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 | КИ23-17/2Б, 032320597 | |  |  |  | А. А. Тисецкая |
|  | номер группы, зачетной книжки | |  | подпись, дата |  | инициалы, фамилия |

**ХОД РАБОТЫ**

Цель работы:

Реализация и исследование детерминированных и недетерминированных конечных автоматов.

Вариант 7.

а) Построить ДКА, допускающий в алфавите {0, 1} множество всех цепочек, в которых число нулей нацело делится на 5, а число единиц – на 3.

б) Построить НКА с количеством состояний, не превышающим 4, для языка { : n ≥ 0} U { : n ≥ 1}.

Решение:

Пункт а) ДКА:

.

Идея: нужно считать количество нулей по модулю 5 и количество единиц по модулю 3. Создать 15 состояний с остатками деления единиц и нулей. . Начальное состояние , потому что не было считано ни единиц, ни нулей. При чтении нуля переходим в , а при чтении единицы .

Принимающее состояние также , потому что необходимо, чтобы оба остатка были равны нулю.

Каждое состояние автомата описывается парой остатков. Граф-схема автомата представлена на рис.1.

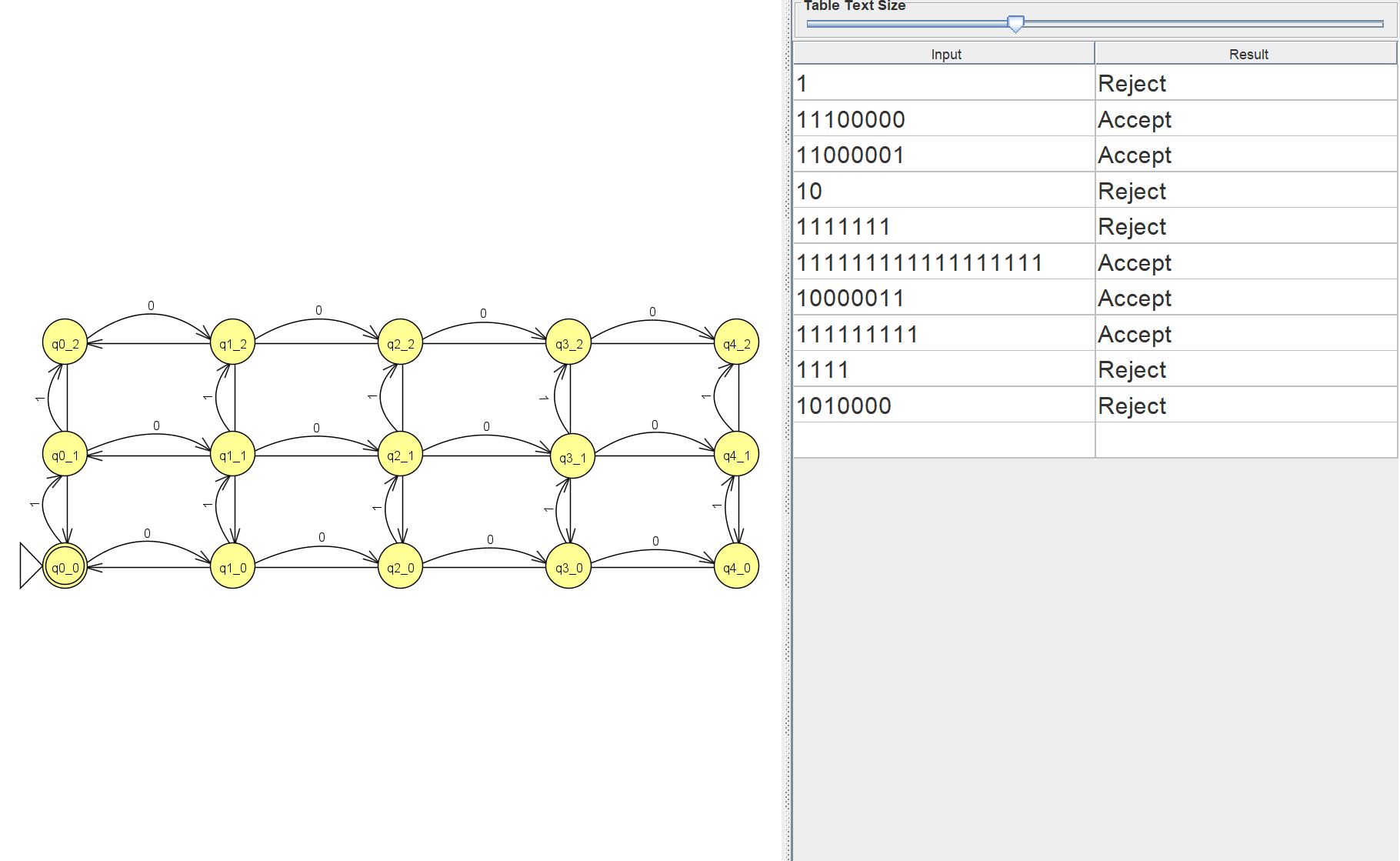


Рисунок 1 – Пример jflap

В основе работы автомата лежит идея хранения состояний в виде остатков от деления количества символов. Каждое состояние кодируется парой (i,j), где i – остаток от числа нулей по модулю 5, а j – остаток от числа единиц по модулю 3. Так как возможных остатков всего 5 и 3 соответственно, общее количество состояний равно 5×3=15.

