Министерство науки и высшего образования РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

**«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт космических и информационных технологий

институт

Программная инженерия

кафедра

**ОТЧЕТ О ПРАКТИЧЕСКОЙ** **РАБОТЕ №1**

Конечные автоматы

тема

Преподаватель А. С. Кузнецов

подпись, дата инициалы, фамилия

Студент КИ23-17/1Б, 032320072 М. А. Мальцев

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2025

# Цель

Реализация и исследование детерминированных и недетерминированных конечных автоматов.

# Задание

Вариант – 6.

Для выполнения практической работы необходимо разработать в системе JFLAP конечные автоматы и произвести программную реализацию на языке   
С++ для следующих автоматов:

1) Построить ДКА, допускающий в алфавите {0, 1} множество всех цепочек, у которых на пятой позиции справа стоит 1.

2) Построить НКА с количеством состояний, не превышающим 3, для языка {an : n ≥ 1} U {bmak : m, k ≥ 0}.

# Ход выполнения

Для начала была установлена программа JFLAP, в которой были построены конечные автоматы из условия задания. Каждый КА был сначала протестирован в JFLAP тестовыми цепочками, затем была написана программная реализация на C++, которая также была протестирована на корректность работы теми же самыми тестовыми цепочками.

# 3.1 Реализация ДКА

Необходимо было реализовать ДКА, допускающий в алфавите {0, 1} множество всех цепочек, у которых на пятой позиции справа стоит 1. Исходя из формулировки задачи было выдвинуто предположение, что на вход автомату может поступать строка неограниченной длины и необходимо определить, является ли пятый символ с конца единицей. Для решения данной задачи была придумана модель скользящего окна, которая имеет длину в 5 символов и проходит от самого начала строки до её конца (в самом начале окно слева от строки и заполнено нулями, в самом конце оно содержит последние пять символов исследуемой строки). В таком окне максимально возможное количество различных комбинаций 1 и 0 это 32, и все они могут получиться при сдвиге окна, поэтому в ДКА будет 32 состояния и из каждого будут выходить два перехода (для 0 и 1). На рисунке 1 показана реализация ДКА в системе JFLAP.

Для выполнения первого задания необходимо создать детерминированный конечный автомат. Так как значение будет зависеть от остатка деления количества «a» на 3, то имеем три состояния: q0 – остаток деления «a» на 3 ровно 0, q1 – остаток деления «a» на 3 ровно 1 и q2 – остаток деления «a» на 3 ровно 0. Финальным состоянием считается q2, так как по заданию требуется, чтобы остаток деления был больше 1, то есть равен 2. Функция переходов изображена в таблице 1.

Таблица 1 – Функция переходов для ДКА

|  |  |  |
| --- | --- | --- |
| **Состояние** | **A** | **B** |
| q0 | q1 | q0 |
| q1 | q2 | q1 |
| q2 | q0 | q2 |

На рисунке 1 показана реализация ДКА в системе JFLAP.

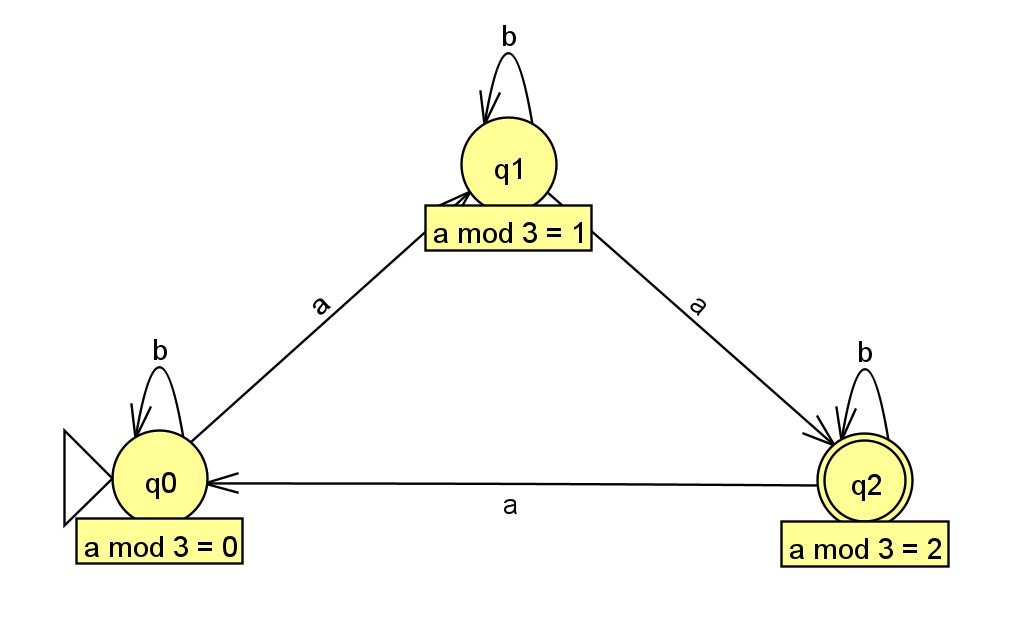


Рисунок 1 – ДКА первого задания

В качестве тестов будем использовать следующие цепочки: «aaabbaa», «abbaa», «abb». Проведя тесты получаем следующие результаты (рисунки 2-4).

