VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Ataskaita

Rydo-Miulerio (Reed, Muller) kodas RM(m,r)

Atliko: 3 kurso, 5 grupės studentas
Ignas Bradauskas

Vilnius
2017

Programos paleidimas ir naudojimas

Programa paleidimo (*.exe) failo nuoroda:
(…\Kodavimas\Kodavimas\bin\Debug\Kodavimas.exe).

- Paleisti programą pakanka paleisti Kodavimas.exe failiuką
- Bitų eilutės užkodavimas:
    - Norit užkoduoti bitų eilutę reikia atsidaryti „Bitų eilės kodavimas" skiltį, suvesti m, r, q parametrus ir tinkamo dydžio bitų eilutę. Įvedus netinkamo ilgio bitų eilutę ir paspaudus „Koduoti" programa išmes pranešimą kokio ilgio turi būti pradinė bitų eilutė. q parametrą reikia įvesti 0.*** formatu, bitų eilutės lauke programa tikisi, kad bus įvesta tik vienetai ir nuliai.
    - Įvedus reikiamo dydžio bitų eilutę spaudžiamas mygtukas „Koduoti".
    - Turėtų atitinkamuose laukuose pasirodyti užkoduota bitų eilutė, įvykusios klaidos ir klaidų įvykimo vietos.
    - Skiltyje „Iš kanalo išėjęs vektorius" galima įdėti ar sumažinti klaidų. Čia taip pat programa tikisi, kad bus įvesta tik vienetai ir nuliai.
    - Tuomet reikia spausti „Dekoduoti". Programa bandys ištaisyti klaidas ir rezultatas bus parodytas „Dekoduotas vektorius:" skiltyje.
- Teksto lauko užkodavimas:
    - Norit užkoduoti teksto lauką reikia atsidaryti „Teksto lauko kodavimas" skiltį, suvesti m, r, q parametrus ir tinkamo dydžio teksto lauką. q parametrą reikia įvesti 0.*** formatu. Rekomenduojami parametrai: m = 5; r = 1; q = 0.1; Tekstas = kodavimas yra jega(1 pav.).
    - Tuomet reikis spausti mygtuką „Pradėti kodavimą".
- Paveikslėlio užkodavimas:
    - Šis funkcionalumas realizuotas dalinai. Pavyksta persiųsti užkoduotą paveikslėlį ir jį atkoduoti, bet neina atvaizduoti pagadinto paveikslėlio.
    - Norint įkelti ir persiųsti paveikslėlį reikia atsidaryti skiltį „Paveikslėlio kodavimas", įrašyti parametrus (būtinai pirma) ir paspausti „Įkelti".
    - Pasirinkti kuo mažesnį paveikslėlio tipo failą ir paspausti Open.

Pradinių tekstų failai ir jų paskirtis

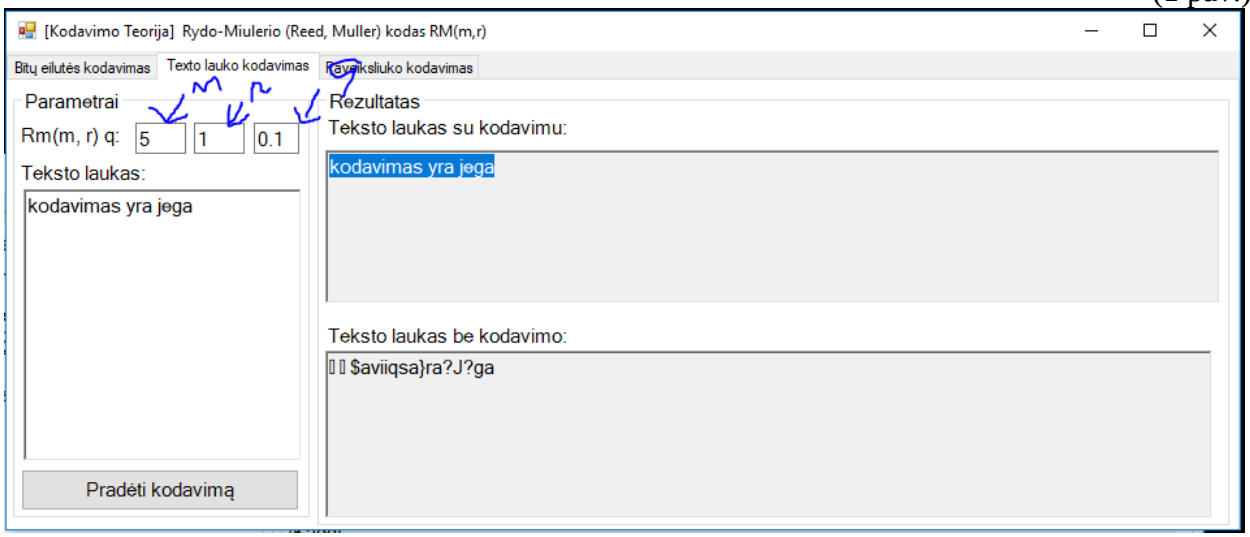
Čia bus surašyti pradiniai tekstų failai ir jų paskirtys:

