Eötvös Loránd Tudományegyetem

Természettudományi Kar

Bedő József Botond

Hibajavító kódolás lineáris blokk-kódokkal

Szakdolgozat

Matematika BSc, matematikai elemző szakirány

Témavezető:
Fialowski Aliz
Algebra és Számelmélet Tanszék

Budapest, 2019

Köszönetnyilvánítás

Először is szeretnék köszönetet mondani témavezetőmnek, Fialowski Aliznak, a téma elvállalásáért, az írás alatt kitartó türelméért és hogy lelkesedésével motivált a szakdolgozat elkészítésében. Továbbá szeretném megköszönni Kiss Emilnek, Frenkel Péternek és Szőnyi Tamásnak, hogy egyetemi éveim során hozzáértésükkel és segítőkészségükkel megismertették és megszerettették velem az algebra és a kódelmélet területét, valamint Kimle Máriának, hogy gimnázium alatt segített kiválasztani ezt a pályát. Végül szeretnék köszönetet mondani családomnak, akik mindvégig támogattak a tanulmányaim során.

Tartalomjegyzék

Bevezetés 5

1. Információelméleti alapfogalmak **6**

1.1. Információ, mint fizikai mennyiség 6

1.2. Az üzenet útja 9

1.3. Csatornakódolás 12

1.3.1. Hamming távolság és a hiba észlelése/javítása 12

1.3.2. Minimális távolság 14

1.3.3. Legközelebbi szomszédok 16

2. Algebrai struktúrák **18**

2.1. Csoportok 18

2.1.1. Részcsoportok 19

2.1.2. Ciklikus csoportok 19

2.1.3. Mellékosztályok 20

2.2. Testek 22

2.2.1. A $\mathbb{Z}_p$ test 22

2.3. Gyűrűk 24

2.3.1. Polinomgyűrűk 25

2.4. Vektorterek 27

3. Lineáris kódok, blokk-kódok **29**

3.1. Korlátok 29

3.2. Lineáris kódok bevezetése 31

3.3. Hibajavítás szindrómákkal 34

3.4. Hamming-kódok 37

4. Ciklikus kódok **44**
4.1. Bináris ciklikus kódok 44
4.1.1. Generátor- és paritáspolinomok 45
4.1.2. Szindróma 48
4.2. Műveletek elvégzése áramkörök segítségével 48
4.2.1. Szisztematikus kódolás 48
4.2.2. Dekódolás szindrómával 50

A. A 4.2 **alfejezetben lévő példa számítógépes kiszámítása** **57**

Bevezetés

Digitális adatátvitel esetén kódolnunk kell az üzenetet, hogy eljuthasson a feladótól a címzetthez, a csatornán keresztül. Ilyenkor mit értünk a kódolás szó alatt? Milyen feladatot lát el kódolás? Csak egy feladatra alkalmas, esetleg több fajtája létezik, különböző problémákra? Mennyire tudja jól ellátni a feladatát, milyen korlátok közé van szorítva? Milyen matematikai elméleten alapul? Milyen alkalmazásai vannak? Szakdolgozatomban ezekre a kérdésekre próbálok választ adni.

Először is fontos megkülönböztetni a kódolásokat feladatuk alapján. A kriptográfia feladata az üzenet titkosítása, a forráskódolás az üzenet tömörítésére használható, a hibajavító kódolás pedig, amiről ez a szakdolgozat szól, a zajos csatornán áthaladó adatok rekonstruálásával foglalkozik.

Az első fejezet kicsit részletesebben foglalja össze az üzenet fentebb említett útját, és a különféle kódolási eljárások alkalmazását. Továbbá egy minimális valószínűségszámítási és információelméleti bevezetéssel egy egyszerű modellt vezet be a csatorna viselkedésére, aminek komolyabb vizsgálata egy egész szakdolgozatot megtölthetne. A második fejezet az elméleti hátteret gyűjti össze az algebra területéről. A későbbiekben bevezetésre kerülő kódok viselkedése a véges testeken és vektortereken alapszik, ezért ezek kapják a fő hangsúlyt, a csoportelméleti alapokról építkezve. A harmadik fejezet a hibajavító kódok egy konkrét fajtájával, a lineáris blokk-kódokkal foglalkozik. Ezek a kódok a második fejezetben ismertetett algebrai fogalmakon alapulnak. Könnyű velük számolni, a tudományág kialakulásának kezdete óta ismertek, és máig használhatók, amik miatt érdemes velük foglalkozni. Ez a fejezet a kódok „jóságával” is foglalkozik, néhány fontos korlát bevezetésével. A negyedik fejezet egy még konkrétabb kódcsaláddal, a ciklikus kódokkal foglalkozik, ami egy speciális esete a lineáris blokk-kódoknak. Ebben a fejezetben bevezetésre kerül a számítógépek használata a kódolás folyamatában, amire egy konkrét példát is adok a függelékben.

A hibajavító kódolást felhasználják többek közt a műholdas televízióadások sugárzásakor, telefonos beszélgetések alkalmával, űrkutatásban az üzenet hosszú útja alatt az információ megóvása érdekében, és fizikai adathordozók, például CD-k lejátszásakor is. Már ez a néhány példa is mutatja, hogy a hibajavító kódolás, vagy algebrai kódelmélet egy releváns alkalmazása a matematikának.

1. fejezet

Információelméleti alapfogalmak

Ebben a fejezetben először alapvető információelméleti definíciókat ismertetek, majd ezek segítségével végighaladok az információ útján egy általános ábrán, bemutatva az egyes lépéseit, és a szakdolgozatomat elhelyezve ebben a folyamatban.

Forrásként a [1], a [2], a [3] és a [11] eredményeire támaszkodtam.

1.1. Információ, mint fizikai mennyiség

Az 1940-es években Claud E. Shannon bevezette az információ fogalmát, mint az üzenet számszerűsíthető tulajdonságát, egy fizikai mennyiséget, ami mérhető, átalakítható, tárolható. Ebben a megközelítésben az információt pusztán szintaktikai szempontból (struktúra) vizsgáljuk, nem pedig szemantikaiból (jelentés). Definiáljuk az információt, és más hozzá kapcsolódó fontos fogalmakat.