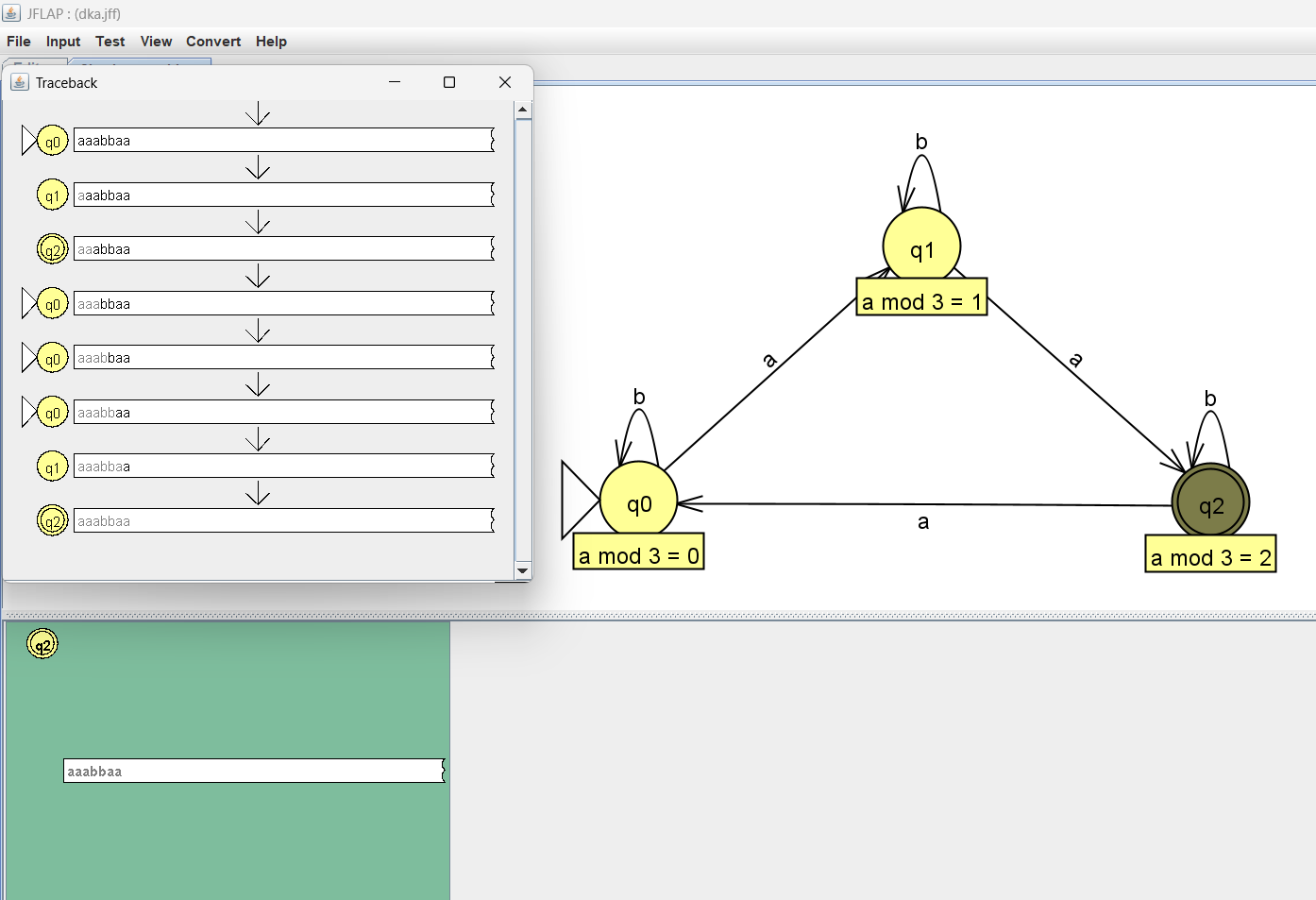


Рисунок 2 – Тест для цепочки «aaabbaa»

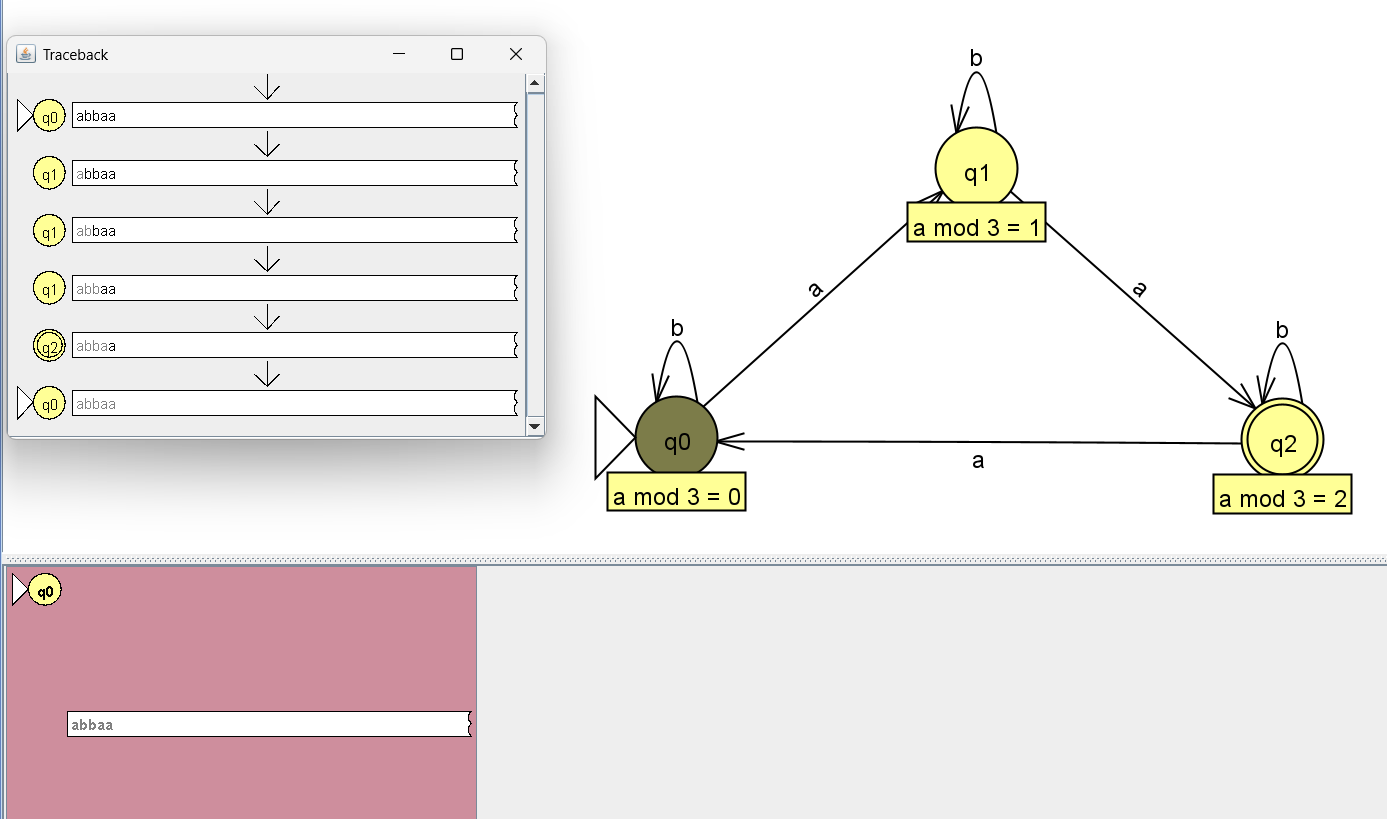


Рисунок 3 – Тест для цепочки «abbaa»