1. (…\Kodavimas\Kodavimas\Channel.cs) – realizuotas pranešimo siuntimas kanalu.
2. (…\Kodavimas\Kodavimas\Decoder.cs) – realizuotas pranešimo dekodavimas.
3. (…\Kodavimas\Kodavimas\Encoder.cs) – realizuotas pranešimo užkodavimas.
4. (…\Kodavimas\Kodavimas\GMatrix.cs) – sukurtas, kad būtų galima kurti G-matricos objektus ir patogiau ja naudotis.
5. (…\Kodavimas\Kodavimas\GMatrixGenerator.cs) – realizuotas G-matricos sukūrimas.
6. (…\Kodavimas\Kodavimas\MainForm.cs) – realizuota sąveika su grafine sąsaja ir kitomis klasėmis.
7. (…\Kodavimas\Kodavimas\MainForm.Designer.cs) – realizuotas grafinės sąsajos vaizdas.
8. (…\Kodavimas\Kodavimas\PictureCoding.cs) – Bandyta realizuoti paveiksliuko suskaldymą į gabaliukus ir sutvėrimą atgal iš jų.
9. (…\Kodavimas\Kodavimas\Program.cs) – Pradinis programos taškas su Main funkcija.
10. (…\Kodavimas\Kodavimas\TextFieldCoding.cs) – Realizuotas teksto skaidymas į bitų eilites ir sutvėrimas atgal iš jų.
11. (…\Kodavimas\Kodavimas\Utils.cs) – Statinė klasė kurioje realizuota keletas metodų kurių prireikė ar gali prireikti įvairiuose klasėse.

