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.

## Построение ДКА

Необходимо было реализовать детерминированный конечный автомат (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 первого задания в системе JFLAP.

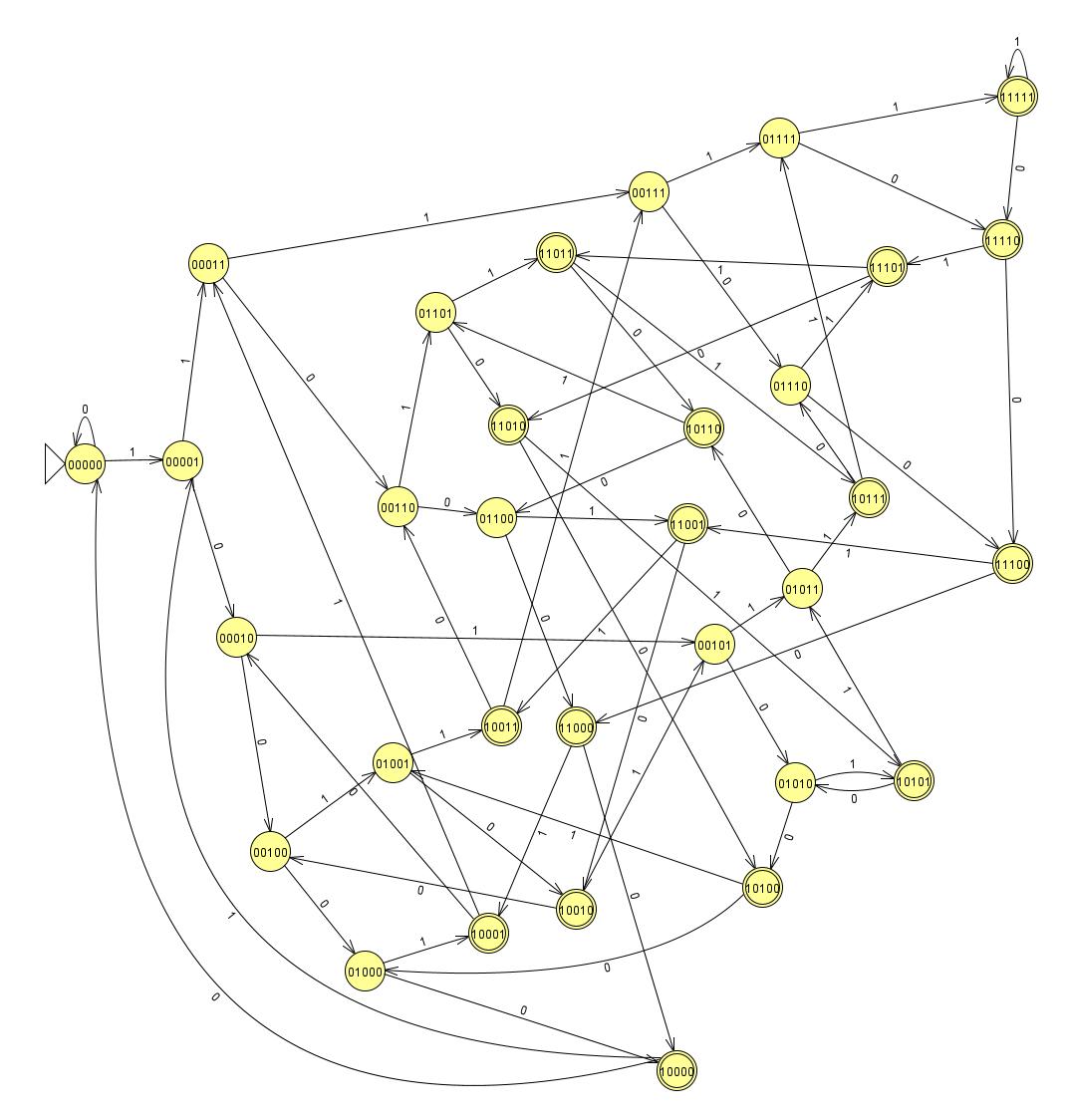


Рисунок 1 – ДКА в JFLAP

Конечный автомат получился довольно объемным благодаря условию задачи. Переписывание 64 функции перехода в таблицу могло занять много времени, поэтому было принято решение перейти к тестам. На рисунке 2 показан тест ДКА для цепочки «0000110101», пятая цифра справа которого является единицей.