1.1.1. Definíció. Egy E_i eseményhez kapcsolódó *információ* a következő:

$$Info(i) = log_2\left(\frac{1}{p_i}\right) \tag{1.1}$$

ahol p_i az E_i bekövetkeztének valószínűsége.

1.1.2. Következmény. Minél kisebb egy adott esemény bekövetkezésének valószínűsége, annál nagyobb információt hordoz magában. Ha egy esemény bekövetkezése biztos (E_i esetén $p_i = 1$) akkor E_i nem hordoz információt ($log_2 1 = 0$).

1.1.3. Következmény. Ha két esemény független, az információ zárt az összeadásra, azaz $Info(1,2) = Info(1) + Info(2)$. Mi ezt az esetet feltételezzük, az egyszerűbb számolások érdekében.

1.1.4. Definíció. *Információs forrásnak* nevezzük egy forrásábécé és egy valószínűségi eloszlás kettősét, például az $\{a_1, \ldots, a_n\}$ halmazt és a $P(a_1), \ldots, P(a_n)$ számokat, amikre teljesül, hogy $\sum_{i=1}^{n} P(a_i) = 1$ és $0 \leq P(a_i) \leq 1$.

Jelölése: S

Ezt, a későbbi számolásokra való tekintettel diszkrét, nulla-memóriás információs forrásnak nevezhetjük. Diszkrét, mert az ábécé halmaza véges sok elemből áll, és nulla-memóriás, mert feltételezzük, hogy a szimbólumok egymástól függetlenül veszik fel értéküket.

Az egyes események bekövetkeztének információtartalmát már definiáltuk, most vizsgáljuk azt, ha ezek az események a diszkrét, nulla-memóriás információs forrásnak a kimenetelei. Ekkor egy információs forrás alapvető tulajdonsága a szimbólumonkéni átlagos információtartalom, azaz az entrópia.

1.1.5. Definíció. Egy $p_1, \ldots, p_n$ valószínűséggel előforduló szimbólumokkal rendelkező S információs forrás *entrópiáján* a következő számot értjük:

$$H(S) = \sum_{i=1}^{n} p_i \cdot log_2\left(\frac{1}{p_i}\right) = -\sum_{i=1}^{n} p_i \cdot log_2(p_i) \tag{1.2}$$

1.1.6. Megjegyzés. A logaritmus alapja tetszőlegesen választható, kettes alap esetén az információ mértékegységét *bitnek* nevezzük.

1.1.7. Megjegyzés. A bit, amit információ mérésére használunk, különbözik attól a bittől, amivel a számítógépes tárhelyt jellemezzük.

Most pedig lássunk be egy tételt, aminek az eredményét hibajavító kódolások előtt mindig felhasználjuk.

1.1.8. Tétel. *I. A minimális entrópia a $H(S) = 0$ bit, és ez pontosan azoknál az S forrásoknál teljesül, amelyek tartalmaznak olyan szimbólumot, melynek bekövetkezési valószínűsége* 1.

II. A maximális entrópia egy n szimbólumból álló S forrás esetében $H(S) = log_2(n)$ bit, és ez pontosan azoknál az S forrásoknál teljesül, amelyeknek minden szimbólumuk ugyanakkora valószínűséggel fordul elő, azaz $p_i = \frac{1}{n}$ minden i esetén.

Bizonyítás. I. Az (1.2) egyenlet szerint $H(S) = \sum_{i=1}^{n} p_i \cdot log_2\left(\frac{1}{p_i}\right)$, ahol p_i és $log_2\left(\frac{1}{p_i}\right)$ is nemnegatívak, ugyanis 1.1.4 definíció alapján $0 \leq p_i \leq 1$, vagyis $1 \leq \frac{1}{p_i}$. Tehát $H(S) \geq 0$,

és $H(S) = 0$ akkor, ha minden i esetén $log_2\left(\frac{1}{p_i}\right) = 0$ (ami $p_i = 1$ esetén történik meg), vagy ha $p_i = 0$. Ez a 1.1.4 definíció alapján csak akkor történhet meg, ha valamelyik szimbólum előfordulásának valószínűsége 1, az összes többié pedig 0.

II. Ha minden szimbólum ugyanakkora valószínűséggel következik be, azaz minden i esetén $p_i = \frac{1}{n}$, akkor

$$H(S) = \sum_{i=1}^{n} \frac{1}{n} \cdot log_2 n = log_2(n) \tag{1.3}$$

bit. Hogy beláthassuk, hogy ez a maximális érték, felhasználjuk a következő egyenlőtlenséget:

$$ln(x) \leq x - 1 \text{ ahol egyenlőség akkor és csak akkor teljesül, ha } x = 1 \tag{1.4}$$

Kiszámoljuk egy tetszőleges S forrás entrópiájának, és $log_2(n)$-nek a különbségét:

$$\begin{aligned}
H(S) - log_2(n) &= \sum_{i=1}^{n} p_i \cdot log_2\left(\frac{1}{p_i}\right) - \sum_{i=1}^{n} p_i \cdot log_2(n) \\
&= \frac{1}{ln(2)} \cdot \sum_{i=1}^{n} p_i \cdot \left(ln\left(\frac{1}{p_i}\right) - ln(n)\right) \\
&= \frac{1}{ln(2)} \cdot \sum_{i=1}^{n} p_i \cdot ln\left(\frac{1}{p_i \cdot n}\right) \\
&\leq \frac{1}{ln(2)} \cdot \sum_{i=1}^{n} p_i \cdot \left(\frac{1}{p_i \cdot n} - 1\right) \\
&= \frac{1}{ln(2)} \cdot \left(\sum_{i=1}^{n} \frac{1}{n} - \sum_{i=1}^{n} p_i\right) \\
&= \frac{1}{ln(2)} \cdot (1 - 1) = 0.
\end{aligned}$$

