

Индивидуальное задание №1

Проектирование и реализация конечного распознавателя

Патычкина Елизавета Вадимовна, КТб01-10

Вариант 1-9

1. Постановка задачи

Построить детерминированный конечный автомат, распознающий все такие цепочки из алфавита $\{0, 1, 2\}$, в которых каждый из символов встречается нечетное количество раз.

2. Словесное описание автомата

Функция `machine_insert` инициализирует таблицу переходов. Автомат запрашивает у пользователя входное слово. Функция `check` проверяет введенное слово на корректность. В случае введения символа, который не содержится в алфавите автомата, он предлагает начать заново работу. В случае, когда слово введено верно, автомат меняет свое состояние в зависимости от текущего состояния и текущего символа из входного слова. Функция `CharacterProcessing` посимвольно обрабатывает слово.

q_0 – начальное состояние, четное количество всех элементов входного алфавита.

q_1 – нечетное количество элемента 0, четное количество элементов 1 и 2.

q_2 - нечетное количество элемента 1, четное количество элементов 0 и 2.

q_3 - нечетное количество элемента 2, четное количество элементов 0 и 1.

q_4 - нечетное количество элементов 0 и 1, четное количество элемента 2.

q_5 - нечетное количество элементов 0 и 2, четное количество элемента 1.

q_6 - нечетное количество элементов 1 и 2, четное количество элемента 0.

q_7 - нечетное количество элементов 0, 1 и 2.

В результате автомат выдает строку “Yes”, если конечное состояние является допускающим, или “No”, если конечное состояние не является допускающим. В конце работы автомат предлагает пользователю повторить работу.

3. Описание распознавателя

$V = \{0, 1, 2\}$ – входной алфавит.

$K = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$ – множество состояний.

M – функция переходов. Таблица 1 – таблица переходов автомата.

Рисунок 1 – диаграмма переходов автомата.

Состояние /символ	0	1	2	Y/N
q_0	q_1	q_2	q_3	N
q_1	q_0	q_4	q_5	N
q_2	q_4	q_0	q_6	N
q_3	q_5	q_6	q_0	N
q_4	q_2	q_1	q_7	N
q_5	q_3	q_7	q_1	N
q_6	q_7	q_3	q_2	N
q_7	q_6	q_5	q_4	Y

Таблица 1

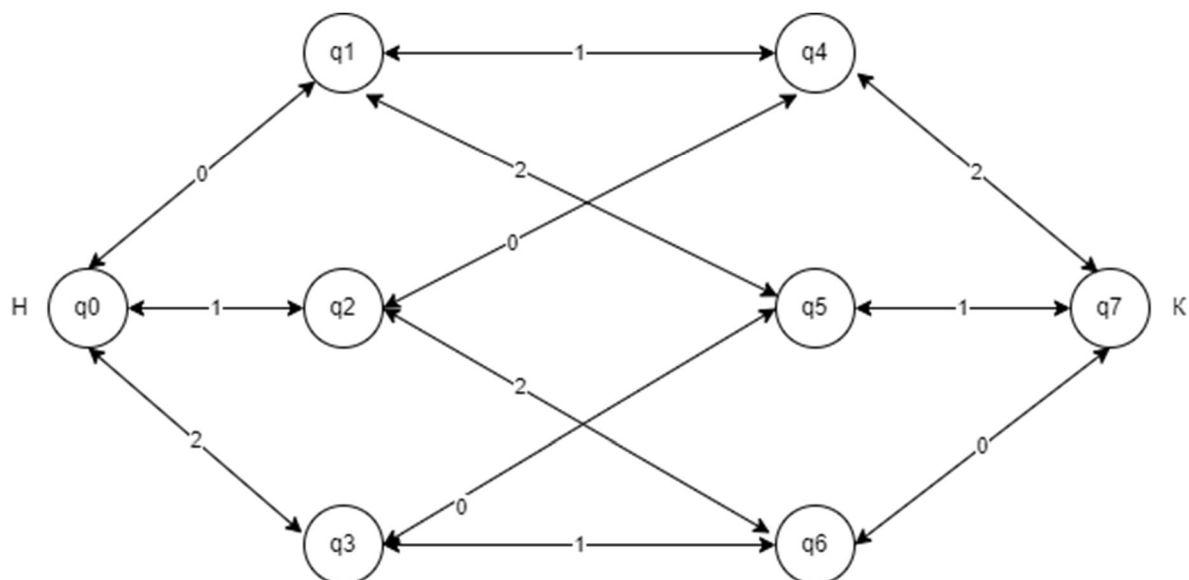


Рисунок 1

$S = \{q_0\}$ – множество начальных состояний.