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2 – Тест для цепочки «00001010101»

Также была проверка на отклонение строки, которая не удовлетворяет условию. На рисунке 3 показан тест ДКА для цепочки «0100001».

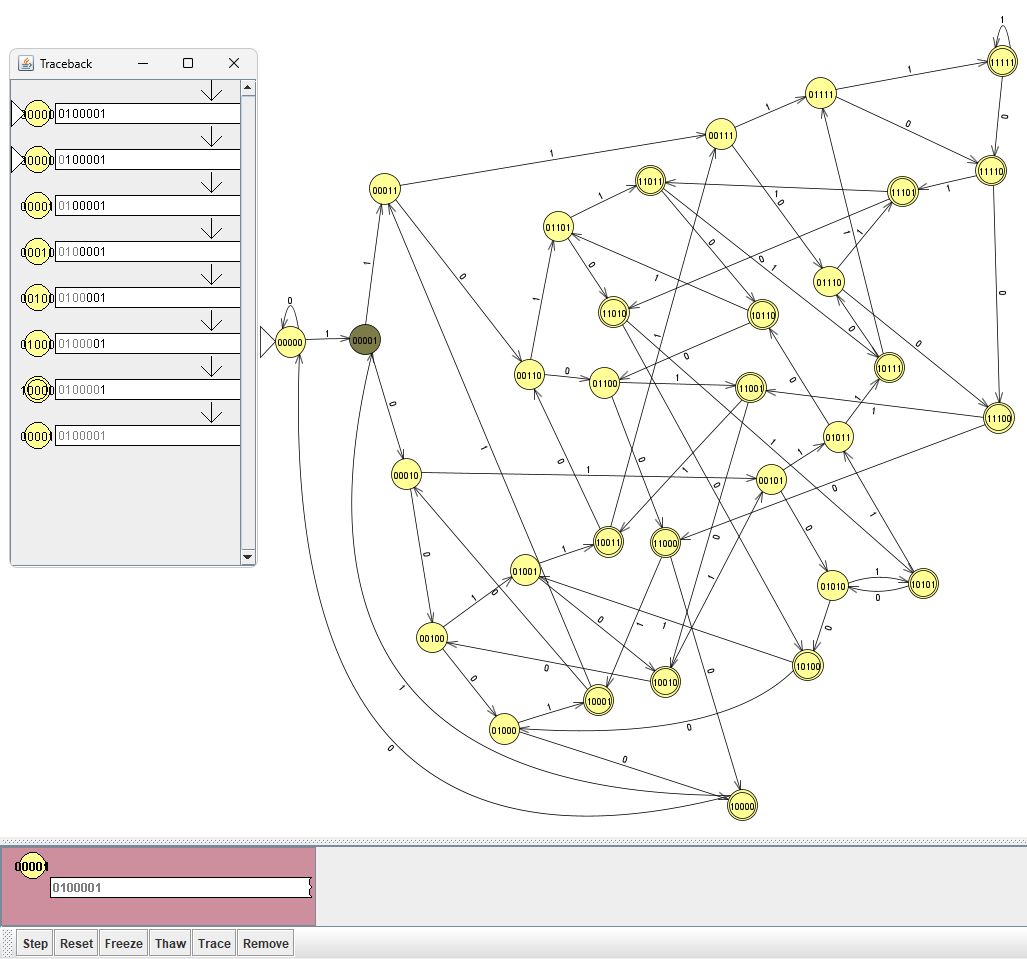


Рисунок 3 – Тест для цепочки «0100001»

Не смотря на свой размер, ДКА хорошо прошла все тесты. Метод минимизации ДКА (алгоритм заполнения таблицы) не дал результатов, поэтому данный вид является её основным.