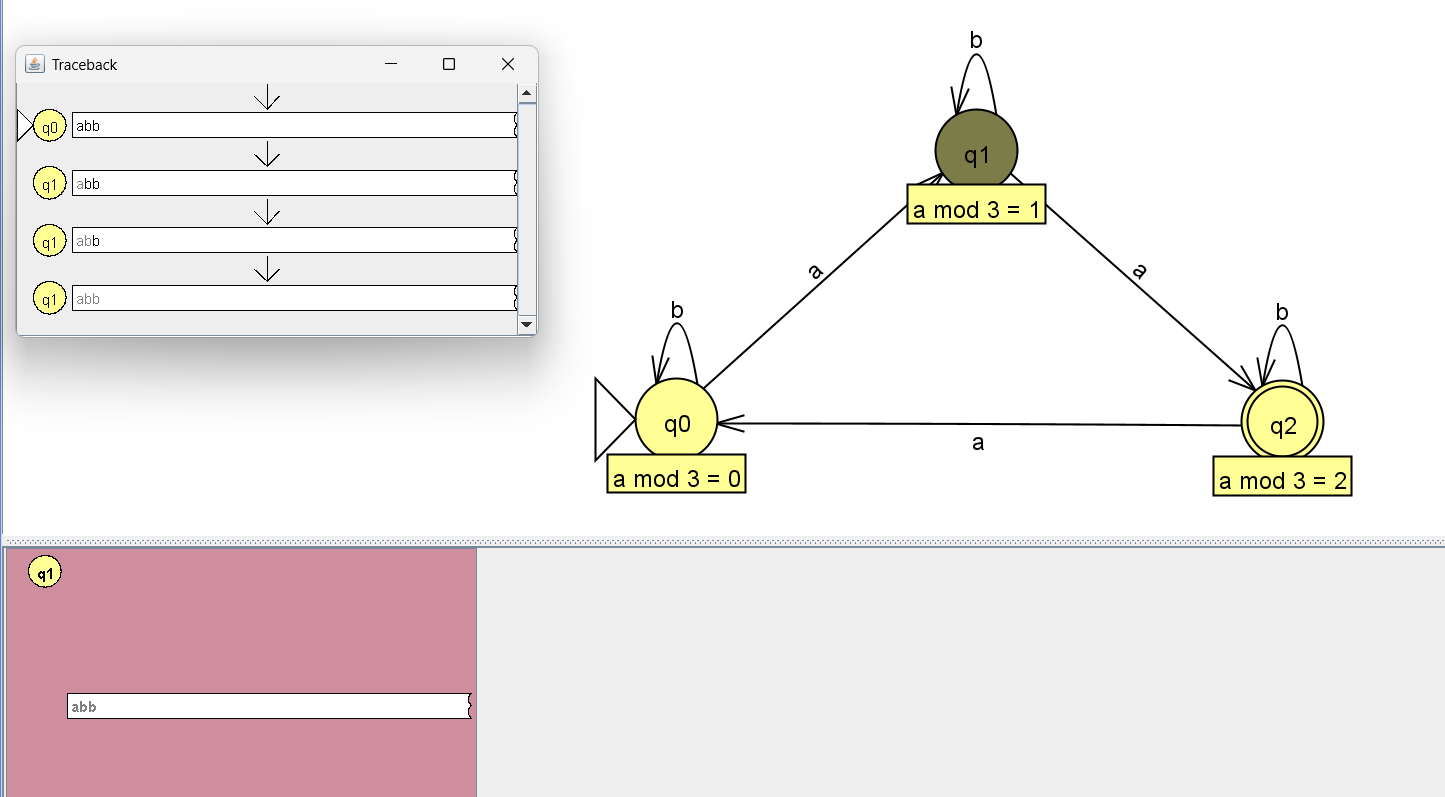


Рисунок 4 – Тест для цепочки «abb»

Программный код для данного автомата был написан на языке Python, продемонстрирован на рисунке 5.

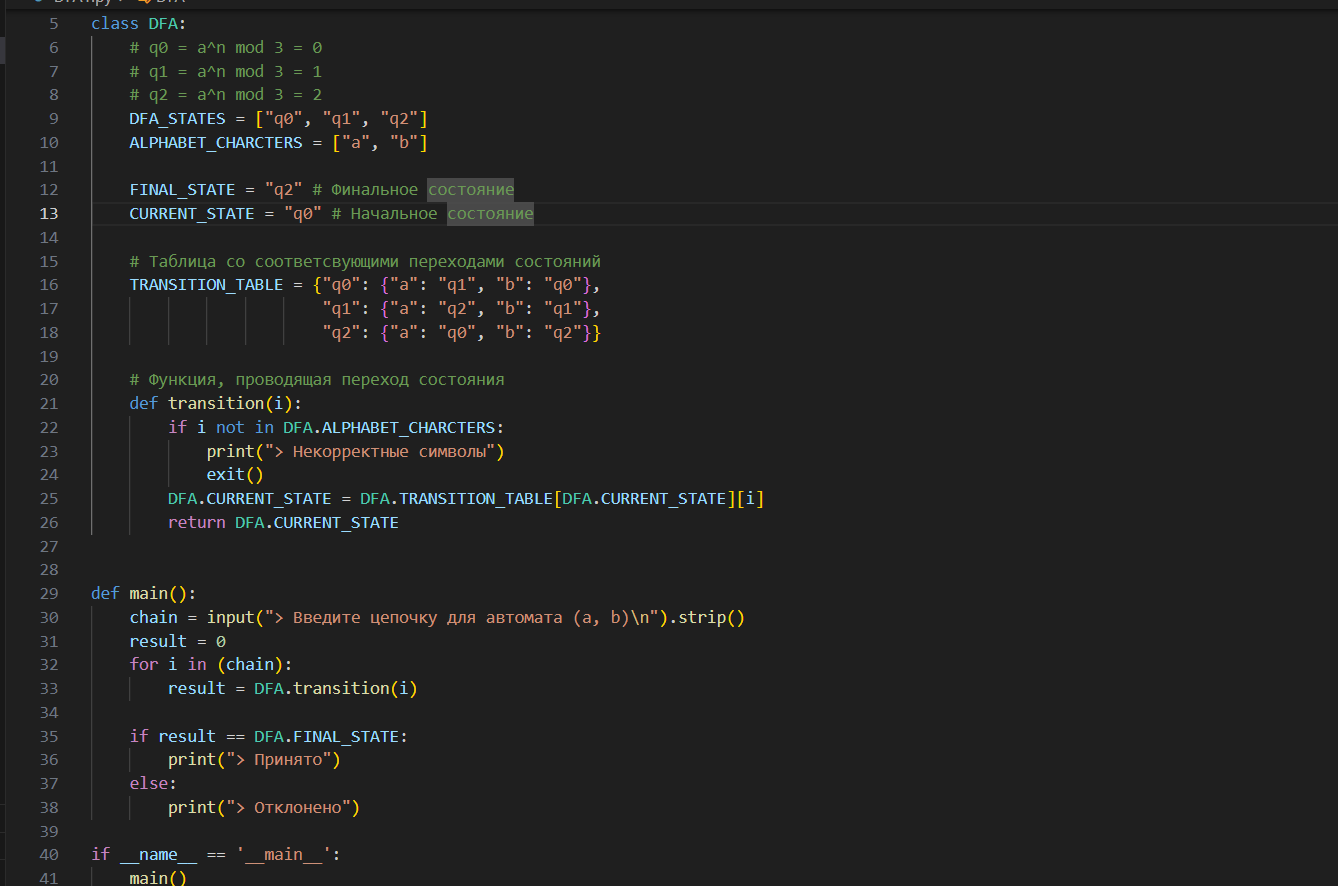


Рисунок 5 – Код для НКА

На рисунке 6 показаны те же тесты, но в консоли программы.