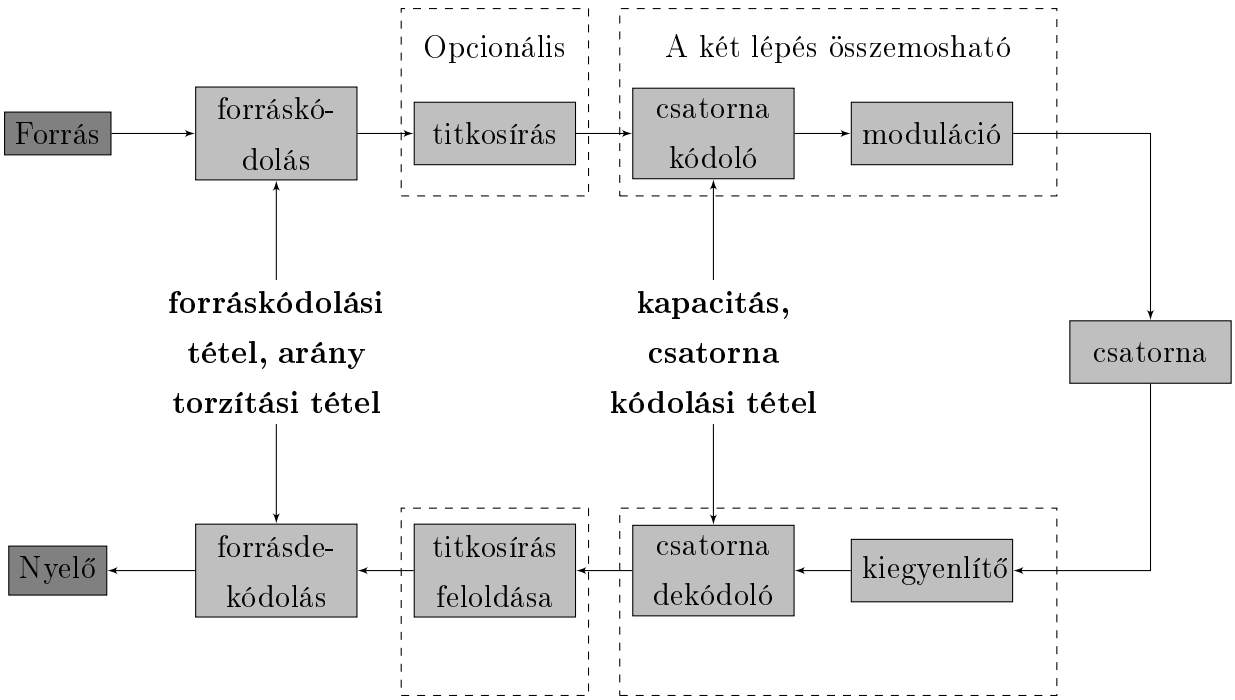
Tehát $H(S) \leq log_2(n)$, és akkor teljesül az egyenlőség, ha $\frac{1}{p_i \cdot n} = 1$, azaz $p_i = \frac{1}{n}$ mindn i esetén. □

Mivel a maximális entrópia, azaz a maximális információ $log_2(n)$, ez azt jelenti, hogy ha kevesebb információt tartalmaz a forrás, akkor az kevesebb szimbólum segítségével is ábrázolható.

1.1.9. Definíció. A forrás *redundanciáját* a következő képlet határozza meg:

$$\text{Redundancia} = log_2(n) - H(S)$$

Ebből a definícióból következik, hogy nem érdemes minden forrást $log_2(n)$ bit segítségével ábrázolni, annál kevesebb is elég lehet. Ezzel eljutottunk a tömörítéshez, mint egy



1.1. ábra. Az üzenet útja

fontos kódolási folyamathoz, ami megelőzi a hibajavító kódolást, és előkészíti az üzenetet. Ebben a szakdolgozatban ezzel a módszerrel nem fogok részletesebben foglalkozni, az eddigi bemutatásának az volt a szerepe, hogy láthassuk mi az az üzenet, honnan indul, és hogyan jut el abba az állapotba, amikor a hibajavító kódolás elkezdődhet. Az üzenet egy általános folyamatával a következő szakaszban foglalkozom még, egy teljesebb kép kedvéért.

1.2. Az üzenet útja

Azért, hogy az üzenet eljuthasson a feladótól a címzetthez, szükség van a *csatornára*, ami az a konkrét fizikai eszköz, amelynek segítségével a kommunikált jelzések és üzenetek eljutnak a címzetthez. Mielőtt ezt megtenné, több lépésben, többféle kódolási eljáráson esik át. Fontos kiemelni, hogy ezek a lépések mind más funkcióval rendelkeznek, és másfajta problémák elkerülésére teszik alkalmassá az üzenetet. Végül, mielőtt az üzenet eljutna a címzetthez, átesik a megfelelő dekódolási eljárásokon.

Az 1.1. ábra ezt a folyamatot szemlélteti.

Az egyes lépések bizonyos miértjei megválaszolása érdekében kimondásra kerül néhány tétel, ám ezeket nem bizonyítom, mert nem kapcsolódnak szorosan a szakdolgozatom témájához.

Forrás

Az adat, amit kommunikálni szeretnénk, digitális formában reprezentálva. Információelméleti szempontból véletlen számok folyamjaként kezeljük, amit egy valószínűségi eloszlás határoz meg. Minden forrás mérhető információmennyiséggel rendelkezik.

Forráskódolás

Adattömörítést végez a redundancia eltüntetésével. A forrás információjának tárolására használt bitek száma meghaladhatja a valódi információ tárolásához szükséges bitek számát, amit az entrópia segítségével számolhatunk ki. A tömörítés elvégezhető információveszteség nélkül, ezt a forráskódolási tétel biztosítja. A 1.1.8 tételben beláttuk, hogy egy bit akkor rendelkezik a legnagyobb információtartalommal, ha a különböző kimenetelek egyforma valószínűséggel következnek be. Ez azért lesz fontos a későbbiekben, mert ez alapján általában feltehető, hogy a hibajavító kódolás közben a különböző szimbólumok ugyanakkora valószínűséggel fordulnak elő. Egy példa a veszteség nélküli bináris forráskódolásra a Huffman kód.

Titkosírás

Elrejti, vagy összekeveri az információt, hogy jogosultság nélküli személyek ne tudják kinyerni azt az üzenetből. Magához az üzenet sikeres átjuttatásához a forrásból a nyelőbe nincs köze, azaz ebből a szempontból ez a lépés elhagyható. Ez a fajta kódolás alapjaiban különbözik a hibajavító kódolástól, ezért a szakdolgozatomban nem foglalkozom vele részletesebben.