Дальше был написан код на C++ для реализации данного ДКА, он показан на рисунке 4.

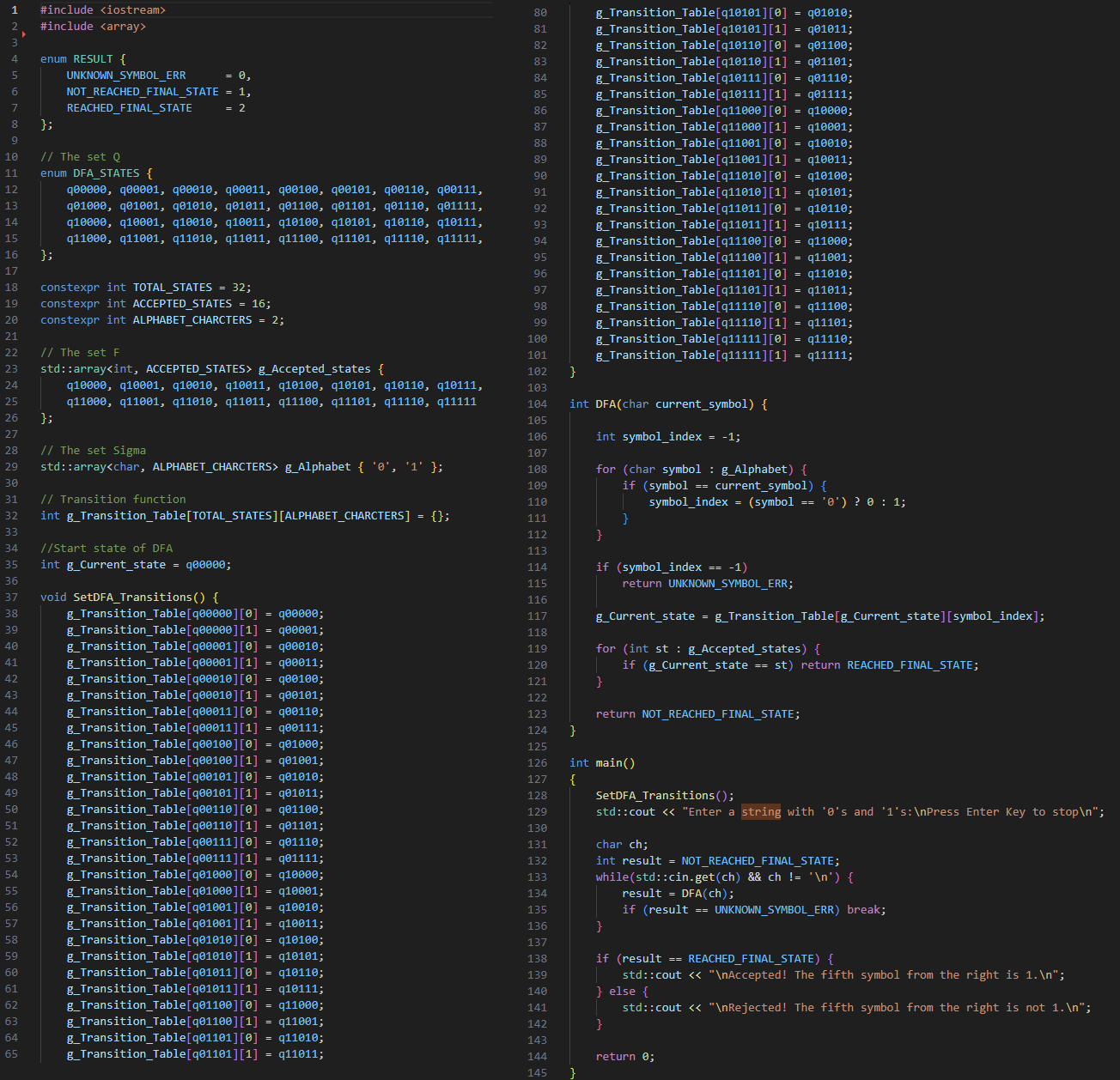


Рисунок 4 – Код для ДКА на C++

После компиляции, сборки и запуска программы были повторно проведены тесты, показанные на рисунке 5.