$Z = \{q_7\}$ – множество заключительных состояний.

4. Тестирование автомата

1) 010210122

0 $q_0 \rightarrow q_1$
 1 $q_1 \rightarrow q_4$
 0 $q_4 \rightarrow q_2$
 2 $q_2 \rightarrow q_6$
 1 $q_6 \rightarrow q_3$
 0 $q_3 \rightarrow q_5$
 1 $q_5 \rightarrow q_7$
 2 $q_7 \rightarrow q_4$
 2 $q_4 \rightarrow q_7$

Yes

2) 012

0 $q_0 \rightarrow q_1$
 1 $q_1 \rightarrow q_4$

2 $q_4 \rightarrow q_7$

Yes

3) 0 (1, 2)

0 $q_0 \rightarrow q_1$

No

4) 123 Строка содержит символы, которые не соответствуют алфавиту автомата

5) 0112010

0 $q_0 \rightarrow q_1$

1 $q_1 \rightarrow q_4$

1 $q_4 \rightarrow q_1$

2 $q_1 \rightarrow q_5$

0 $q_5 \rightarrow q_3$

1 $q_3 \rightarrow q_6$

0 $q_6 \rightarrow q_7$

Yes

6) 111222

1 $q_0 \rightarrow q_2$

1 $q_2 \rightarrow q_0$

1 $q_0 \rightarrow q_2$

2 $q_2 \rightarrow q_6$

2 $q_6 \rightarrow q_2$

2 $q_2 \rightarrow q_6$

No

7) 01202211

0 $q_0 \rightarrow q_1$

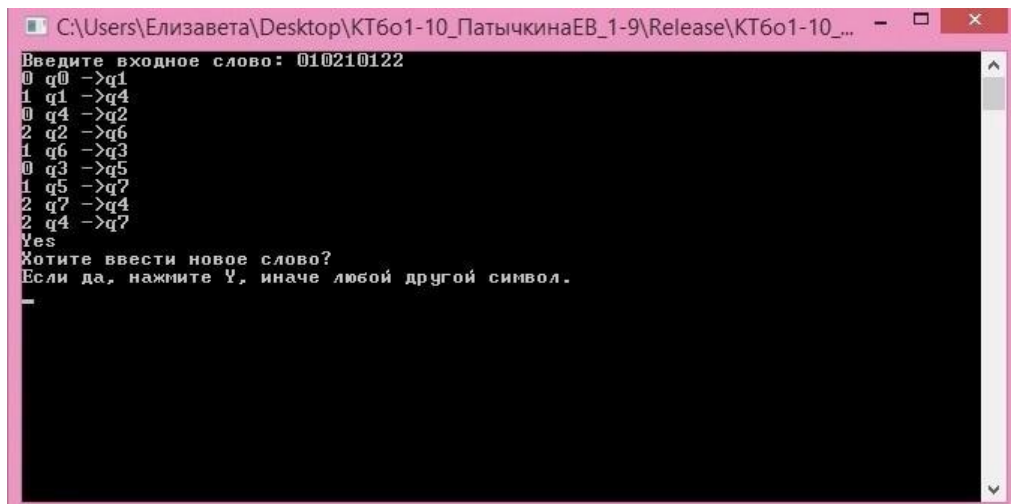
1 $q_1 \rightarrow q_4$

2 $q_4 \rightarrow q_7$

0 q7 -> q6
2 q6 -> q2
2 q2 -> q6
1 q6 -> q3
1 q3 -> q6
No

5. Прогон автомата на тестах

- 1) Тест 1
- 2) Тест 2
- 3) Тест 3
- 4) Тест 4
- 5) Тест 5
- 6) Тест 6
- 7) Тест 7



```
C:\Users\Елизавета\Desktop\КТ6о1-10_ПатычкинаЕВ_1-9\Release\КТ6о1-10_...
Введите входное слово: @10210122
0 q0 -> q1
1 q1 -> q4
0 q4 -> q2
2 q2 -> q6
1 q6 -> q3
0 q3 -> q5
1 q5 -> q7
2 q7 -> q4
2 q4 -> q7
Yes
Хотите ввести новое слово?
Если да, нажмите Y, иначе любой другой символ.
_
```