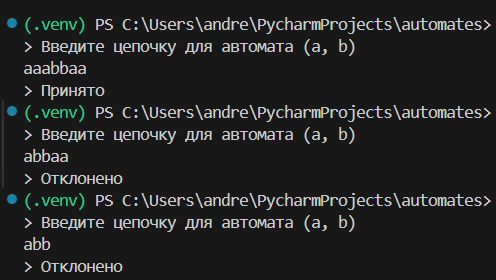


Рисунок 6 – Тесты для ДКА в Python

# 3.2 Построение НКА

Для выполнения второго задания необходимо создать недетерминированный конечный автомат. В данном задании по условию мы можем использовать не более 5 состояний, сами цепочки должны соответствовать следующим выражениям: ababn или aban, где n >= 0. У этих цепочек есть общее начало «ab», что значит, что после этих шагов мы имеем разветвление: либо «aba» + an либо «aba» + bn, построим функцию переходов в виде таблицы (таблица 2).

Таблица 2 – Функция переходов для НКА

|  |  |  |
| --- | --- | --- |
| **Состояние** | **A** | **B** |
| q0 | q1 | – |
| q1 | – | q2 |
| q2 | q3, q4 | – |
| q3 | q3 | – |
| q4 | – | q4 |

Конечными состояниями в нашем случае будут q2, q3, q4. На рисунке 7 показана реализация НКА в системе JFLAP.

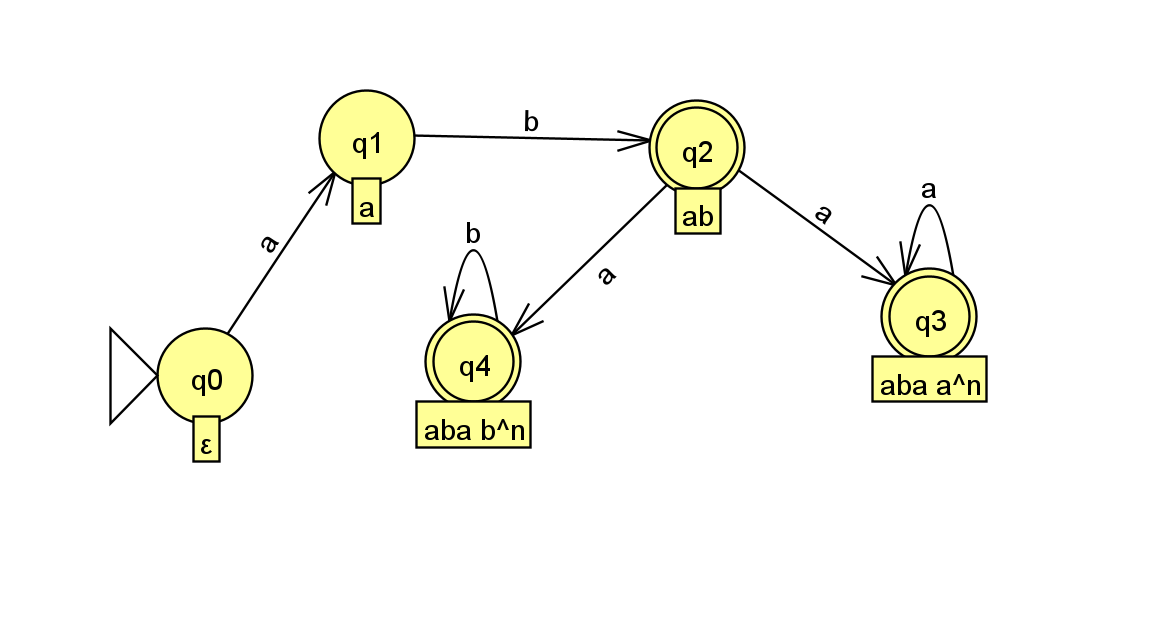


Рисунок 7 – НКА второго задания

В качестве тестов будем использовать следующие цепочки: «bab», «ab», «abaaab», «abaa», «ababbb», «ababba». Проведя тесты получаем следующие результаты (рисунки 8-13).