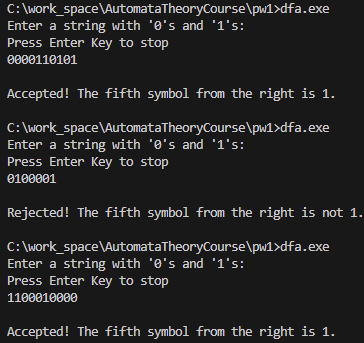


Рисунок 5 – Тесты для ДКА на C++

## Построение НКА

Также необходимо было реализовать недетерминированный конечный автомат (НКА), допускающий все строки из алфавита {an : n ≥ 1} U {bmak : m, k ≥ 0}. Необходимо было создать не более трёх состояний. Для реализации НКА был принят во внимание тот факт, что ДКА является недетерминированным, например, в том случае, если из какого-либо состояния автомата есть несколько переходов с одним и тем же входным символом, а также то, что цепочка считается допустимой в том случае, если есть хотя бы одно допускающее состояние из всех «параллельно активных». Также анализ алфавита показал, что принимается либо любое количество символа «а», либо любое количество символа «b», либо любое количество «b» и после этого любое количество «а», либо пустая цепочка (при m=k=0 bmak равно пустой строке ε). На рисунке 6 показана реализация ДКА первого задания в системе JFLAP.

Изображение выглядит как диаграмма, круг, линия, рисунок

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 6 - НКА в JFLAP.

Таблица функций переходов данного НКА представлена на таблице 1.

Таблица 2 – Функция переходов для НКА

|  |  |  |
| --- | --- | --- |
| **Состояние** | **a** | **b** |
| q0 | q1, q1 | q0 |
| q1 | q1 | q2 |
| q2 | q2 | q2 |

Дальше необходимо было провести все самые главные тесты для этой цепочки, они показаны на рисунке 7, 8, 9, 10, 11.

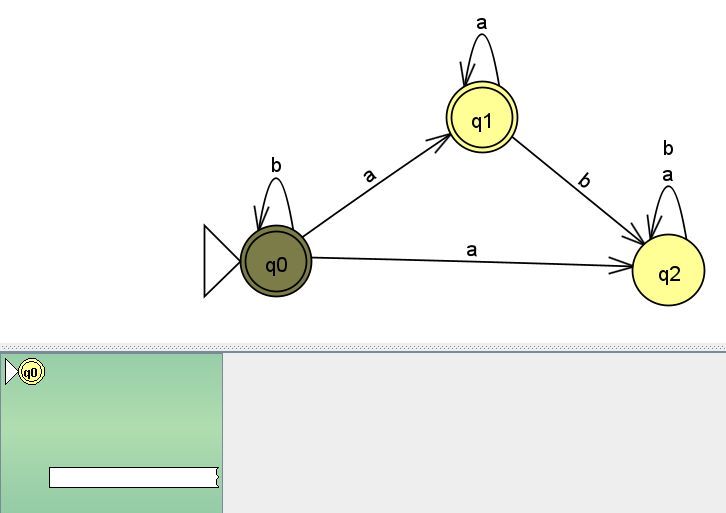


Рисунок 7 – Тест для пустой цепочки (ε).