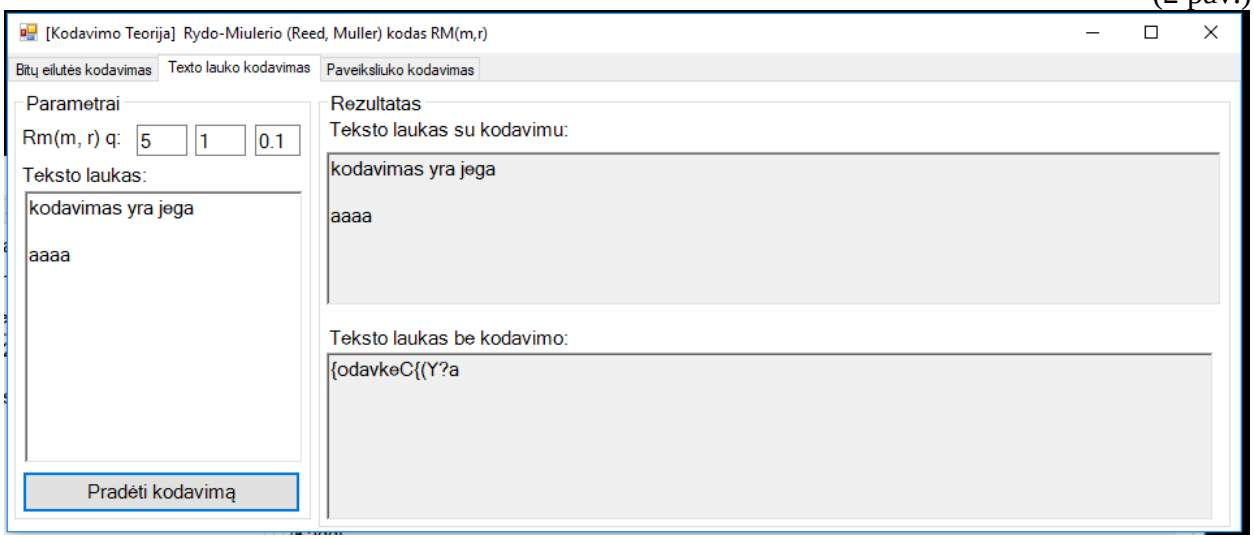
Vartotojo sąsajos aprašymas

Teksto lauko kodavimo pavyzdžiai:
Q yra klaidos tikimybė kiekvienam bitui.

(1 pav.)



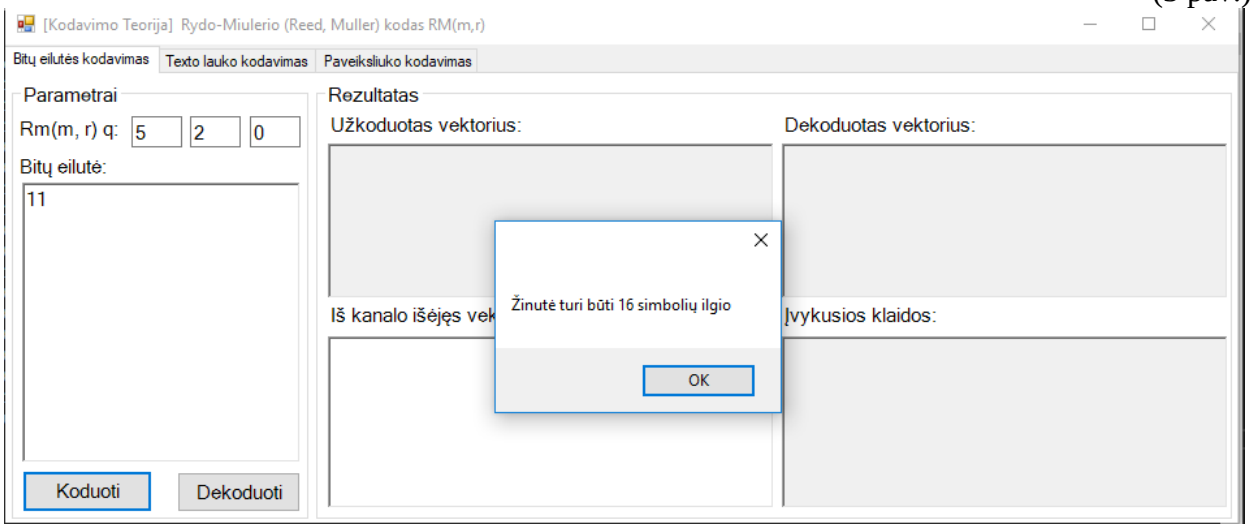
(2 pav.)