Csatorna kódoló

Redundáns információt ad az üzenethez, hogy annak segítségével a csatornában bekövetkező hibákat ki tudja javítani. Ennek érdekében az itt hozzáadott redundancia (a forrásban lévővel ellentétben) erre a célra alkalmas struktúrával rendelkezik. A hozzávett redundáns bitek miatt az output hosszabb, mint az input. Általában k hosszú szimbólumokat

fogad be, és n hosszú szimbólumokat ad ki. Ennek alapján definiálható a kód aránya: $R = k/n$. A hibajavító kódolás ebben a lépésben, és ennek a dekódolásában zajlik, ezért a szakdolgozatom is erre a két lépésre fókuszál.

Moduláció

A szimbólum szekvenciákat jellé alakítja, ami alkalmas a csatornán való áthaladásra. A csatornától függően különféle feltételeknek kellhet megfelelnie ezeknek a jeleknek (pl.: meghatározott frekvencia). Ez a lépés elvégezhető egyszerre a csatorna kódolással, ekkor a lépés neve: *kódolt* moduláció.

Csatorna

Az a közvetítő, amin az információ áthalad. Egyszerű példák a telefonvonalak, az internetkábelek, mobiltelefon csatornák, de ide tartoznak a DVD-k, CD-k is, ahol nem azonnali az információ átvitele. Amíg a jel áthalad ezeken a közvetítő közegeken, sérülhet, például eltolódhat, távolság miatt csillapodhat, bizonyos részei elveszhetnek, más jellel interferálhat. DVD-k esetén probléma lehet a karcolódás, és az ujjlenyomatok. Ezeket a problémákat együttesen *zajnak* nevezzük. A zaj problémájának megoldása a hibajavító kódolás feladata. A lehetséges hibák bekövetkeztének felmérésére, a csatornák tanulmányozására különféle matematikai modellek készültek, amik megpróbálják jól ábrázolni az egyes csatornák sajátosságait, de kellően absztraktak, hogy egyszerűen lehessen velük számolni. Néhány egyszerűbb modell például a BSC (binary symmetric channel) csatorna, az AWGN (additive white Gaussian noise) csatorna. Mivel a bonyolultabb modellek gyakran ezeken alapulnak, az egységesség és egyszerűség kedvéért ebben a szakdolgozatban a *BSC modellt* fogom használni. A csatornáknak különböző információ átvivő képességük van. Azt a mennyiséget, amennyi információt megbízhatóan át tud vinni egy csatorna, *kapacitásnak* nevezzük, és C-vel jelöljük. A kapacitás a Shannon csatorna kódolási tétele alapján szorosan kapcsolódik a hibajavító kódolás használatához. A tétel szerint, ha az átvitel R aránya kisebb, mint a C kapacitás, akkor létezik olyan kód, aminek a hibázási valószínűsége tetszőlegesen kicsire állítható.

Kiegyenlítő

Fogadja a jelet a csatornától, és szimbólumok szekvenciájává alakítja azokat.

Csatorna dekódoló

A csatorna kódoló által az üzenethez illesztett redundáns információ segítségével kijavítja a csatornában keletkezett hibákat.

Titkosírás feloldása

Eltávolítja az üzenet titkosítását.

Forrásdekódolás

Kicsomagolja az üzenetet.

Nyelő

Az üzenet végső célja.

1.3. Csatornakódolás

A kódszó megváltozik bizonyos helyeken a csatornában. Onnan tudhatjuk, hogy hiba történt, hogy nem egy kódszót kapunk vissza, hanem egy jelentés nélküli szót. Ebben az esetben hibát észlelünk, majd ezt úgy próbáljuk kijavítani, hogy megkeressük a szimbólumsorozathoz legközelebbi kódszót. Ez az alfejezet ezt a folyamatot részletezi, definiálva alapvető fogalmakat, amiket végül egy példán keresztül mutatok be. Ehhez az alfejezethez már a [5] és [7] forrásokat használtam.

1.3.1. Hamming távolság és a hiba észlelése/javítása

1.3.1. Definíció. Legyen $q \in \mathbb{N}$. Ekkor Q-t *q elemű ábécének* nevezzük, ami q különböző elem halmaza. Egy Q ábécén értelmezett n hosszú szó az $(x_1, x_2, \ldots, x_n)$ szekvencia, ahol $x_i \in Q$ minden i esetén.

1.3.2. Definíció. Legyen Q egy ábécé, és legyen $n \in N$. Egy Q-n értelmezett *n hosszú kód* a Q^n egy legalább kételemű részhalmaza, jelölése: C. A C elemeit *kódszavaknak* hívjuk. A C halmaz elemszámát $|C|$ jelöli.

1.3.3. Megjegyzés. Egy adott kódban minden kódszó ugyanolyan hosszú.

1.3.4. Megjegyzés. Feltesszük, hogy minden kód legalább kettő kódszót tartalmaz, különben nem lenne értelme a hibajavításnak.

1.3.5. Definíció. Biteken értelmezett bináris ábécé: $\{0, 1\}$. Bináris kód a $\{0, 1\}$-en értelmezett kód.

Bár elméleti szempontból a kódolás minden elemszámú ábécé esetén ugyanúgy működik, az egyszerűség és egységesség kedvéért a harmadik fejezetben a legtöbb definíció, és számolás bináris ábécé mellett kerül kimondásra és megoldásra. A bináris ábécé választásának további előnye, hogy ilyen módon dolgozzák fel az üzeneteket a számítógépek is.

1.3.6. Definíció. Jelölje Q az ábécét, és legyen $\mathbf{u}, \mathbf{v} \in Q^n$ két n hosszú szó. A *Hamming távolság* $\mathbf{u}$ és $\mathbf{v}$ közt azoknak a pozícióknak a száma, ahol $\mathbf{u}$ és $\mathbf{v}$ különböznek. Jelölése: $d(\mathbf{u}, \mathbf{v}) = |\{i \in \{1, 2, \ldots, n\} : u_i \neq v_i\}|$.

A Hamming távolságot röviden távolságnak nevezzük, ez az a fogalom, amire a bevezetésben utaltam.