Изображение выглядит как снимок экрана, диаграмма, дизайн

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 8 – Тест для цепочки «aaa»

Изображение выглядит как диаграмма, снимок экрана, линия, круг

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 9 – Тест для цепочки «bbb»

Изображение выглядит как диаграмма, снимок экрана, текст, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 10 – Тест для цепочки «aabb»

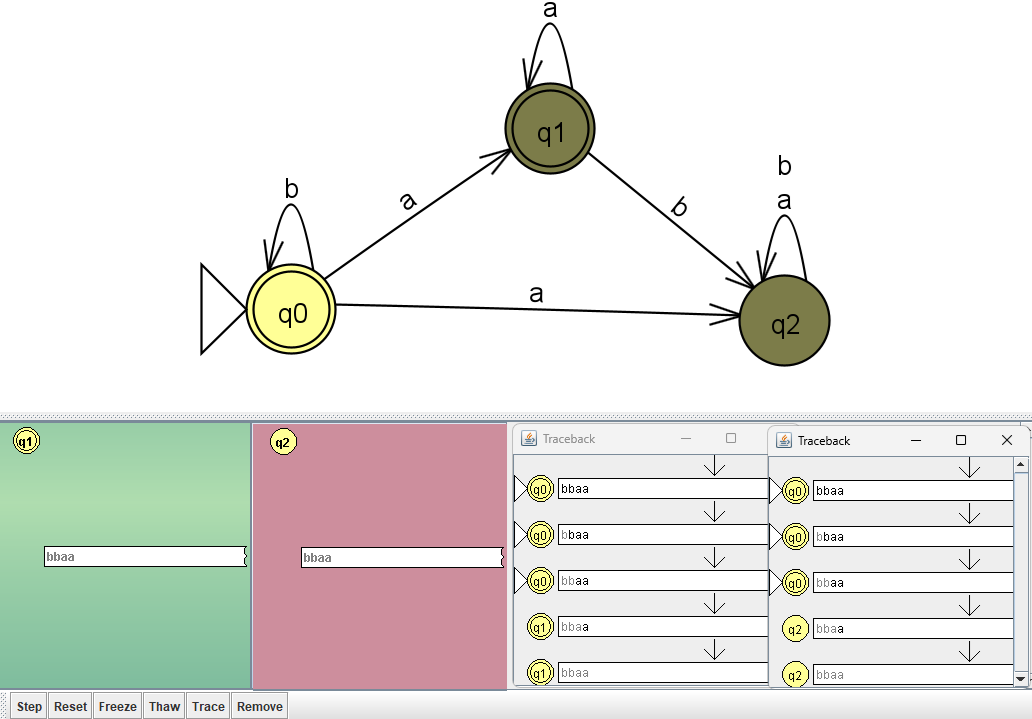


Рисунок 11 – Тест для цепочки «bbaa»

Все тесты были пройдены успешно. На рисунке 12 показан код на C++ для реализации данного НКА.