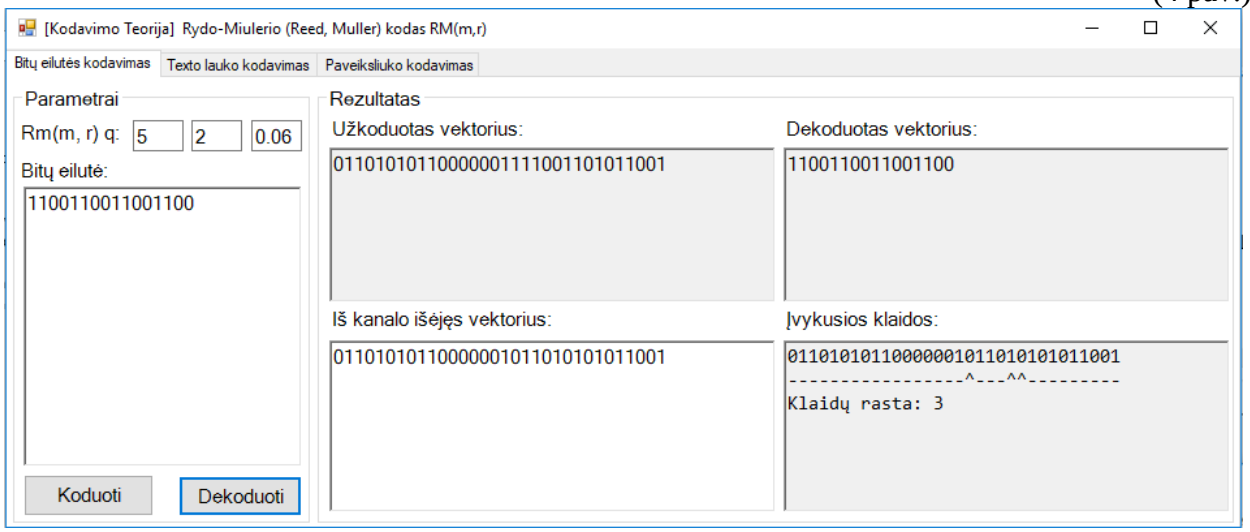
Bitų eilutės kodavimo pavyzdžiai:

Įvedus neteisingą bitų eilutės simbolių kiekį išmetama lentelė su nurodymu koks jis turėtų būti (3 pav.). Įvedus teisingus parametrus ir žinutę reikia paspausti mygtuką „Koduoti" ir tada gausime užkoduotą vektorių, iš kanalo išėjusį vektorių, įvykusių klaidų kiekių ir vietas. Tada galime pamodifikuoti iš kanalo išėjusį vektorių ir spausti mygtuką „Dekoduoti". Tuomet turėtų atsirasti dekoduotas vektorius.

(3 pav.)