1.3.7. Példa. Négy hosszú bináris szavakat használva, azt kapjuk, hogy $d(0011, 1101) = 3$. Ha az $\{A, B, C\}$ elemekkel dolgozunk, akkor $d(ACCB, ABBC) = 3$.

1.3.8. Tétel. *Legyen Q egy q-betűs ábécé, legyenek $\mathbf{u}, \mathbf{v}, \mathbf{w}$ szavak Q^n-ből. Ekkor:*

1. $d(\mathbf{u}, \mathbf{v}) \geq 0$
2. $d(\mathbf{u}, \mathbf{v}) = 0 \Longleftrightarrow \mathbf{u} = \mathbf{v}$
3. $d(\mathbf{u}, \mathbf{v}) = d(\mathbf{v}, \mathbf{u})$
4. $d(\mathbf{u}, \mathbf{w}) \leq d(\mathbf{u}, \mathbf{v}) + d(\mathbf{v}, \mathbf{w})$

Bizonyítás. Az 1. – 3. rögtön következik a 1.3.6 definícióból, emiatt csak a 4. állítást látom be.

Jelölje $d(\mathbf{u}, \mathbf{v}) = a$ és $d(\mathbf{v}, \mathbf{w}) = b$. Ekkor az $i_1 \ldots i_a$ indexeken különbözik $\mathbf{u}$ a $\mathbf{v}$-től, és a $j_1 \ldots j_b$ indexeken különbözik $\mathbf{v}$ a $\mathbf{w}$-től. Továbbá $u_i = v_i$ pontosan akkor, ha $i \neq i_1, \ldots, i_a$, és $v_i = w_i$ pontosan akkor, ha $i \neq j_1, \ldots, j_b$. Ebből következik, hogy $u_i = w_i$ pontosan akkor, ha $i \neq i_1, \ldots, i_a, j_1, \ldots, j_b$. Tehát azoknak az indexeknek a száma, ahol $\mathbf{u}$ különbözik $\mathbf{w}$-től, legfeljebb $a+b$. Ezzel beláttuk, hogy $d(\mathbf{u}, \mathbf{w}) \leq a+b = d(\mathbf{u}, \mathbf{v}) + d(\mathbf{v}, \mathbf{w})$ $\square$

1.3.9. Definíció. Legyen C egy n hosszú kód Q ábécé felett, és legyen $t \in \mathbb{N}$. Azt mondjuk, hogy C:

- *t-hibajelző*, ha bármely $\mathbf{u} \in C$ kódszóra, és bármely n hosszú, Q-n értelmezett $\mathbf{v}$ szóra, melyre $\mathbf{v} \neq \mathbf{u}$, és $d(\mathbf{u}, \mathbf{v}) \leq t$, akkor $\mathbf{v} \notin C$.
- *t-hibajavító*, ha $\mathbf{u} \in C$ tetszőleges kódszóra, és bármely n hosszú, Q-n értelmezett $\mathbf{v}$ szóra, melyre $d(\mathbf{u}, \mathbf{v}) \leq t$, akkor $d(\mathbf{u}, \mathbf{v}) < d(\mathbf{u}', \mathbf{v})$ minden olyan $\mathbf{u}' \in C$ kódszóra, amelyre $\mathbf{u} \neq \mathbf{u}'$.

1.3.10. Megjegyzés. Az előző definíció szemléletesebb megfogalmazása:

- Egy kód t-hibajelző, ha bármely kódszóban bármely legfeljebb t helyet megváltoztatva, nem kaphatunk másik kódszót.
- Egy kód t-hibajavító, ha egy $\mathbf{v}$ szó benne van $\mathbf{u}$ kódszó t-távolságú környezetében, akkor semelyik másik kódszó t-távolságú környezetében nincs benne.

1.3.2. Minimális távolság

1.3.11. Definíció. Legyen C kód. Ekkor C *minimális távolságát* d_{min}-nel jelöljük, és a következőképp definiáljuk: $d_{min} = min\{d(\mathbf{u}, \mathbf{w}) : \mathbf{u}, \mathbf{w} \in C, \mathbf{u} \neq \mathbf{w}\}$.

1.3.12. Következmény. A d_{min} kiszámításakor C minden kódszavát összehasonlítjuk az összes többi kódszóval pontosan egyszer, ez $\frac{|C|(|C|-1)}{2}$ összehasonlítást jelent.

1.3.13. Példa. Legyen C egy bináris, 3 hosszú kód, a következő kódszavakkal: $001, 010, 100, 111$. Ekkor $d(\mathbf{u}, \mathbf{v}) = 2$ minden kódszó esetén, azaz $d_{min} = 2$. Ha a kódszavakhoz hozzávennénk a 000-t is, akkor $d_{min} = 1$- re változna, ugyanis $d(000, 001) = 1$.

1.3.14. Tétel. *Legyen a C kód minimális távolsága d_{min} és legyen $t \in \mathbb{N}$. Ekkor a C kód:*

1. *t-hibajelző* $\iff t < d_{min}$
2. *t-hibajavító* $\iff 2t < d_{min}$

Bizonyítás. Először 1-et bizonyítjuk. Ha $d_{min} > t$, akkor C legfeljebb t hibát jelez. Legyen $\mathbf{u}$ kódszó, és $\mathbf{u}'$ egy olyan szó, amire teljesül, hogy $1 \leq d(\mathbf{u}, \mathbf{u}') \leq t$. Ekkor $\mathbf{u}'$ nem lehet kódszó, mert különben $d_{min} \leq d(\mathbf{u}, \mathbf{u}') \leq t$.

Ezzel ellentétben, ha $d_{min} \leq t$, akkor C nem jelezhet t hibát. Legyenek $\mathbf{u}$ és $\mathbf{u}'$ kódszavak úgy, hogy $d(\mathbf{u}, \mathbf{u}') = d_{min}$. Ekkor $d(\mathbf{u}, \mathbf{u}') \leq t$, és a hiba, ami $\mathbf{u}$-t $\mathbf{u}'$-vé alakította, észrevétlen marad.