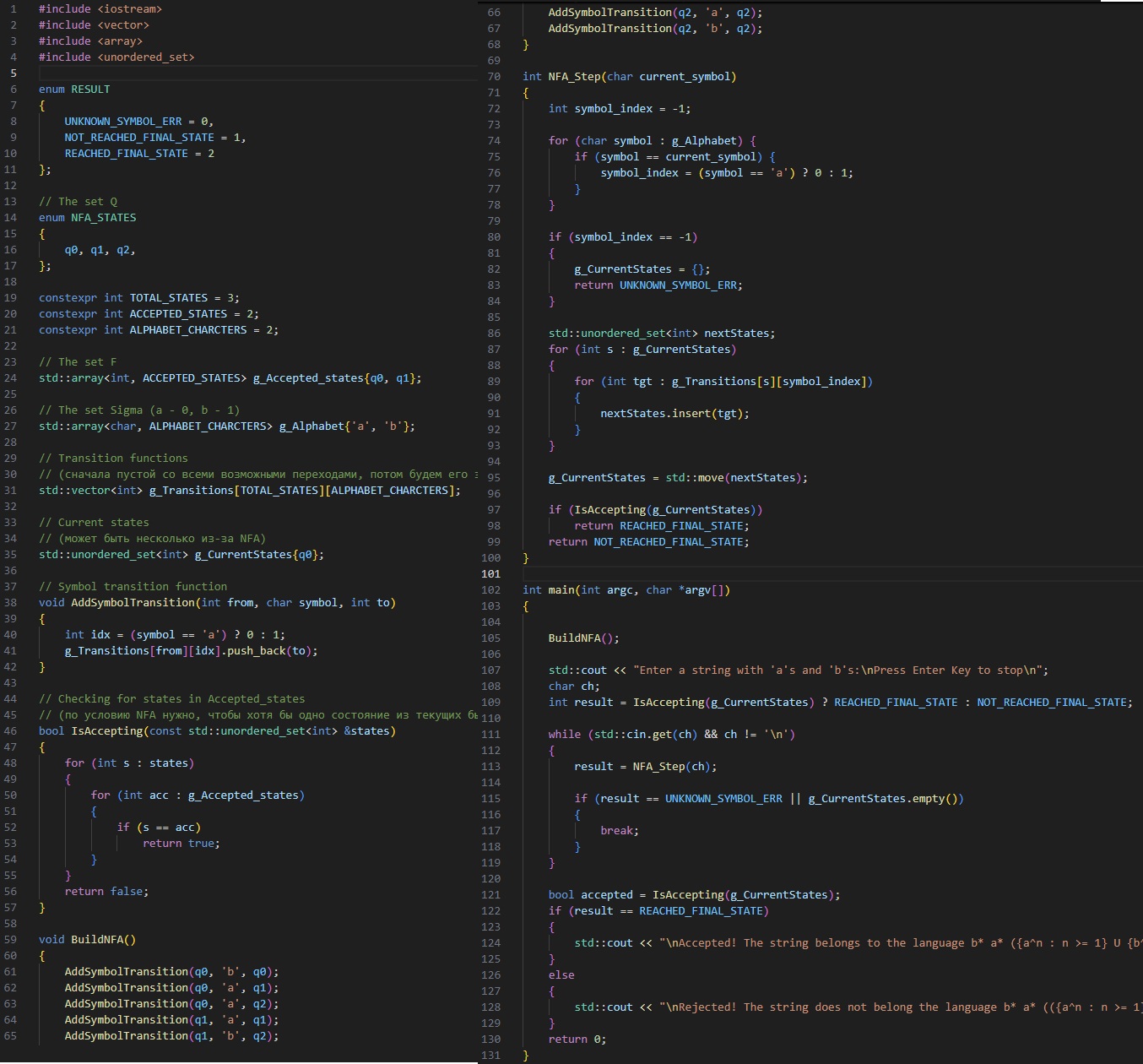


Рисунок 12 – Код для НКА на C++

После компиляции, сборки и запуска программы были повторно проведены тесты, показанные на рисунке 13.

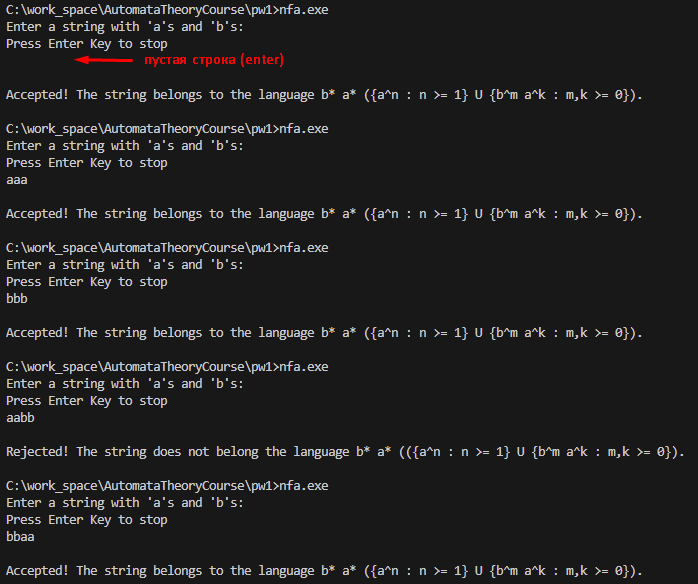


Рисунок 13 – Тесты для НКА на C++

# 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 на С++.