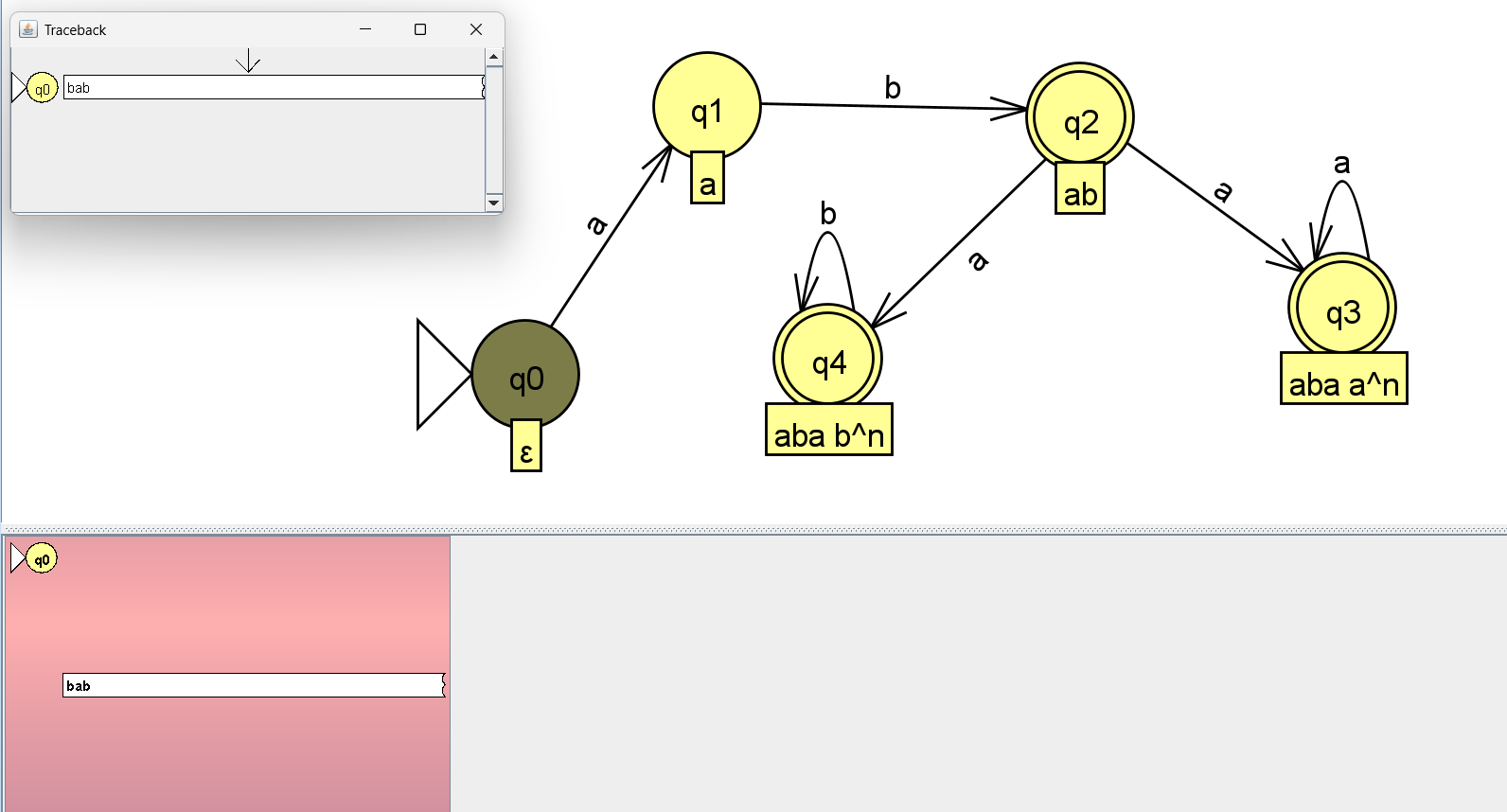


Рисунок 8 – Тест для цепочки «bab»

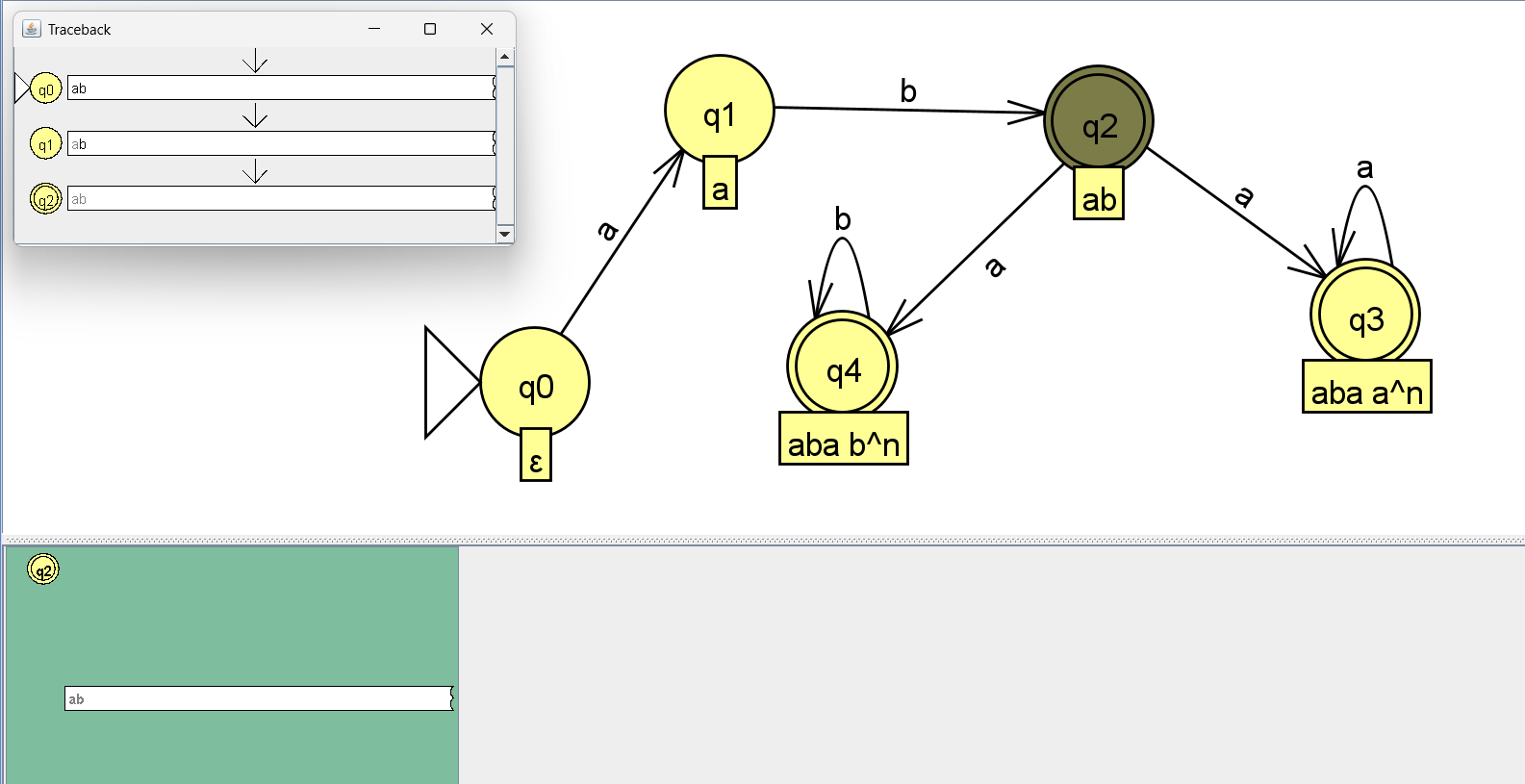


Рисунок 9 – Тест для цепочки «ab»