Most lássuk be a 2. állítást. Legyen $d_{min} \leq 2t$. Megmutatjuk, hogy C nem tud t tetszőleges helyen bekövetkezett hibát kijavítani. Ehhez legyen $\mathbf{u}, \mathbf{v} \in C$ két kódszó, amire $d(\mathbf{u}, \mathbf{u}') = d_{min}$. Legyenek $i_1, i_2, \ldots, i_r$ azok az indexek, amikben $\mathbf{u}$ különbözik $\mathbf{v}$-től. Ekkor $r \leq 2t$. Tegyük fel, hogy $\mathbf{u}$ üzenetet küldtük el, és a zaj megváltoztatta az u_i szimbólumokat, ahol $i = i_2, i_4, i_6, \ldots$ (vagyis a páros indexekben) a $\mathbf{v}$ értékeire. Így kapjuk a $\mathbf{u}'$ szót, ahol

$$u_i' = \begin{cases} u_i = v_i & : i \neq i_1, i_2, \ldots, i_r, \\ u_i & : i = i_1, i_3, \ldots, \\ v_i & : i = i_2, i_4, \ldots \end{cases}$$

Ekkor nyilvánvaló, hogy $d(\mathbf{u}, \mathbf{u}') \leq \frac{r}{2} \leq t$, viszont $d(\mathbf{u}', \mathbf{v}) \leq d(\mathbf{u}', \mathbf{u})$. Emiatt lehet, hogy $\mathbf{u}'$-t hibásan $\mathbf{v}$-nek dekódoljuk.

Ezzel ellentétben, legyen $d_{min} > 2t$. Ekkor C tud javítani t hibát. Tegyük fel, hogy egy $\mathbf{u}$ szót küldünk, és egy $\mathbf{u}'$-t kapunk vissza $d(\mathbf{u}, \mathbf{u}') \leq t$ Hamming távolsággal. A következő helyzet áll elő bármely $\mathbf{v} \neq \mathbf{u}$ kódszó esetén: $d(\mathbf{u}, \mathbf{v}) \geq d_{min} > 2t$, és a 1.3.8 tételben bizonyított háromszög egyenlőtlenség szerint

$$d(\mathbf{u}, \mathbf{u}') + d(\mathbf{u}', \mathbf{v}) \geq d(\mathbf{u}, \mathbf{v}) > 2t.$$

Tehát,

$$d(\mathbf{u}', \mathbf{v}) > 2t - d(\mathbf{u}, \mathbf{u}') \geq 2t - t = t \geq d(\mathbf{u}, \mathbf{u}').$$

Ezzel beláttuk a 2. állítást is. □

1.3.15. Következmény. Egy d_{min} minimum távolságú kód $(d-1)$-hibajelző és $\lfloor \frac{d-1}{2} \rfloor$ hibajavító.

1.3.16. Definíció. Ha C egy n hosszú, M méretű, d minimum távolságú kód, akkor C-t (n, M, d)-*kódnak* nevezzük.

1.3.17. Megjegyzés. Az előző tételben láttuk, hogy egy kód több hibát tud jelezni, mint kijavítani. Kényelmes megoldás lehetne, ha csak hibajelzéssel foglalkoznánk, és hibás üzenet esetén megkérnénk a feladót, hogy ismételje meg azt. A valóságban ez nem

mindíg kivitelezhető, (például valós idejű adatfolyamnál) emiatt a hibajavító kódolással is foglalkozni kell.

1.3.3. Legközelebbi szomszédok

Az 1.3.2. alfejezetben definiáltuk a két kód közti távolságot, és ennek segítségével végeztünk hibajelzést és hibajavítást is, ami a legközelebbi szomszéd módszerét használta. Most egy geometriai megközelítést adok az eddigi fogalmakhoz.

1.3.18. Definíció. Legyen Q egy q elemű ábécé, és legyen $\mathbf{u}$ egy n hosszú szó. Ekkor az $\mathbf{u}$ középpontú t sugarú *Hamming gömb* a

$$B_t(\mathbf{u}) = \{\mathbf{v} : \mathbf{v} \in Q^n \text{ és } d(\mathbf{u}, \mathbf{v}) \leq t\}$$

segítségével definiálható.

1.3.19. Lemma. *Legyen C egy kód. Ekkor C t-hibajavító akkor és csak akkor, ha bármely két különböző $\mathbf{u}, \mathbf{u}' \in C$ kódszóra a $B_t(\mathbf{u})$ és $B_t(\mathbf{u}')$ Hamming gömbök diszjunktak.*

Bizonyítás. Ha C egy t-hibajavító kód, akkor 1.3.9 szerint semelyik kódszótól mért, legfeljebb t hosszú Hamming távolságban nincs másik kódszó, ami épp B_t definíciója. Ha bármely két kódszóra a Hamming gömbök diszjunktak, ez azt jelenti, hogy tetszőleges $\mathbf{u}$ kódszó $B_t(\mathbf{u})$ gömbjében bármely $\mathbf{v}$ szóra teljesül, hogy egyértelműen $\mathbf{u}$-hoz van a legközelebb, ahogyan épp 1.3.9 definiálja a t-hibajavító kódokat. □

Az eddig megismert hibajavító kódolási fogalmak közt a következő tétel von kapcsolatot.

1.3.20. Tétel. *Legyen C kód. Ekkor a következő állítások ekvivalensek:*

1. *C t-hibajavító;*
2. *C minimális távolsága legalább $2t + 1$;*
3. *A legközelebbi szomszéd módszere visszaadja a küldött kódszót, amennyiben legfeljebb t hiba történt;*
4. *Ha $\mathbf{u}, \mathbf{u}' \in C$ különböző kódszavak, akkor a Hamming gömbök $B_t(\mathbf{u})$ és $B_t(\mathbf{u}')$ diszjunktak.*

Bizonyítás.

- 1. $\iff$ 2.: Az 1.3.14 tétel második állítása.
- 1. $\iff$ 3.: Ha C egy t-hibajavító kód, akkor 1.3.9 szerint egy szót annak a kódszónak kódol, amihez az a legközelebb van. Ha a legközelebbi szomszéd módszere visszaadja az $\mathbf{u}$ kódszót a legfeljebb t hibával kapott $\mathbf{v}$ szó alapján, az azt jelenti, hogy $d(\mathbf{u}, \mathbf{v}) < d(\mathbf{u}', \mathbf{v})$ minden olyan $\mathbf{u}' \in C$ kódszóra, amelyre $\mathbf{u} \neq \mathbf{u}'$, ami épp a t-hibajavító kód definíciója.
- 1. $\iff$ 4.: Az 1.3.19 lemmában épp ezt láttuk be.

