

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Ataskaita
Reed-Muller kodas

Darbą atliko:
4 kurso, 4 grupės studentas
Marijus Laucevičius

Vilnius
2018 m.

Programos paleidimas

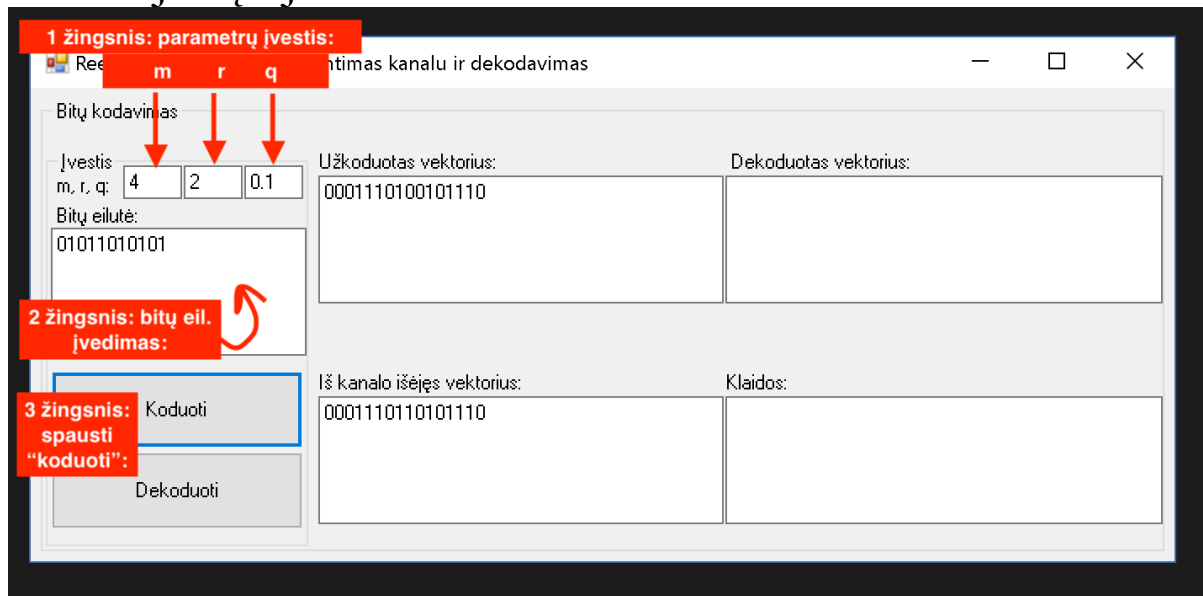
Programos paleidimo (*.exe) failas patalpintas Projekto direktorijoje tokiu adresu: /obj/Debug/Reed-Muller-marijuslau.exe

- Bitų eilutės kodavimas:
 - Bitų eilutės kodavimo lange reikia įvesti norimus m, r ir q parametrus. Taip pat reikia įvesti norimą užkoduoti eilutę. Pvz.: (1001, 1111111, 00010101 ir t.t). Neįvedus šių parametrų arba įvedus juos nekorektiškai (pvz. prirašius raidžių vietoje skaičių), bandant naudotis programa, ji parodys pranešimo langą su įspėjimu. Taip pat bus parodytas pranešimas, jei su nurodytais parametrais eilutės užkoduoti neįmanoma.
 - Įvedus tinkamus ar netinkamus parametrus spaudžiamas mygtukas “Koduoti”.
 - Jei parametrai netinkami įvyks jau minėti scenarijai, t.y. bus parodyti pranešimo langai.
 - Jei parametrai tinkami ir tokia eilutę galima užkoduoti, tada užkoduota eilutė pasirodys lange “Užkoduotas vektorius”. Taip pat lange “Iš kanalo išėjęs vektorius” bus parodytas užkoduotas vektorius po siuntimo galimai triukšmingu kanalu.
 - Dekodavimas kol kas neįgyvendintas. T.y. yra tik užkodavimas ir siuntimas kanalu.
- Teksto lauko kodavimas:
 - Neįgyvendinta
- Paveiksliuko kodavimas:
 - Neįgyvendinta

Failai ir jų paskirtys

1. (.../Reed-Muller-marijuslau/Program.cs) – Programos įėjties taškas, su Main funkcija.
2. (.../Reed-Muller-marijuslau/MainForm.cs) – Pagrindinės formos failas, šiame faile realizuota saveika su ja.
3. (.../Reed-Muller-marijuslau/MainForm.designer.cs) – Pagrindinės formos grafinis vaizdas.
4. (.../Reed-Muller-marijuslau/GenMatrixGenerator.cs) – Rydo-Miulerio kodo generuojančios funkcijos sugeneravimas, naudojantis parametrais iš pagrindinės formos
5. (.../Reed-Muller-marijuslau/Encoder.cs) – Bitų eilutės užkodavimo funkcionalumas.
6. (.../Reed-Muller-marijuslau/Channel.cs) – Siuntimas kanalu.
7. (.../Reed-Muller-marijuslau/Helpers.cs) – Pagalbinės funkcijos naudojamos programoje.