Для удобства эта пара сводится к одному индексу по формуле

Например, состояние (0,0) соответствует индексу 0 и является начальным и принимающим, (2,1) соответствует индексу 7, а (4,2) – индексу 14.

Переходы между состояниями реализованы через арифметику остатков. Если приходит символ 0, то увеличивается счётчик нулей: новый остаток равен (i+1) mod 5, а остаток по единицам не меняется. Формула перехода имеет вид:

Если приходит символ 1, то остаток по нулям остаётся прежним, а остаток по единицам увеличивается на 1 по модулю 3:

Благодаря этим формулам строится таблица переходов для всех 15 состояний. Конечным состоянием объявляется только (0,0) то есть случай, когда число нулей делится на 5, а число единиц делится на 3. Реализация ДКА на ЯП Java представлена на рис.2



Рисунок 2 – Реализация ДКА на ЯП Java

Вывод программы представлен на рисунке 3 и эквивалентен результату, выдаваемому JFLAP на тех же тестовых цепочках.

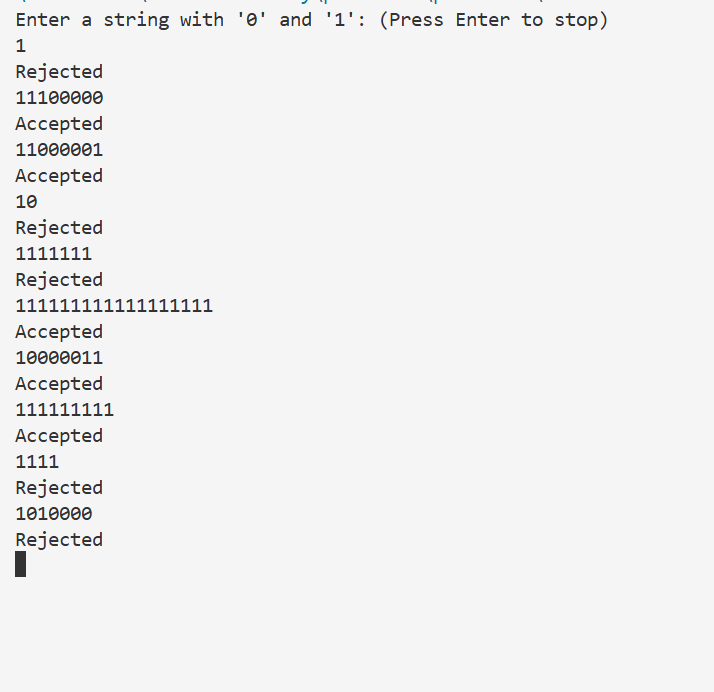


Рисунок 3 – Пример результатов программы

Пункт б) Построить НКА с количеством состояний, не превышающим 4, для языка { : n ≥ 0} U { : n ≥ 1}.

Это все строки только из a (включая пустую) или строки, которые начинаются с одного или больше b, и заканчиваются ровно на одном a.

Состояния.

q0 – стартовое.

q1 – для ветки только a. Финальное.

q2 – для чтения блока b.

q3 – для завершающего a в b^+ a. Финальное.

Переходы.

Из q0 есть два ε-перехода (без символа).