Ezzel láthatjuk, hogy az állítások tényleg ekvivalensek. $\square$

Most nézzünk egy egyszerű kódolási módszert, és vizsgáljuk meg az eddig definiált fogalmakat rajta.

1.3.21. Definíció (*ismétlő kódok*)**.** Legyen $n \in \mathbb{N}$ és Q egy q-elemű ábécé, ahol $q \geq 2$. A Q-n értelmezett n hosszú ismétlő kódok kódszói mind $(s, s, s, \ldots, s)$ alakúak, ahol $s \in Q$.

1.3.22. Példa. Ha $Q = \{0, 1\}$, és $n = 3$, akkor $C = \{000, 111\}$.

1.3.23. Lemma. *Legyen $n \in \mathbb{N}$ és C egy n hosszú ismétlő kód egy q-elemű Q ábécé felett, ahol $q \geq 2$. Ekkor:*

- *A C kód $(n-1)$-hibajelző, de nem n-hibajelző kód.*
- *Ha $n = 2m + 1$, akkor C az m-hibajavító, de nem $(m+1)$-hibajavító.*
- *Ha $n = 2m$, akkor C az $(m-1)$-hibajavító, de nem m-hibajavító.*

1.3.24. Lemma. *Legyen $n \in \mathbb{N}$. Ekkor bármely n ismétlő kód minimum távolsága n.*

1.3.25. Definíció. Az n hosszú, q-elemű ábécén értelmezett ismétlő kódot (n, q, n)-*kódnak* nevezzük.

Ebben a példában láthattuk, hogy az ismétlő kódok könnyen megérthetők, egyszerű szerkezetűek, és alkalmasak hibajelzésre és hibajavításra. Ezek után felmerülhet a kérdés, hogy miért lesznek szükségesek ennél bonyolultabb módszerek? Erre a kérdésre a 3. fejezetben adjuk meg a választ. Mielőtt ezekre rátérnénk, szükséges egy másfajta elméleti alapozás.

2. fejezet

Algebrai struktúrák

A lineáris blokk-kódok bevezetéséhez, és azok jellemzéséhez szükséges a következő algebrai ismeretek áttekintése. Ennek kidolgozására a [1], [4], [6], [10] forrásokat használtam fel.

2.1. Csoportok

2.1.1. Definíció. A $\langle G, * \rangle$ *csoport* a G halmaz és a G-n értelmezett $*$ bináris művelet kettőse, ahol:

- a $*$ asszociatív: minden $a, b, c \in G$ esetén: $(a * b) * c = a * (b * c)$.
- létezik egy $e \in G$ egységelem, melyre $a * e = e * a = a$ minden $a \in G$-re.
- minden $a \in G$-hez létezik $b \in G$, melyre $a * b = e$. Itt b neve az a elem inverze.

2.1.2. Megjegyzés. Ha a szövegkörnyezetből egyértelmű, akkor $\langle G, * \rangle$ helyett röviden csak G-t írunk.

2.1.3. Definíció. Ha G csoportnak véges sok eleme van, akkor G neve *véges csoport*. Egy véges csoport elemszámát a *csoport rendjének* nevezzük, és $|G|$-vel jelöljük.

2.1.4. Definíció. Egy $\langle G, * \rangle$ csoport *kommutatív*, ha $a * b = b * a$, minden $a, b \in G$ esetén.

2.1.5. Megjegyzés. A kommutatív csoportokat Abel-csoportoknak is nevezzük, általában a bináris műveletet $+$-szal jelöljük.

2.1.6. Definíció. Legyen G csoport a $*$ műveletre, és H csoport a $\bullet$ műveletre. Azt mondjuk, hogy egy $\varphi : G \to H$ leképezés *csoporthomomorfizmus*, ha művelettartó, vagyis

ha tetszőleges $a, b \in G$ esetén $\varphi(a * b) = \varphi(a) \bullet \varphi(b)$. Ha φ kölcsönösen egyértelmű is a G és H halmazok között, akkor φ *izomorfizmus*. A G és a H izomorf csoportok, ha létezik közöttük izomorfizmus. Ennek jele $G \cong H$.

2.1.7. Példa. A $\langle \mathbb{Z}_5, + \rangle$ Abel-csoportot alkot, ahogy azt a következő szorzástábla mutatja:

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

2.1.1. Részcsoportok

2.1.8. Definíció. Legyen G egy csoport. Ha H részhalmaza G-nek, amely maga is csoport a G-beli műveletre nézve, akkor azt mondjuk, hogy H *részcsoportja* G-nek. Ezt úgy jelöljük, hogy $H \leq G$.

2.1.9. Megjegyzés. H-t *triviális részcsoportnak* nevezzük, ha $H = G$, vagy $H = \{e\}$, ahol $e \in G$ a egységelem.

2.1.10. Példa. A $H = \langle \{0, 3\}, + \rangle$ csoport részcsoportja a $G = \langle \mathbb{Z}_6, + \rangle$ csoportnak.

2.1.11. Példa. $\langle \mathbb{Z}, + \rangle \leq \langle \mathbb{Q}, + \rangle \leq \langle \mathbb{R}, + \rangle \leq \langle \mathbb{C}, + \rangle$.

2.1.2. Ciklikus csoportok

2.1.12. Definíció. Legyen $*$ asszociatív művelet az R halmazon, és $a \in R$. Ekkor tetszőleges n pozitív egészre legyen:

- $a^n = a * a * \cdots * a$ (n tényező).
- Ha a $*$-ra nézve van egy e egységelem, akkor legyen $a^0 = e$.
- Ha a invertálható, és inverze b, akkor legyen $a^{(-n)} = b^n$.

Ezek az a elem egész kitevőjű hatványai. Ha a műveletet $+$ jelöli, akkor hatvány helyett többszörösről beszélünk, és az na írásmódot alkalmazzuk.

2.1.13. Definíció. Minden $a \in G$ esetén az $\{a^n | n \in \mathbb{Z}\}$ halmaz egy részcsoportot generál G-ben, ennek a részcsoportnak a neve: *ciklikus részcsoport.* Az a elemet a részcsoport *generátorának* nevezzük. Az a elem által generált ciklikus részcsoportot $\langle a \rangle$-val jelöljük.