Тест 1

```
C:\Users\Елизавета\Desktop\КТ6o1-10_ПатычкинаЕВ_1-9\Release\КТ6o1-10_...
Введите входное слово: 012
0 q0 ->q1
1 q1 ->q4
2 q4 ->q7
Yes
Хотите ввести новое слово?
Если да, нажмите Y, иначе любой другой символ.
```

Тест 2

```
C:\Users\Елизавета\Desktop\КТ6o1-10_ПатычкинаЕВ_1-9\Release\КТ6o1-10_...
Введите входное слово: 0
0 q0 ->q1
No
Хотите ввести новое слово?
Если да, нажмите Y, иначе любой другой символ.
Y
Введите входное слово: 1
1 q0 ->q2
No
Хотите ввести новое слово?
Если да, нажмите Y, иначе любой другой символ.
Y
Введите входное слово: 2
2 q0 ->q3
No
Хотите ввести новое слово?
Если да, нажмите Y, иначе любой другой символ.
```

Тест 3

```
C:\Users\Елизавета\Desktop\КТ6o1-10_ПатычкинаЕВ_1-9\Release\КТ6o1-10_...
Введите входное слово: 123
Строка содержит символы, которые не соответствуют алфавиту автомата.
Хотите ввести новое слово?
Если да, нажмите Y, иначе любой другой символ.
```

Тест 4

```
C:\Users\Елизавета\Desktop\КТ6o1-10_ПатычкинаЕВ_1-9\Release\КТ6o1-10_...
Введите входное слово: 0112010
0 q0 ->q1
1 q1 ->q4
1 q4 ->q1
2 q1 ->q5
0 q5 ->q3
1 q3 ->q6
0 q6 ->q7
Yes
Хотите ввести новое слово?
Если да, нажмите Y, иначе любой другой символ.
-
```

Тест 5

```
C:\Users\Елизавета\Desktop\КТ6o1-10_ПатычкинаЕВ_1-9\Release\КТ6o1-10_...
Введите входное слово: 111222
1 q0 ->q2
1 q2 ->q0
1 q0 ->q2
2 q2 ->q6
2 q6 ->q2
2 q2 ->q6
No
Хотите ввести новое слово?
Если да, нажмите Y, иначе любой другой символ.
-
```

Тест 6

```
C:\Users\Елизавета\Desktop\КТ6o1-10_ПатычкинаЕВ_1-9\Release\КТ6o1-10_...
Введите входное слово: 01202211
0 q0 ->q1
1 q1 ->q4
2 q4 ->q7
0 q7 ->q6
2 q6 ->q2
2 q2 ->q6
1 q6 ->q3
1 q3 ->q6
No
Хотите ввести новое слово?
Если да, нажмите Y, иначе любой другой символ.
-
```

Тест 7

6. Структура данных для хранения системы переходов автомата

Для хранения системы переходов автомата используется структура `machine`. В нее входит таблица переходов (двумерный массив) размерностью 8×3 . Структура `map` (состояние и буква, характеризующая допуск или недопуск соответствующему состоянию), хранящая данные о допустимости или не допустимости конкретного состояния автомата. Две целочисленные переменные `state_start` и `state_finish`. Первая переменная хранит начальное состояние, а вторая конечное состояние при текущем обрабатываемом символе.

7. Словесное описание идеи программной реализации одного шага работы автомата

Рассмотрим вводное слово: 010210122

0 `q0 -> q1` нечетное количество элемента 0, четное количество элементов 1 и 2

1 `q1 -> q4` нечетное количество элементов 0 и 1, четное количество элемента 2

0 `q4 -> q2` нечетное количество элемента 1, четное количество элементов 0 и 2

2 `q2 -> q6` нечетное количество элементов 1 и 2, четное количество элемента 0

1 `q6 -> q3` нечетное количество элемента 2, четное количество элементов 0 и 1

0 `q3 -> q5` нечетное количество элементов 0 и 2, четное количество элемента 1

1 `q5 -> q7` нечетное количество элементов 0, 1 и 2

2 q7 -> q4 нечетное количество элементов 0 и 1, четное количество элемента 2

2 q4 -> q7 нечетное количество элементов 0, 1 и 2

Заключительное состояние q7 является допускающим, следовательно выводим “Yes”.

