

Kļūdu korekcija II

1. Heminga kods

Heminga koda vienkāršākais gadījums ir 7 bitu Heminga kods. Tas ir $[7, 4, 1]$ -kods. Tas ir, šis kods sastāv no $2^4=16$ 7 bitu virknēm. Tā kā kods spēj koriģēt 1 kļūdu, tad katras 2 no šīm virknēm atšķiras vismaz 3 vietās.

Pilns 16 virkņu saraksts atrodams iepriekšējās lekcijas konspektā. Šajā lekcijā mēs koncentrēsimies uz koda struktūru, t.i., kā aprakstīt un analizēt to, bez visu 16 virkņu uzskaitīšanas. 16 virknes, kas ietilpst Heminga kodā var aprakstīt šādi:

$$x_1 x_2 x_3 x_4 y_1 x_4 y_2 y_3,$$

kur $x_1 x_2 x_3 x_4$ var būt jebkuras vērtības no $\{0, 1\}$, bet y_1, y_2, y_3 tiek izrēķināti no x_1, x_2, x_3, x_4 šādā veidā:

$$y_1 = (x_1 + x_2 + x_3) \bmod 2,$$

$$y_2 = (x_1 + x_2 + x_4) \bmod 2,$$

$$y_3 = (x_1 + x_3 + x_4) \bmod 2.$$

Teorēma. Jebkuras 2 šādā veidā konstruētas virknes atšķiras vismaz 3 vietās.

Pierādījums. Apskatām 2 no šīm virknēm

$$x_1 x_2 x_3 y_1 x_4 y_2 y_3$$

un

$$x'_1 x'_2 x'_3 y'_1 x'_4 y'_2 y'_3.$$

Apskatām, cik no x_i ($i=1, 2, 3, 4$) atšķiras no atbilstošā x'_i .

1. gadījums. Atšķiras viens x_i .

Katrs x_i ietilpst vismaz 2 no formulām priekš y_1, y_2, y_3 un, mainoties x_i vērtībai, mainīsies šo formulu vērtības. Tāpēc virknes atšķiras vismaz 3 vietās: vienā x_i un vismaz divos y_i .

2. gadījums. Atšķiras divi x_i .

Apzīmējam atšķirīgos bitus ar x_i un x_j . Lai kādi būtu i un j , mēs vienmēr varam atrast vienu no y_i formulām, kurā ietilpst viens no x_i un x_j , bet ne otrs. Virknes atšķirsies vismaz 3 vietās: divos x_i un šajā vienā y_i .

3. gadījums. Atšķiras trīs vai vairāk x_i .

Tad mums uzreiz ir 3 atšķirības starp virknēm.

2. Piemēri kodēšanai un atkodēšanai, izmantojot Heminga kodu

1. piemērs. Nokodēt virkni 0110.

Risinājums. Ņemam $x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 0$. Tad, aprēķinot y_1, y_2, y_3 saskaņā ar dotajām formulām, iegūst $y_1 = 0, y_2 = 1, y_3 = 1$. Tātad kodētais ziņojums būs:

0110011.

2. piemērs. Atkodēt virkni 0111101.

Risinājums. Pārbaudām katru no y_1, y_2, y_3 saskaņā ar formulām. Konstatējam, ka y_1 vērtība nesakrīt ar vajadzīgo, jo pēc formulas sanāk

$$y_1 = (x_1 + x_2 + x_3) \bmod 2 = (0 + 1 + 1) \bmod 2 = 0,$$

bet saskaņā ar saņemto virkni y_1 būtu jābūt 1. Pārbaudot pārējos bitus, konstatējam, ka y_2 sakrīt, bet y_3 nesakrīt. Tālāko var vizualizēt šādā veidā:

	x_1	x_2	x_3	y_1	x_4	y_2	y_3
	X	X	X	X			
			X	X			X
	X		X		X		X

Tā kā y_1 nesakrīt, tad kļūdai jābūt vai nu y_1 vai no vienā no y_1 formulā ietilpstošajiem bitiem (x_1, x_2, x_3). Tie ir atzīmēti tabulas pirmajā rindā.

Tā kā y_2 sakrīt, tad kļūda var būt tikai tajos bitos, kas neietekmē y_2 (vai nu y_1 , vai y_3 , vai arī x_3 , kurš neietilpst formulā priekš y_2). Tie ir atzīmēti tabulas otrajā rindā.

Tā kā y_3 nesakrīt, tad kļūdai jābūt vai nu y_1 vai no vienā no y_1 formulā ietilpstošajiem bitiem. Tie ir atzīmēti tabulas trešajā rindā.

Vienīgais bits, kas ir atzīmēts visās rindās ir x_3 . Tātad mēs varam vienošķinīgi pateikt, ka tas ir kļūdainais bits. Tāpēc sākotnējais ziņojums bija 0101101.

3. piemērs. Atkodēt virkni 1010010.

Risinājums. Pārbaudot katru no y_1, y_2, y_3 saskaņā ar formulām, konstatējam, ka visas saņemtās vērtības sakrīt ar tām, kas aprēķinātas pēc formulas. Tātad kļūdu nav un šī ir pareiza Heminga koda virkne.

4. piemērs. Atkodēt virkni 1101110.

Risinājums. Pārbaudot katru no y_1, y_2, y_3 saskaņā ar formulām, konstatējam, ka nesakrīt tikai y_1 . Līdzīgi kā 2. piemērā izveidojam tabulu:

	x_1	x_2	x_3	y_1	x_4	y_2	y_3
	X	X	X	X			
			X	X			X
		X		X		X	

Konstatējam, ka kļūda var būt tikai bitā y_1 . Tāpēc nosūtītais ziņojums bija 1100110.