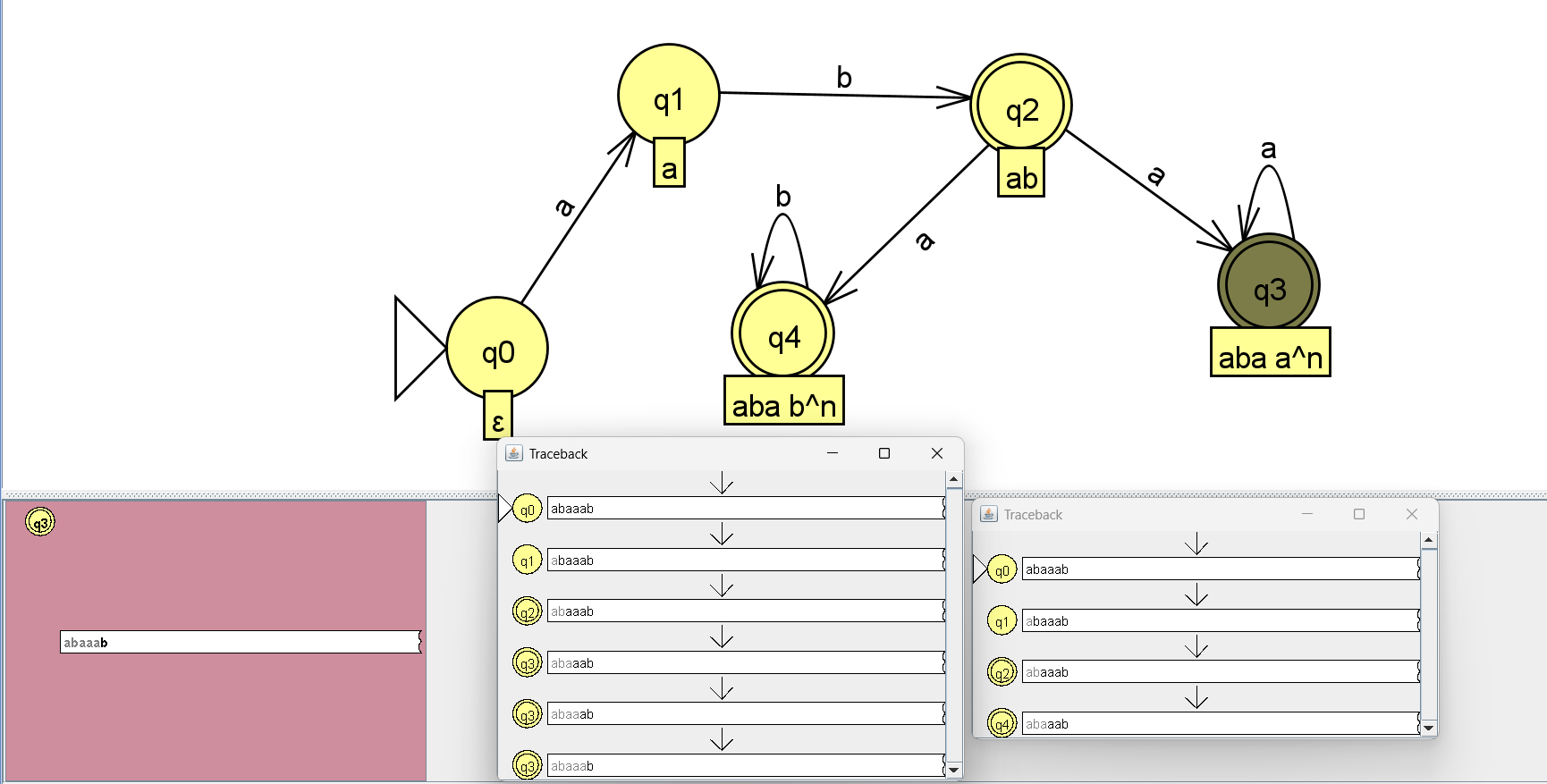


Рисунок 10 – Тест для цепочки «abaaab»

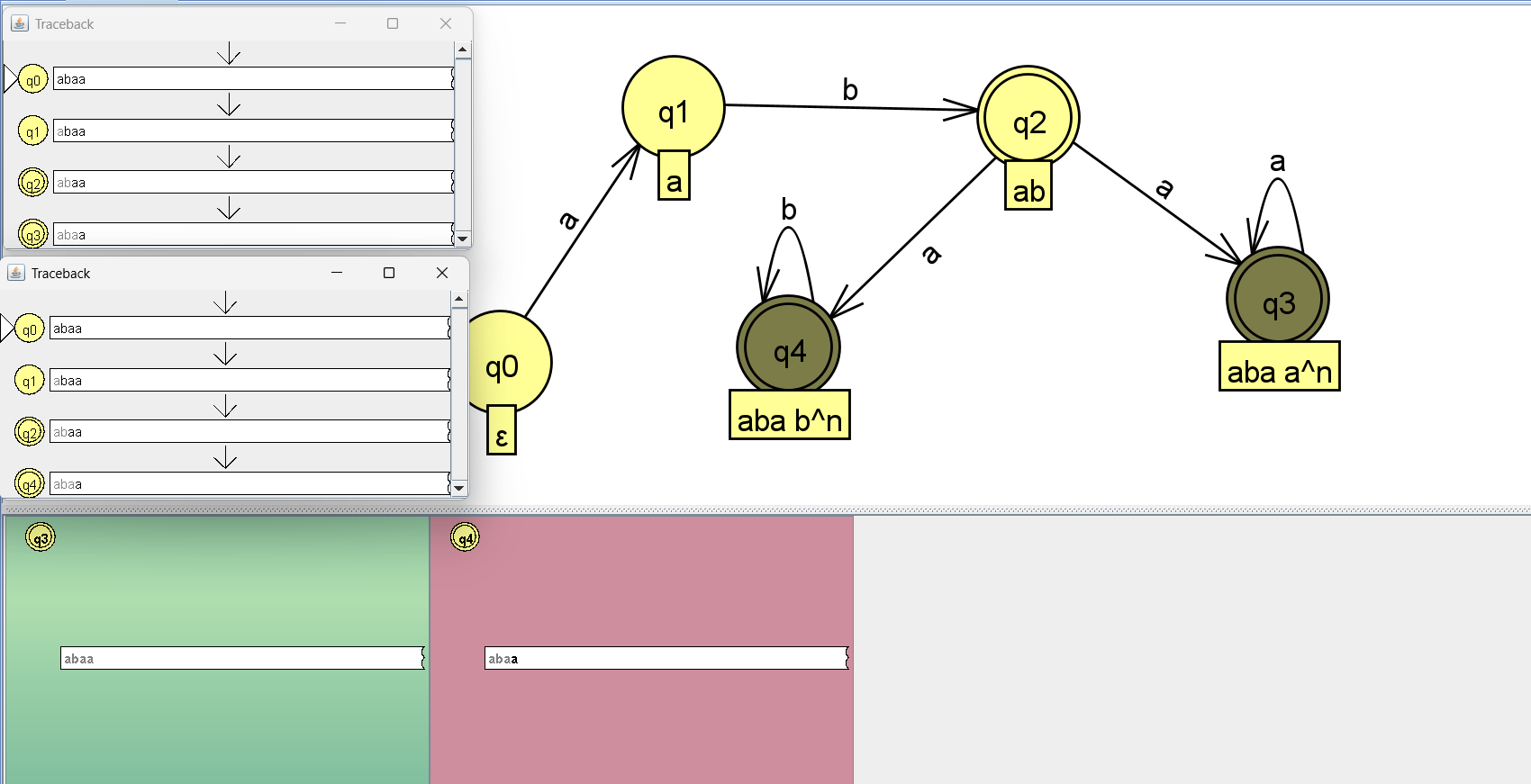


Рисунок 11 – Тест для цепочки «abaa»

