

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”

Факультет	Программной Инженерии и Компьютерной Техники
Направление подготовки (специальность)	Нейротехнологии и программирование
Дисциплина	Теория систем

ЛАБОРАТОРНАЯ РАБОТА 1
ОТЧЕТ

Выполнил студент: **Фам Данг Чунг Нгиа**

Группа: **P3321**

Преподаватель: **Русак Алена Викторовна**

г. Санкт-Петербург

2026

Содержание

1. Описание задания.....	3
2. Описание реализации конечного автомата.....	3
2.1. Объект управления.....	3
2.2. Состояния автомата.....	4
2.3. Входные события (input events / triggers).....	4
2.4. Выходные сигналы (output signals).....	4
2.5. Таблицы	6
3. Диаграмма переходов	9
4. Выполнение	9
5. Вывод.....	9

1. Описание задания

Требуется спроектировать управляющий конечный автомат:

- определить возможные состояния (не менее 5) и переходы между ними;
- определить входные события, инициирующие переходы, и выходные сигналы;
- построить диаграмму переходов автомата;
- реализовать работу автомата на языке Python (с использованием библиотеки `transitions` или без неё).

2. Описание реализации конечного автомата

2.1. Объект управления

В качестве объекта управления выбран **лифтовый механизм** (elevator controller).

Система должна:

- принимать вызов с определённого этажа,
- определять направление движения (вверх/вниз),
- управлять двигателем лифта,
- открывать и закрывать двери,
- обрабатывать препятствия при закрытии дверей,
- реагировать на аварийную ситуацию (emergency),
- переходить в состояние **FAULT** и возвращаться в рабочий режим после **reset**.

Реализация выполнена на Python с использованием библиотеки **transitions**.

Автомат реализован как **автомат Мили (Mealy machine)**, поскольку выходные сигналы формируются при переходах между состояниями (after callbacks), а не только при входе в состояние.

2.2. Состояния автомата

- **IDLE** — лифт находится в состоянии ожидания вызова.
- **MOVE_UP** — лифт движется вверх.
- **MOVE_DOWN** — лифт движется вниз.
- **DOOR_OPENING** — двери открываются.
- **DOOR_OPEN** — двери полностью открыты.
- **DOOR_CLOSING** — двери закрываются.
- **FAULT** — аварийный режим.

2.3. Входные события (input events / triggers)

Входной алфавит автомата включает следующие события:

- **call(f)** — вызов лифта на этаж f (разделяется логически на случаи: $f > cf$, $f < cf$, $f = cf$)
- **arrived** — лифт прибыл на целевой этаж.
- **door_opened** — двери полностью открылись.
- **timeout** — истекло время ожидания при открытых дверях.
- **door_closed** — двери полностью закрылись.
- **obstacle** — обнаружено препятствие при закрытии дверей.
- **emergency** — аварийная ситуация.
- **reset** — сброс аварийного режима.

2.4. Выходные сигналы (output signals)

Выходной алфавит автомата включает следующие управляющие сигналы:

- **lift_motor_up_on** — включить двигатель движения вверх
- **lift_motor_down_on** — включить двигатель движения вниз
- **lift_motor_stop** — остановить двигатель лифта
- **door_motor_open** — открыть двери
- **door_motor_close** — закрыть двери
- **alarm_on** — включить сигнал аварии
- **alarm_off** — выключить сигнал аварии
- **display_idle** — отображать состояние IDLE
- **display_moving_up** — отображать движение вверх
- **display_moving_down** — отображать движение вниз
- **display_arrived** — отображать прибытие
- **display_door_open** — отображать «двери открыты»
- **display_closing** — отображать «закрытие дверей»
- **display_fault** — отображать аварийное состояние

Выходные сигналы формируются при переходах между состояниями (after transition callbacks), что соответствует модели автомата Мили:

$$\lambda : X \times U \rightarrow Y$$

2.5. Таблицы

<p>Таблица переходов</p> <p>$\delta : X \times U \rightarrow X$</p>										
Текущее состояние $x \in X$	Символы входного алфавита									
	$u \in U$									
	call(f), f > cf	call(f), f < cf	call(f), f = cf	arrived	door_opened	timeout	door_closed	obstacle	emergency	reset
IDLE	MOVE_UP	MOVE_DOWN	DOOR_OPENING	IDLE	IDLE	IDLE	IDLE	IDLE	FAULT	IDLE
MOVE_UP	MOVE_UP	MOVE_UP	MOVE_UP	DOOR_OPENING	MOVE_UP	MOVE_UP	MOVE_UP	MOVE_UP	FAULT	MOVE_UP
MOVE_DOWN	MOVE_DOWN	MOVE_DOWN	MOVE_DOWN	DOOR_OPENING	MOVE_DOWN	MOVE_DOWN	MOVE_DOWN	MOVE_DOWN	FAULT	MOVE_DOWN
DOOR_OPENING	DOOR_OPENING	DOOR_OPENING	DOOR_OPENING	DOOR_OPENING	DOOR_OPEN	DOOR_OPENING	DOOR_OPENING	DOOR_OPENING	FAULT	DOOR_OPENING
DOOR_OPEN	DOOR_OPEN	DOOR_OPEN	DOOR_OPEN	DOOR_OPEN	DOOR_OPEN	DOOR_CLOSING	DOOR_OPEN	DOOR_OPEN	FAULT	DOOR_OPEN
DOOR_CLOSING	DOOR_CLOSING	DOOR_CLOSING	DOOR_CLOSING	DOOR_CLOSING	DOOR_CLOSING	DOOR_CLOSING	IDLE	DOOR_OPENING	FAULT	DOOR_CLOSING
FAULT	FAULT	FAULT	FAULT	FAULT	FAULT	FAULT	FAULT	FAULT	FAULT	IDLE