3. Heminga kods: vispārējais gadījums.

Vispārējā gadījumā Heminga kods sastāv no $2^n - 1$ bitu virknēm, kurās $2^n - n - 1$ biti tiek izmantoti ziņojumam, bet n ir kontrolbiti, kas tiek izrēķināti no ziņojuma bitiem.

Lai aprakstītu šo kodu, sanumurējam $2^n - 1$ bitu pozīcijas ar skaitļiem 1, 2, ..., $2^n - 1$, šos skaitļus pierakstot binārajā skaitīšanas sistēmā (000001, 000010, ..., 111111). Ir n

skaitļi, kuru binārajā pierakstā ir tieši viens 1 (000001, 000010, ..., 100000). Šajās pozīcijās būs kontrolbiti.

Pārējās pozīcijās ir ziņojuma biti, kas var būt patvaļīgi. Kontrolbiti tiek izrēķināti šādi:

$$x_{0...010...0} = \sum_{i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_n} x_{i_1 \dots i_{k-1} 1 i_{k+1} \dots i_n} \pmod{2}.$$

Lai atrastu, vai ir kļūda, rīkojās šādi. Ja kontrolbits pozīcijā 0...010...0 (ar 1nieku k-tajā ciparā) nesakrīt ar to, kas izrēķināts pēc formulas, tad mēs zinām, ka kādā no bitiem, kuru numuriem ir 1 k-tajā pozīcijā ir kļūda. Ja kontrolbits pozīcijā 0...010...0 (ar 1nieku k-tajā ciparā) sakrīt ar to, kas izrēķināts pēc formulas, tad kļūda var būt tikai tajos bitos, kuru numuriem k-tajā pozīcijā ir 0 (jo visi biti ar 1 k-tajā pozīcijā ietilpst formulā).

Šādā veidā pēc katra kontrolbita var noteikt vienu bitu priekš pozīcijas, kurā ir kļūda, numura. Kontrolbiti kopumā tad pilnībā nosaka šīs pozīcijas numuru. Ja iegūtais numurs ir 000...000 (t.i., visi kontrolbiti sakrita), tad kļūdas nav vispār. Citādi, mēs zinām, kurā vietā tā ir.

4. Heminga koda optimalitāte

Teorēma. Ja $S \subseteq \{0, 1\}^{2^n-1}$ ir kods, kas spēj koriģēt vienu kļūdu, tad

$$|S| \leq 2^{2^n-n-1}.$$

Šī teorēma nozīmē, ka virkņu skaitu Heminga kodā nevar uzlabot pat par 1 virkni!

Pierādījums. Apzīmējam koda virknes ar v_1, \dots, v_m . Ar V_i apzīmējam kopu, kur ietilpst v_i un visas virknes, kas atšķiras no v_i tieši vienā vietā. Tad:

1. Nevienai V_i nav kopīgu elementu ar kādu citu V_j (jo citādi, ja mēs saņemtu virkni, kas pieder gan V_i , gan V_j , mēs nespētu veikt kļūdu korekciju).
2. Katrā V_i ietilpst tieši 2^n virknes: v_i un 2^n-1 virknes, kas atšķiras no tās 1 pozīcijā.

Tāpēc kopās V_1, \dots, V_m kopā ir $2^n m$ elementi. Tā ka vispār ir 2^{2^n-n-1} virkņu garumā 2^n-1 , tad

$$2^n m \leq 2^{2^n-n-1},$$

$$m \leq 2^{2^n-n-1}.$$

5. Lineāri kodi

Jebkuru kodu, kurā katrs nokodētās virknes bits ir aprakstāms ar formulu

$$(x_1 + \dots + x_{ik}) \pmod{2}$$

sauc par lineāru kodu. Heminga kods ir lineārs kods un gandrīz visi citi praksē lietotie kodi arī ir lineāri.

Lineāru kodu var aprakstīt ar tā ģeneratormatricu. Ja n ir kodētā ziņojuma garums, bet k – nokodēto bitu skaits, tad ģeneratormatrica ir $n \times k$ matrica. Ja bits x_i ietilpst formulā

pēc kuras rēķina j -to nokodētā ziņojuma bitu, tad šīs matricas (i, j) -ajā vietā ir 1. Citādi tur ir 0. Piemēram, Heminga koda ģeneratormatrica izskatās šādi:

1	0	0	0
0	1	0	0
0	0	1	0
1	1	1	0
0	0	0	1
1	1	0	1
1	0	1	1

1., 2., 3. un 5. rinda apraksta, ka atbilstošie kodētā ziņojuma biti sakrīt ar sākotnējā ziņojuma bitiem. Pārējās rindas apraksta formulas priekš kontrolbitiem.

Lai nokodētu ziņojumu, mēs aprakstām to ar vektoru:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

un tad reizinām šo vektoru ar ģeneratormatricu M . Nokodētais ziņojums būs Mx , visus tā elementus rēķinot pēc moduļa 2.

Priekš atkodēšanas tiek izmantota paritātes pārbaudes matrica, kas Heminga kodam ir šāda:

1	1	1	1	0	0	0
1	1	0	0	1	1	0
1	0	1	0	1	0	1

Katra tabulas rinda apraksta vienu no Heminga koda pārbaudēm (vai kontrolbits sakrīt ar noteiktu bitu summu pēc mod 2). Ja x – nokodētais ziņojums, P – paritātes pārbaudes matrica un k lūdu nav, tad, rēķinot pēc mod 2, jāizpildās

$$Px = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Paritātes pārbaudes matricu var izmantot arī, lai noteiktu, kur ir kļūdas, ja tādas ir, bet tas ir sarežģītāk un šajā kursā netiks aplūkots.