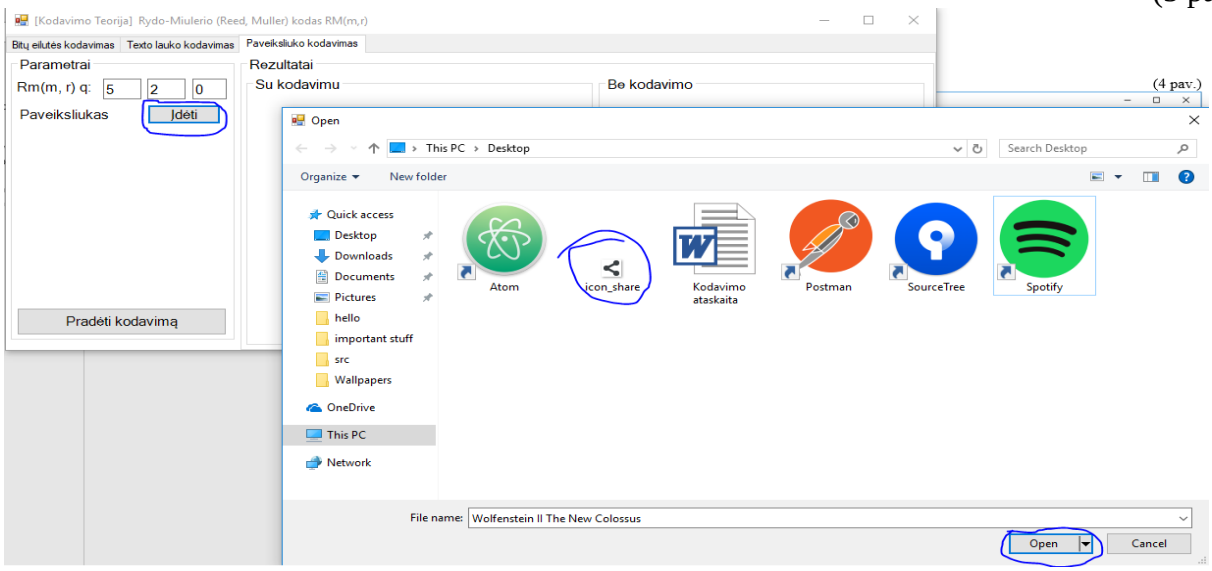


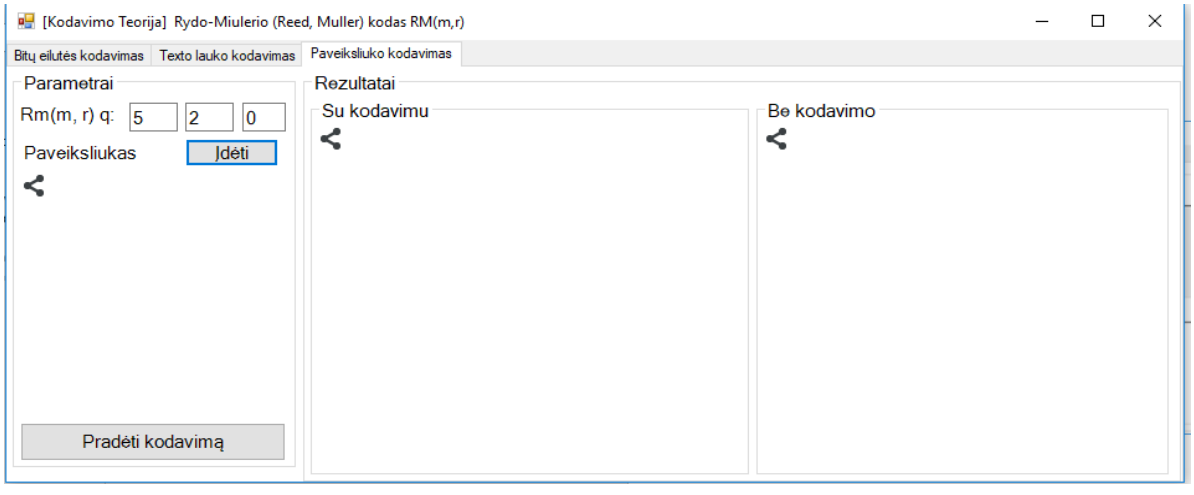
(4 pav.)



Paveikslliuko kodavimo pavyzdžiai:
Ši programos dali šiuo metu deramai neveikia. Eina tiktai įkelti paveikslliuką ir, nustačius parametrą q = 0, jį užkoduoti ir atkoduoti. Joje nėra saugiklių ir programą galima nesunkiai užlaušti.

(5 pav.)





Programavimo sprendimai

Kanalo sprendimai

Kanalas paima simbolių eilutę ir tikimybę joje įvelti klaidą. Iteruodamas per kiekvieną simbolių eilutės simbolį jis atlieka matematinį veiksmą patikrina ar gautas rezultatas yra mažesnis negu klaidos tikimybė. Jeigu taip, tuomet simbolis pakeičiamas į vienetą jeigu jis buvo nulis ir į nulį jeigu jis buvo vienetas.

```
static class Channel
{
    public static string Transmit(string message, double q)
    {
        char[] messageChars = message.ToCharArray();

        Random rn = new Random();

        for (int i=0; i<messageChars.Length; i++)
        {
            if((q * 100) > rn.Next(0, 100))
            {
                messageChars[i] = switchZeroOne(messageChars[i]);
            }
        }

        return new string(messageChars);
    }

    private static char switchZeroOne(char c)
    {
        return (c == '0') ? '1' : '0';
    }

}
```

Teksto pavertimo į bitų eilutę sprendimai

Metodas paima simbilių eilutę ir paverčia ją į bitų eilutę. Tada cikle skaldo ją ir gabaliukais deda į sąrašą tol kol jos nelieka. Tuomet kita programos dalis kiekvieną sąrašo įrašą siunčia per kanalą.

```
public List<string> divideTextFieldToBitList()                  // padalinu bitsringą į reikalingo dydžio gabaliukus
{

    string bitString = textToBitString(_TextField);
    List<string> bitStringListToSend = new List<string>();
    StringBuilder sb = new StringBuilder();

    int messageIndex = 0;
    while(bitString.Length >= _n)
    {
        for (int i = 0; i < _n; i++)
        {
            sb.Append(bitString[i]);

        }
        bitStringListToSend.Add(sb.ToString());
        bitString = bitString.Substring(_n);
        sb.Clear();
    }
    leftLength = bitString.Length;

    return bitStringListToSend;
}
```

