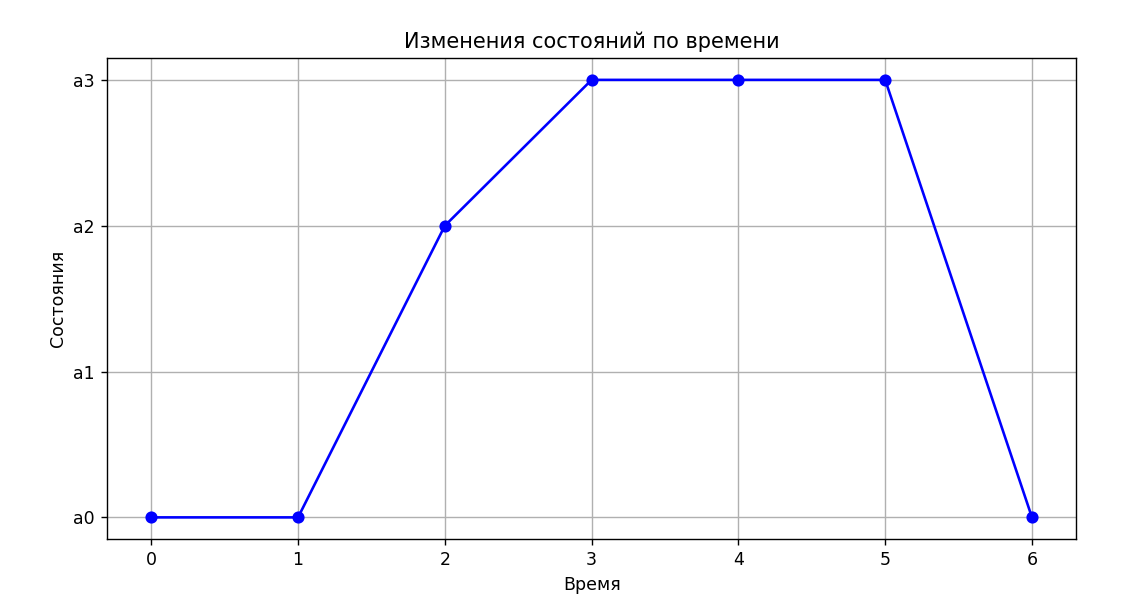
    plt.grid(**True**)

    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.