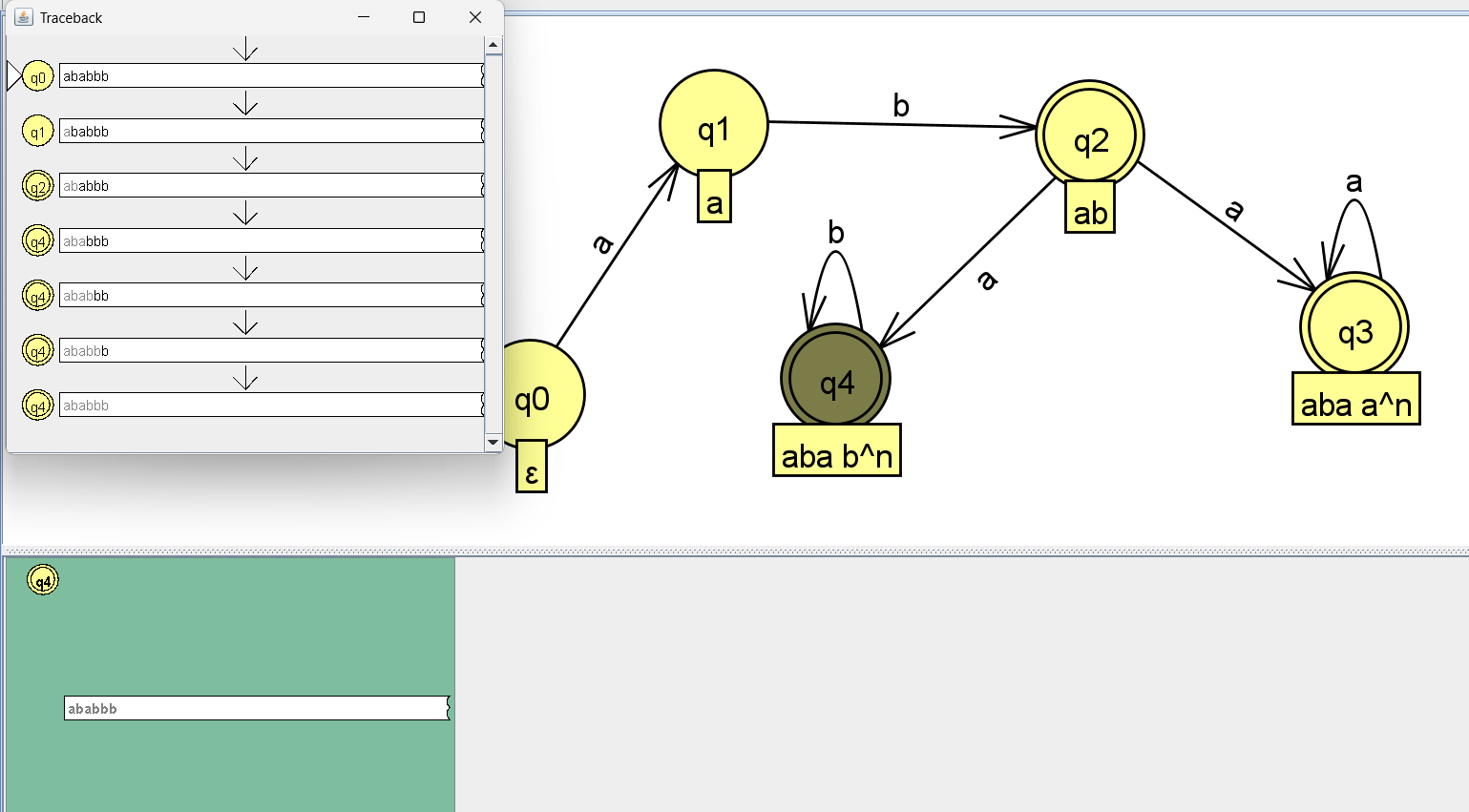


Рисунок 12 – Тест для цепочки «ababbb»

