Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Пермский национальный исследовательский политехнический университет

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Дискретная математика»

Тема: «Синтезирование автомата»

Семестр 4

Выполнил работу

Студент группы ИВТ-23-2б

Меновщиков Глеб Николаевич

Проверил

старший преподаватель кафедры ИТАС

Рустамханова Гульшат Ильдаровна

Пермь 2025

**Постановка задачи**

Синтезировать автомат, распознающий заданный язык. Написать программу анализатор, которая определяет, принадлежит ли слово заданному языку.

**Варианты:**

Формальный язык задан в алфавите {𝑎, 𝑏, 𝑐, 𝑑} и содержит слова любой длины

**Вариант 7:**

В которых гласные и согласные буквы перемежаются

**Диаграмма Мура и таблица переходов**

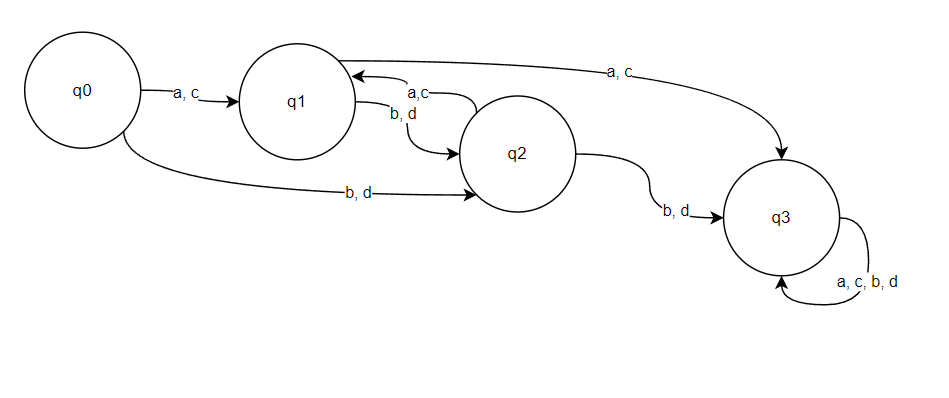


Рисунок 1 - Диаграмма Мура

Таблица переходов:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **вход\←состояние** | **q0 / T** | **q1 / T** | **q2 / T** | **q3 / F** |
| **a** (нечет) | q1 | q3 | q1 | q3 |
| **b** (чет) | q2 | q2 | q3 | q3 |
| **c** (нечет) | q1 | q3 | q1 | q3 |
| **d** (чет) | q2 | q2 | q3 | q3 |

Регулярное выражение:

V=(a∣c), C=(b∣d).

L=((VC) ∗ (ε ∪ V)) ∪ ((CV) ∗ (ε ∪ C)).

**Код программы**

using System;

using System.Collections.Generic;

class AlternatingVC

{

// Задаём множества гласных и согласных:

static readonly HashSet<char> Vowels = new HashSet<char> { 'a', 'c' };

static readonly HashSet<char> Consonants = new HashSet<char> { 'b', 'd' };

enum State { Start, VowelState, ConsonantState, Dead }

static bool IsAlternating(string s)

{

State state = State.Start;

foreach (char ch in s)

{

switch (state)

{

case State.Start:

if (Vowels.Contains(ch))

state = State.VowelState;

else if (Consonants.Contains(ch))

state = State.ConsonantState;

else

state = State.Dead;

break;

case State.VowelState:

if (Consonants.Contains(ch))

state = State.ConsonantState;

else

state = State.Dead;

break;

case State.ConsonantState:

if (Vowels.Contains(ch))

state = State.VowelState;

else

state = State.Dead;

break;

case State.Dead:

// остаёмся в ловушке

state = State.Dead;

break;

}

if (state == State.Dead)

break;

}

// принимаем, если не в ловушке и хотя бы один символ

return (state == State.VowelState || state == State.ConsonantState) && s.Length > 0;

}

static void Main()

{

Console.WriteLine("Введите слово над {a,b,c,d}:");

string input = Console.ReadLine().Trim();

bool ok = IsAlternating(input);

Console.WriteLine(ok

? "Слово принадлежит языку (гласные/согласные чередуются)."

: "Слово НЕ принадлежит языку.");

}

}

**Результаты работы программы**

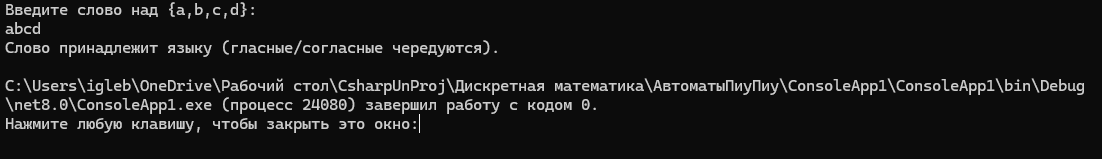
****

Рисунок 2 - результаты работы программы