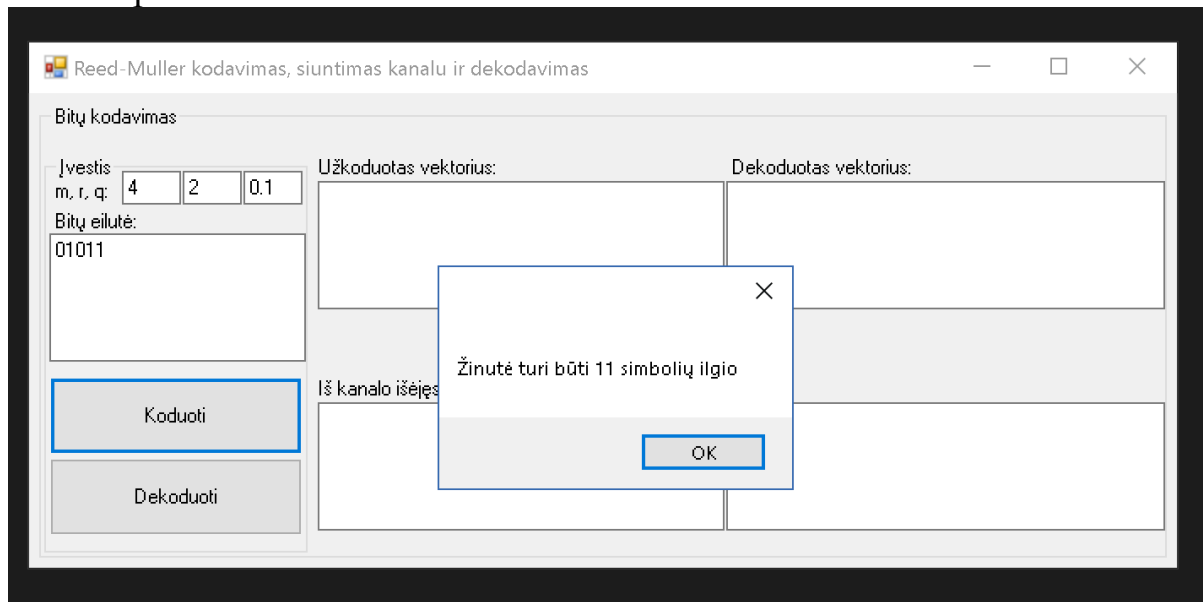
Vartotojo sąsaja



Vartotojo sąsajoje įgyvendintas tik bitų užkodavimas.

Galima pamatyti užkoduotą vektorių, bei išsiųstą kanalu vektorių.

Klaidos pranešimas:



Pritaikyti sprendimai

- 1-ojo laipsnio vektoriai generuojančiajai RM matricai gaunami ganėtinai paprastai:

```
// Funkcija grąžinanti 1-ojo laipsnio vektorius
int[] GetIntVector(int length, int take)
{
    int[] vector = new int[length];
    int index = 0;

    int i = 0;
    while (i < length)
    {
        for (int j = 0; j < take; j++)
        {
            vector[index] = 1;
            index++;
        }
        for (int j = 0; j < take; j++)
        {
            vector[index] = 0;
            index++;
        }
        i += take * 2;
    }

    return vector;
}
```

- Vektoriai nuo 2-ojo lygio ir gaunami naudojant funkciją GetAdditionalVector(...); Ši funkcija pasinaudodama kita (VectorsAnd(...)) lygina jau turimus vektorius ir kai gaunamas naujas patikrinimas ar jis yra unikalus, palyginant su jau visais turimais. Jei taip, jis įdedamas į matricą.

```
// Funkcija skirta gauti vektoriams nuo 2-ojo lygio
int[] GetAdditionalVector(int lenght, int startPoint)
{
    int[] vector = null;
    int currLenght = lenght;
    for (int i = startPoint; i < currLenght; i++)
    {
        if (i > startPoint)
        {
            StartFlag = true;
            Console.WriteLine("Start point:" + startPoint);
            return vector;
        }
        for (int j = i + 1; j < currLenght; j++)
        {
            vector = VectorsAnd(Matrix[i], Matrix[j]);
            if (!IsThereAnEqual(vector))
                return vector;
        }
        vector = null;
    }
    return vector;
}
```

- Įvestosios bitų eilutės dauginimas su generuojančiąja matrica:

```
// Vektoriaus * matricos daugybos funkcija
private int[][] VectorMultiply(string bitString, int[][] genMatrix, int dimension, int lenght)
{
    int[][] multipliedMatrix = new int[dimension][];
    int index = 0;
    foreach(char a in bitString)
    {
        multipliedMatrix[index] = new int[lenght];
        int newMatrixIndex = 0;
        foreach(int j in genMatrix[index])
        {
            multipliedMatrix[index][newMatrixIndex] = (int)char.GetNumericValue(a) * j;
            newMatrixIndex++;
        }
        index++;
    }
    return multipliedMatrix;
}
```

Programos testavimas

Vektoriaus (bitų eilutės testavimas):

1- $m=3$, $r=1$, $q=0.05$ bitų eilutė = 1110

Užkoduotas vektorius:

11000011

Išsiųstas kanalu vektorius:

11001011

2- $m=5$, $r=2$, $q=0.1$, bitų eilutė=1100110011001100

Užkoduotas vektorius:

011010101110000001111001101011001

Išsiųstas kanalu vektorius:

111010101110001001101001101011001

3- $m=3$, $r=3$, $q=0.01$, bitų eilutė=11110000

Užkoduotas vektorius:

01101001

Išsiųstas kanalu vektorius:

01101001

Literatūros sąrašas

- [Sta02] [V.Stakėnas. Kodavimo teorija. Paskaitų kursas, 2002.](#)
- [Sta07] [V. Stakėnas. Kodai ir šifrai. Vilnius, 2007.](#)