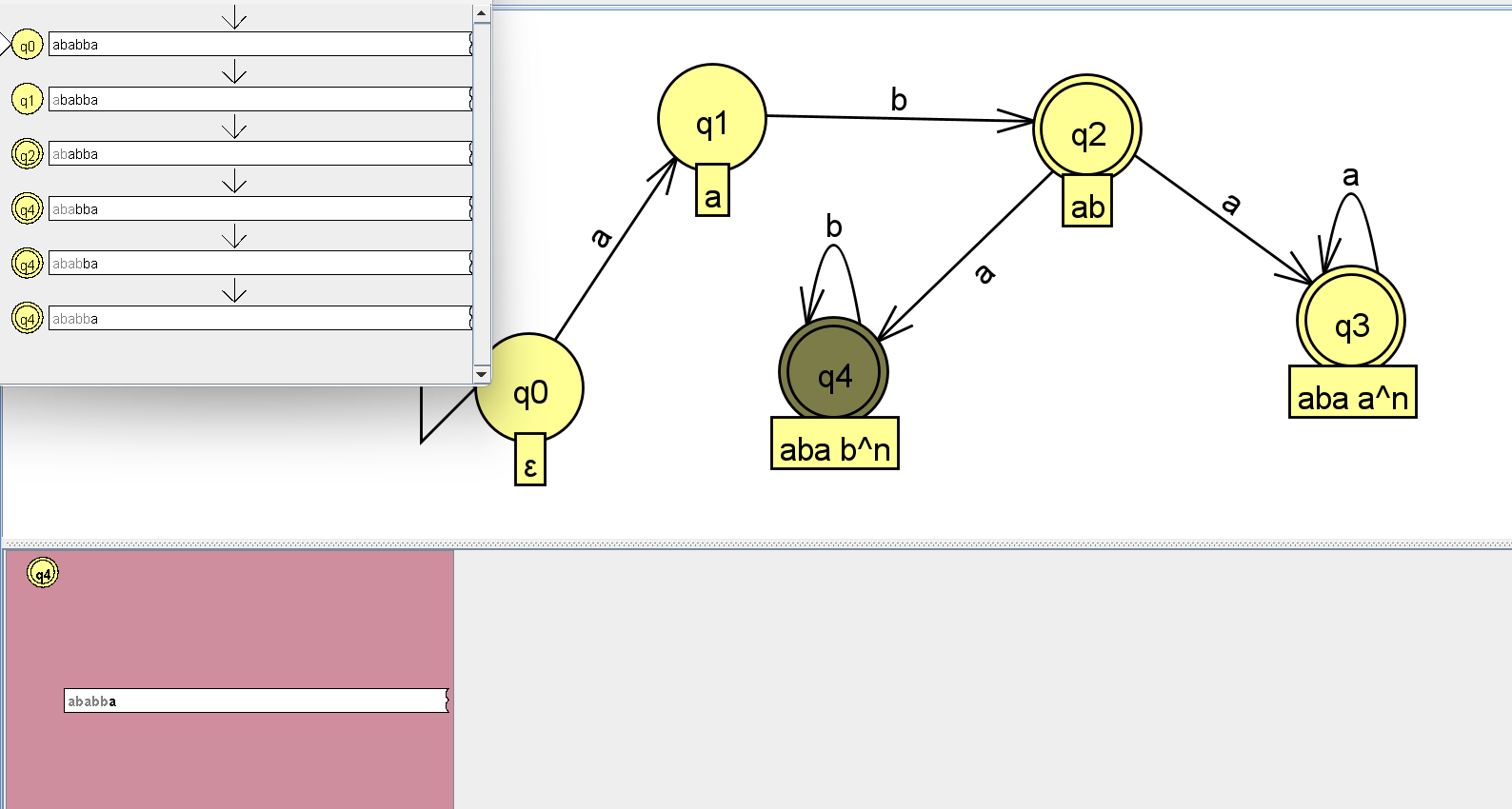


Рисунок 13 – Тест для цепочки «ababba»

Программный код для данного автомата был написан на языке Python, продемонстрирован на рисунке 14.

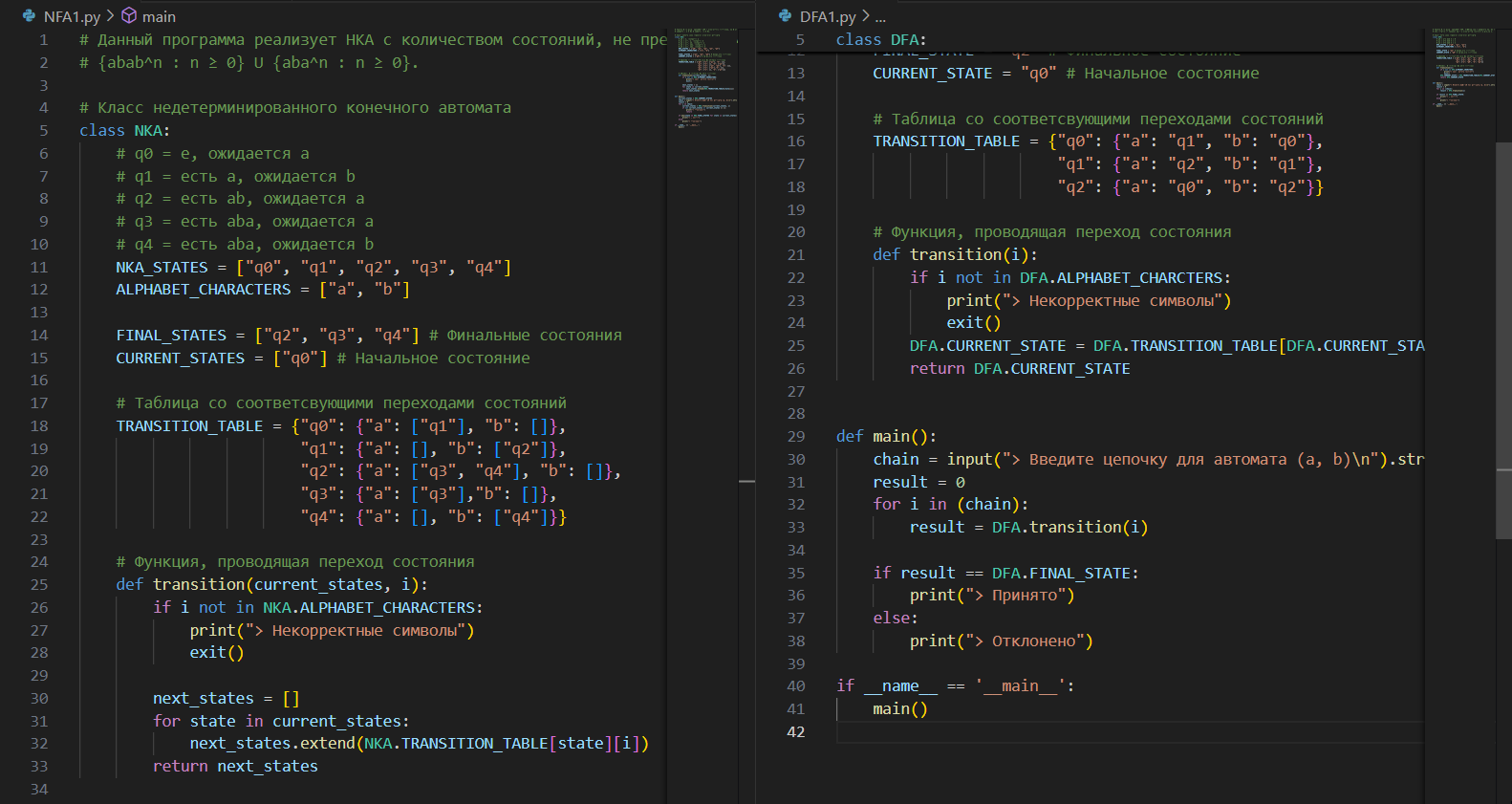


Рисунок 14 – Код для НКА

На рисунке 15 показаны те же тесты, но в консоли программы.

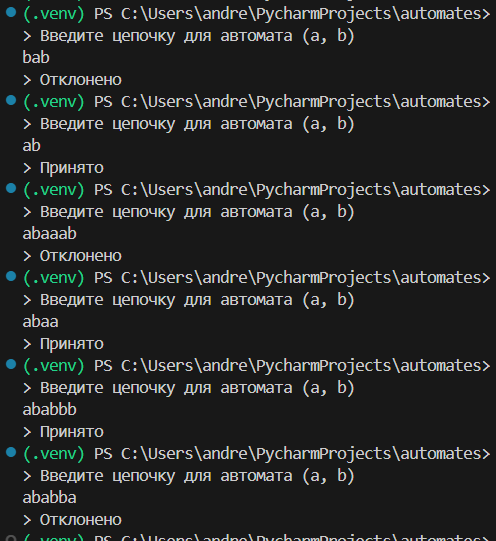


Рисунок 15 – Тесты для ДКА в Python

# Выводы

В ходе данной практической работы был изучен материал о детерминированных и недетерминированных конечных автоматах, были выполнены все задания, построены ДКА и НКА в системе JFLAP и реализован программный код.