Один в q1 – это ветка a\*.

Второй в q2 – это ветка b^+ a.

Ветка a\*.

В q1 есть петля по символу a.

Так как мы попадаем в q1 через ε-переход, пустая строка сразу принимается – мы находимся в финальном состоянии без чтения символов. Строки a, aa, aaa и так далее тоже принимаются, потому что можно ходить по петле. Если же встретится символ b, путь обрывается, и строка не принимается.

Ветка b^+ a.

В q2 есть петля по символу b, то есть можно прочитать один или несколько b. Затем по символу a происходит переход из q2 в q3.

Состояние q3 финальное и не имеет исходящих переходов, поэтому строка должна заканчиваться ровно на этом a. Граф-схема представлена на рисунке 4.

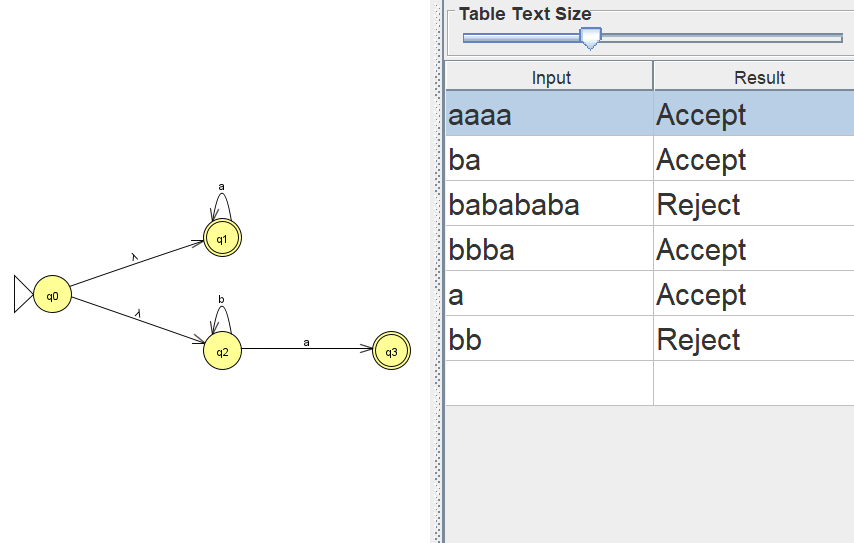


Рисунок 4 – Пример jflap

Код, представленный ниже (рисунки 5-6) реализует недетерминированный конечный автомат с ε-переходами (NFA с epsilon) на Java. Основная идея заключается в том, что автомат может одновременно находиться в нескольких состояниях, а ε-переходы позволяют перемещаться между ними без чтения символа из входной строки. В реализации состояния автомата представлены целыми числами, а переходы хранятся в виде вложенной карты Map<Integer, Map<Character, Set<Integer>>>, где для каждого состояния и символа хранится множество состояний, в которые можно перейти. Метод addTransition добавляет новый переход, создавая множество при необходимости, а epsilonClosure вычисляет множество всех состояний, достижимых из заданного множества через цепочку ε-переходов, используя стек для обхода. Метод move находит все состояния, достижимые из текущих по конкретному символу, а метод accepts последовательно обрабатывает каждый символ входной строки, обновляя текущее множество состояний с учётом обычных и ε-переходов и проверяя наличие принимающих состояний после обработки всей строки.