Testavimas

Vektoriaus testavimas

1. m=5 r=4 q=0.1 bitų eil = 11111111110000000000011111111110
   Rezultatas: Klaidos neištaisytos
   Klaidų rasta: 4
   Užkoduotas vektorius: 00101000001010001111111011101001
   Iš kanalo išėjęs vektorius: 00101001101010001011110011101001
   Dekoduotas vektorius: 11111101110010001100110001111100

2. m=6 r=1 q=0.1 bitų eil. = 1110001
   Rezultatas: Ištaisė visas klaidas
   Klaidų rasta: 7
   Užkoduotas vektorius:
   0101010101010101101010101010101010101010101010101001010101010101010101
   Iš kanalo išėjęs vektorius:
   0101000101010101101010101011101011001110101010000101010101010111
   Dekoduotas vektorius: 1110001

3. m=5 r=2 q=0.1 bitų eil. = 1110001111111111
   Rezultatas: Ištaisė visas klaidas
   Užkoduotas vektorius: 11101000011111100111111000010111
   Iš kanalo išėjęs vektorius: 01111000011111100111111000010011
   Dekoduotas vektorius: 1110001111111111

Teksto lauko testavimas

1. m=5 r=2 q=0.01 teksto laukas = aaaaaaaa
   Rezultatas: ištaisė visas klaidas
   Be kodavimo: acaacaca
   Su kodavimu: aaaaaaaa

Vektoriaus eksperimentai

1. Atlikta bandymų: 7
   m=2 r=1 q=0.1 vektoriaus ilgis=3
   Dekoduojama teisingai tiktai kada kanale nebuvo klaidų. Jei kanale buvo nors viena klaida dekoduoja neteisingai. Su šiais parametrais kodas klaidų netaiso.
2. Atlikta bandymų: 7
   m=3 r=1 q=0.1 vektoriaus ilgis=4
   Dekoduojama teisingai visada jeigu buvo tik viena klaida arba klaidų nebuvo. Atsiradus daugiau klaidų visada dekoduoja neteisingai. Išvada – kodas su šiais parametrais taiso vieną klaidą.
3. Atlikta bandymų: 9
   m=4 r=1 q=0.1-0.15 vektoriaus ilgis=5
   Dekoduoja visada teisingai kai klaidų yra iki 3 imtinai. Esant daugiau klaidų visada dekoduoja klaidingai.
4. Atlikta bandymų: 9
   m=5 r=1 q=0.2 vektoriaus ilgis=6
   Dekoduoja visada teisingai kai klaidų yra iki 7 imtinai. Esant daugiau klaidų visada dekoduoja klaidingai.
5. Atlikta bandymų: 6
   m=5 r=2 q=0.1 vektoriaus ilgis=16
   Dekoduoja visada teisingai kai klaidų yra iki 3 imtinai. Esant daugiau klaidų visada dekoduoja klaidingai. Esant tokiai klaidų tikimybei tokie parametrai būtų pakankamai geri, nes iš 6 bandymų tik viename buvo daugiau klaidų negu 3.
6. Atlikta bandymų: 12
   m=6 r=1 q=0.2 vektoriaus ilgis=7
   Dekoduoja visada teisingai kai klaidų yra iki 16 imtinai. Esant daugiau klaidų visada dekoduoja klaidingai. Esant tokiai klaidų tikimybei tokie parametrai beveik garantuotai užtikrintų vektoriaus saugų gavimą. Iš 12 bandymų tik vienas buvo atkoduotas neteisingai.

Literatūra

- V. Stakėnas. Kodai ir šifrai. Vilnius, 2007 .
- https://vbktech.wordpress.com/2011/07/08/c-net-converting-a-string-of-bits-to-a-byte-array/
- https://social.msdn.microsoft.com/Forums/vstudio/en-US/a04472e3-c143-4626-bc1d-dc8f6f4c68c1/net-c-string-to-bit-array?forum=netfxbcl