<p>Таблица выходов</p> <p>$\lambda : X \times U \rightarrow Y$</p>										
Текущее состояние $x \in X$	Символы входного алфавита									
	$u \in U$									
	call(f), f > cf	call(f), f < cf	call(f), f = cf	arrived	door_opened	timeout	door_closed	obstacle	emergency	reset
IDLE	lift_motor_up_on, display_moving_up	lift_motor_down_on, display_moving_down	door_motor_open, display_arrived	none	none	none	none	none	lift_motor_stop, alarm_on, display_fault	none
MOVE_UP	none	none	none	lift_motor_stop, door_motor_open, display_arrived	none	none	none	none	lift_motor_stop, alarm_on, display_fault	none
MOVE_DOWN	none	none	none	lift_motor_stop, door_motor_open, display_arrived	none	none	none	none	lift_motor_stop, alarm_on, display_fault	none

DOOR_ OPENING	none	none	none	none	display_ door_op en	none	none	none	alarm_ on, display_ _fault	none
DOOR_ OPEN	none	none	none	none	none	door_m otor_clo se, display_ closing	none	none	alarm_ on, display_ _fault	none
DOOR_ CLOSING	none	none	none	none	none	none	displa y_idle	door_m otor_op en, display_ door_op en	alarm_ on, display_ _fault	none
FAULT	none	none	none	none	none	none	none	none	alarm_ on, display_ _fault	alar m_of f, displ ay_id le

Можете удобно смотреть в этой ссылке:

https://www.notion.so/Tables-30d44f44451280139ff5c6f7c0867483?source=copy_link

3. Диаграмма переходов

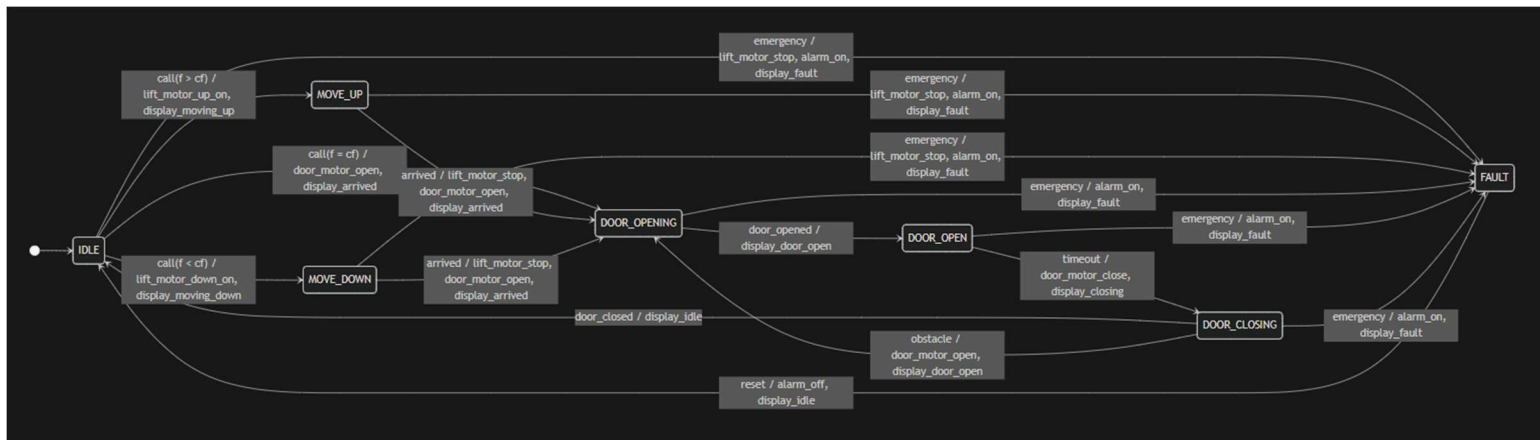


Рисунок 1 – Граф автомата Мили

4. Выполнение

https://colab.research.google.com/drive/145xEHzNt6RPTZ9HL_g4TnDHRTdl94zkJ?usp=sharing

5. Вывод

В ходе работы был спроектирован и реализован управляющий конечный автомат для лифтовой системы. Были определены состояния, входные события, выходные сигналы, а также функции переходов δ и выходов λ . Автомат реализован как автомат Мили, поскольку выходы формируются при переходах между состояниями и зависят от входного события. Построена диаграмма переходов и выполнена программная реализация на Python с использованием библиотеки transitions. Проведённое тестирование показало корректность работы автомата во всех предусмотренных сценариях (движение, открытие/закрытие дверей, обработка препятствия и аварийного режима).