8. Листинг программы

```
//ЮФУ, ИКТИБ, МОП ЭВМ
//Программирование и основы теории алгоритмов
//Индивидуальное задание №1
//Проектирование и реализация конечного распознавателя
//КТб01-10, Патычкина Елизавета Вадимовна

#include <iostream>
#include <vector>
#include <map>
#include <string>
#define row 8
#define column 3
using namespace std;

void machine_insert(int m[row][column], map <int, char> &YN);

bool check(string line);

int CharacterProcessing(int t[row][column], char c, int state);

struct machine
{
    int table[row][column];
    map <int, char> YN;
    int state_start = 0;
    int state_finish = 0;
};

int main()
{
    setlocale(LC_ALL, "Russian");
    struct machine mymachine;
    string answer;
    string line;
    machine_insert(mymachine.table, mymachine.YN);
    do
    {
        cout << "Введите входное слово: ";
        cin >> line;
        if (check(line) == false)
        {
            cout << "Строка содержит символы, которые не соответствуют алфавиту автомата." << endl;
        }
        else
        {
            for (int i = 0; i < line.size(); i++)
            {
```

```

        mymachine.state_finish = CharacterProcessing(mymachine.table,
line[i], mymachine.state_start);
        cout << line[i] << ' ' << 'q' << mymachine.state_start << ' '
<< "->" << 'q' << mymachine.state_finish << endl;
        mymachine.state_start = mymachine.state_finish;
    }
    int FinalState = mymachine.state_finish;
    if (mymachine.YN[FinalState] == 'Y')
        cout << "Yes" << endl;
    else
        cout << "No" << endl;
}
mymachine.state_start = 0;
cout << "Хотите ввести новое слово?" << '\n' << "Если да, нажмите Y, иначе
любой другой символ." << endl;
cin >> answer;
} while (answer == "Y");
}

```

```

//Входные параметры: int m[row][column] - таблица переходов
//                      map <int, char> &YN - ссылка на структуру map с допускающими/не
допускающими состояниями
//Функция: заполняет таблицу переходов и структуру допускающими/не допускающими
состояниями

```

```

//Выходные параметры: функция ничего не возвращает
void machine_insert(int m[row][column], map <int, char> &YN)
{

```

```

    m[0][0] = 1;
    m[0][1] = 2;
    m[0][2] = 3;
    m[1][0] = 0;
    m[1][1] = 4;
    m[1][2] = 5;
    m[2][0] = 4;
    m[2][1] = 0;
    m[2][2] = 6;
    m[3][0] = 5;
    m[3][1] = 6;
    m[3][2] = 0;
    m[4][0] = 2;
    m[4][1] = 1;
    m[4][2] = 7;
    m[5][0] = 3;
    m[5][1] = 7;
    m[5][2] = 1;
    m[6][0] = 7;
    m[6][1] = 3;
    m[6][2] = 2;
    m[7][0] = 6;
    m[7][1] = 5;
    m[7][2] = 4;
    YN[0] = 'N';
    YN[1] = 'N';
    YN[2] = 'N';
    YN[3] = 'N';
    YN[4] = 'N';
    YN[5] = 'N';
    YN[6] = 'N';
    YN[7] = 'Y';
}

```

```

//Входные параметры: string line - введенная пользователем строка(слово)
//Функция: проверяет строку на верность введенных символов
//Выходные параметры: функция возвращает ИСТИНА, если все символы удовлетворяют алфавиту
автомата

```

```

//                                     возвращает ЛОЖЬ, если хотя бы один символ не удовлетворяют
// алфавиту автомата
bool check(string line)
{
    for (int i = 0; i < line.size(); i++)
    {
        if ((line[i] == '0') || (line[i] == '1') || (line[i] == '2'))
        {
        }
        else
            return false;
    }
    return true;
}

//Входные параметры: int t[row][column] - таблица переходов
//                  char c - рассматриваемый символ строки
//                  int state - текущее начальное состояние
//Функция: обрабатывает один символ входной строки, определяет новое состояние,
//          определяет конечное состояние
//Выходные параметры: int t[state][num] - новое конечное состояние
int CharacterProcessing(int t[row][column], char c, int state)
{
    int num;
    if (c == '0')
        num = 0;
    else
    {
        if (c == '1')
            num = 1;
        else
        {
            if (c == '2')
                num = 2;
        }
    }
    return t[state][num];
}

```