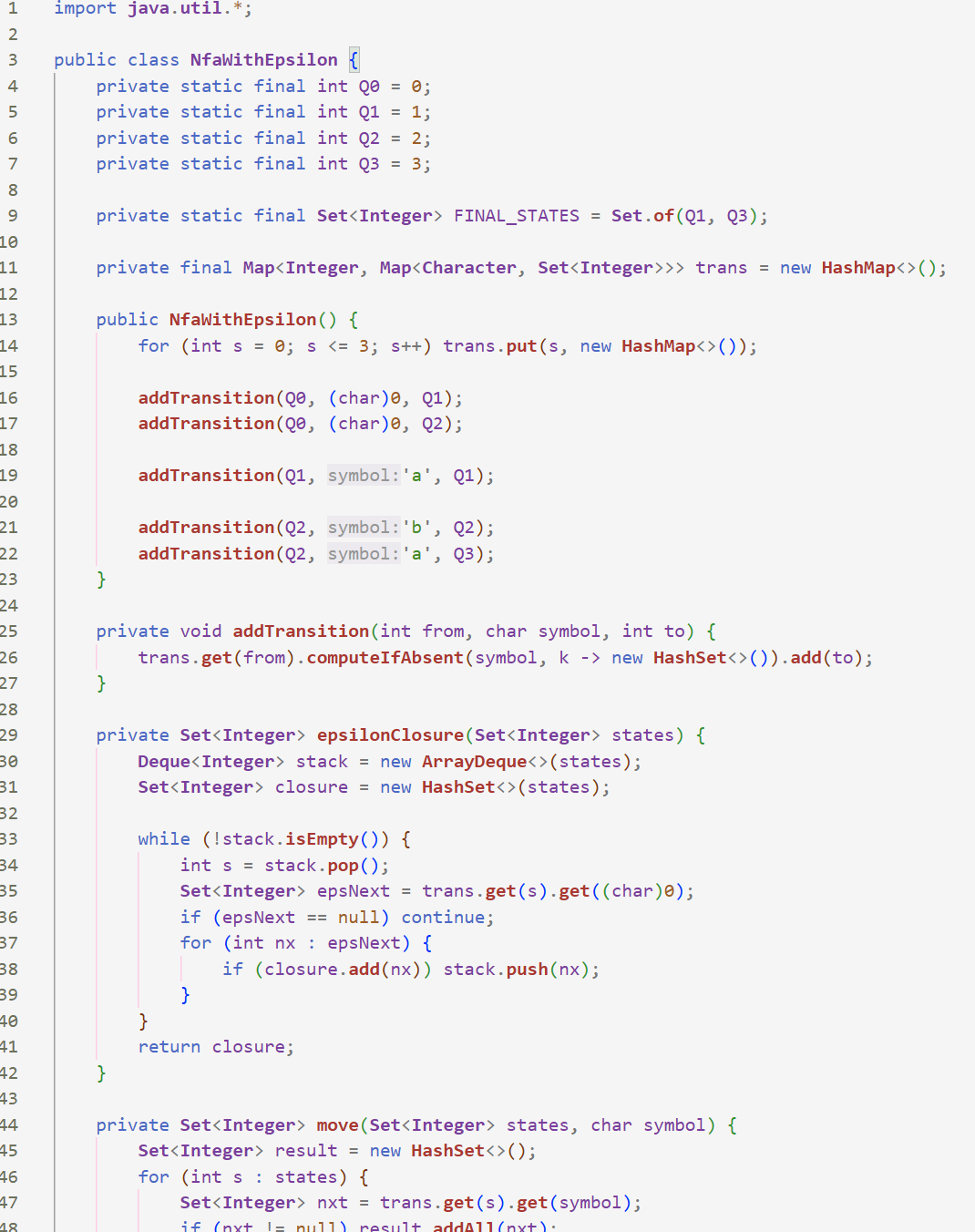


Рисунок 5 – Реализация НКА на ЯП Java

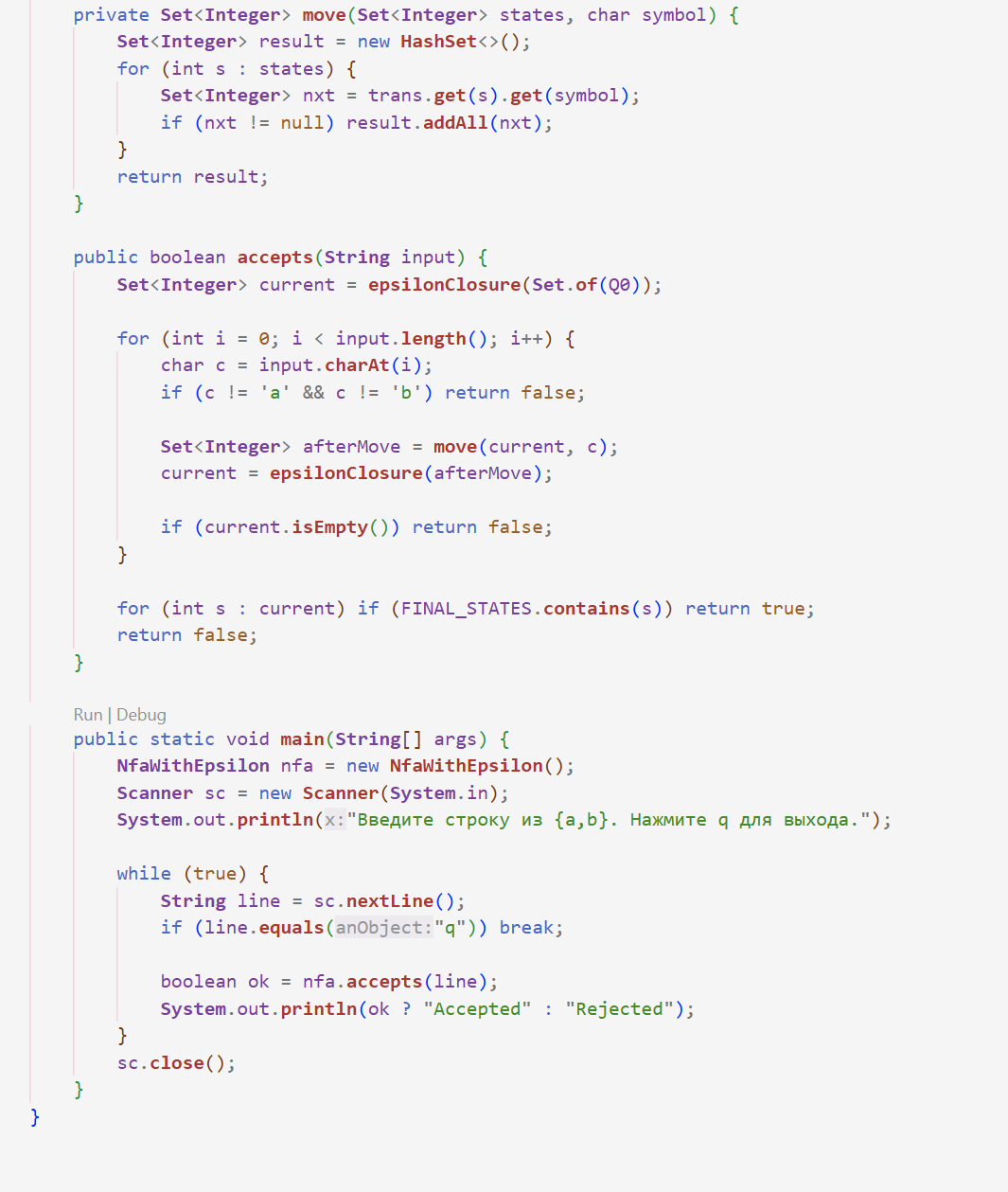


Рисунок 6 – Реализация НКА на ЯП Java

Вывод программы представлен на рисунке 7 и эквивалентен результату, выдаваемому JFLAP на тех же тестовых цепочках.

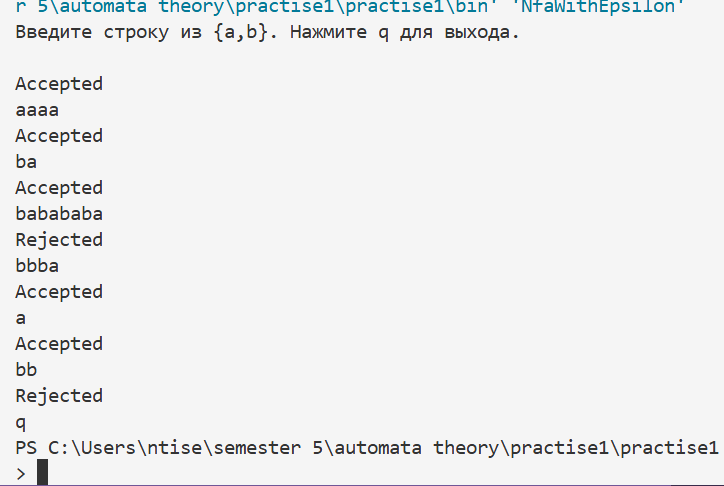


Рисунок 7 – Вывод программы

**ЗАКЛЮЧЕНИЕ**

Поставленные задачи решены: автоматы корректно принимают и отклоняют строки в соответствии с формальными определениями языков; программные реализации демонстрируют практическую применимость построений и служат проверкой корректности диаграмм и таблиц переходов.