2.1.14. Definíció. Ha egy csoport minden eleme megkapható egy adott elemének hatványozásával, a csoportot *ciklikus csoportnak* nevezzük.

2.1.15. Példa. A $\langle \mathbb{Z}_5, + \rangle$ csoport ciklikus az $a = 2$ esetén.

2.1.16. Definíció. Ha G csoport, $a \in G$, akkor a legkisebb n számot, amire a^n a G csoport egységelemével egyenlő, az *a elem rendjének* nevezzük. Ha nem létezik ilyen n szám, akkor azt mondjuk, hogy *a végtelen rendű.*

2.1.3. Mellékosztályok

2.1.17. Definíció. Legyen H részcsoportja G-nek, és legyen $a \in G$ tetszőleges. Ekkor H csoport a szerinti *bal oldali mellékosztálya* az $\{a * h | h \in H\}$ halmaz, jelölése $a * H$. A jobb oldali mellékosztályt hasonlóan definiáljuk: $H * a = \{h * a | h \in H\}$.

2.1.18. Megjegyzés. Kommutatív csoportnál a jobb és bal oldali mellékosztályok megegyeznek.

2.1.19. Definíció. Legyen G csoport, H pedig egy részcsoportja. Legyen $a * H$ egy bal oldali mellékosztálya. Ekkor $b \in a * H$ akkor és csak akkor, ha $b = a * h$ valamely $h \in H$ elemre, tehát az egyszerűsítési szabály alapján $a^{(-1)} * b \in H$.

2.1.20. Definíció. Legyen H egy halmaz, és osszuk föl H-t nem üres, páronként diszjunkt halmazok egyesítésére. Egy ilyen felosztást H egy *partíciójának* nevezünk, a benne szereplő halmazokat pedig a *partíció osztályainak.*

2.1.21. Lemma. *H minden G-beli mellékosztályának ugyanannyi eleme van.*

Bizonyítás. Azt fogjuk megmutatni, hogy minden mellékosztálynak ugyanannyi eleme van, mint H-nak, ugyanis H önmagában is egy mellékosztály, mégpedig az e egységelem szerint. Legyen $a * h_1 \in a * H$ és $a * h_2 \in a * H$ két elem ugyanabban a mellékosztályban. Ha $a * h_1 = a * h_2$, akkor az egyszerűsítés szabálya szerint $h_1 = h_2$-nek kell teljesülnie. Tehát a mellékosztályok elemei egyértelműen meghatározhatóak H elemeiből. $\square$

2.1.22. Definíció. Minden mellékosztály rendelkezik a következő tulajdonságokkal: reflexivitás, szimmetria, tranzitivitás. Ezek következtében a mellékosztályok *ekvivalenciarelációt* alkotnak.

2.1.23. Lemma. *A H különböző G szerinti mellékosztályai diszjunktak.*

Bizonyítás. Legyen A és B különböző mellékosztályai H-nak, azaz $A \neq B$. Indirekt tegyük fel, hogy A és B nem diszjunktak, vagyis létezik egy c elem, amely mindkét mellékosztálynak eleme. Megmutatjuk, hogy ebből következik, hogy $A \subseteq B$. Legyen $b \in B$. Bármely $a \in A$-ra a és c ugyanahhoz a mellékosztályhoz tartoznak, (ugyanis $c \in A$). Továbbá b és c is ugyanahhoz a mellékosztályhoz tartoznak, ($c \in B$). A tranzitivitás miatt a-nak és b-nek ugyanabban a mellékosztályban kell lennie, vagyis A minden eleme eleme B-nek is, tehát $A \subseteq B$. Az érvelést megfordítva azt lájuk, hogy $B \subseteq A$. Ezzel beláttuk, hogy $A = B$, így ellentmondásra jutottunk. $\square$

2.1.24. Tétel (Lagrange). *Véges csoport minden részcsoportjának rendje osztója a csoport rendjének.*

Bizonyítás. A 2.1.21 lemma szerint ha $H \leq G$, akkor a H szerinti bal oldali mellékosztályok a G egy partícióját adják. A 2.1.23 alapján G elemszámát úgy kaphatjuk meg, hogy H elemszámát megszorozzuk az osztályok számával. Így $|H|$ osztója $|G|$-nek. $\square$

2.1.25. Megjegyzés. A Lagrange-tétel felírható szimbólumokkal a következőképp: Ha $|G| < \infty$ és $H \leq G$, akkor $|H| \mid |G|$.

A Lagrange-tétel egy fontos következménye a következő lemma.

2.1.26. Lemma. *Minden prímrendű csoport ciklikus.*

Bizonyítás. Legyen G prímrendű, legyen $a \in G$ és legyen e a egységelem. Legyen $H = \langle a \rangle$ az a elem által generált ciklikus részcsoport. Ekkor $a \in H$ és $e \in H$, vagyis $|H| \neq 1$. Továbbá a Lagrange-tétel szerint H rendjének osztania kell G rendjét. Mivel G prímrendű, így ez csak akkor lehetséges, ha $|H| = |G|$. Ekkor a elem generálja G csoportot, tehát G ciklikus csoport. $\square$

2.1.27. Definíció. Legyen G csoport és H részcsoportja G-nek. A különböző H szerinti bal mellékosztályok számát a H részcsoport G-beli *indexének* nevezzük, jelölése $|G : H|$.

2.1.28. Következmény. Bármely két H-szerinti bal oldali mellékosztály vagy megegyezik, vagy diszjunkt, és a bal oldali mellékosztályok uniója G. Ugyanez a jobb oldali mellékosztályokra is igaz. Ha G véges, akkor $|G| = |H| \cdot |G : H|$.

2.2. Testek

A csoportok segítségével vezessünk be egy újabb algebrai struktúrát.

2.2.1. Definíció. A *test* egy $\mathbb{F}$ halmaz két bináris operációval, $+$ és $\cdot$, amikre a következők teljesülnek:

1. $\langle \mathbb{F}, + \rangle$ kommutatív csoport a 0 egységelemmel.
2. $\langle \mathbb{F} \backslash \{0\}, \cdot \rangle$ kommutatív csoprt az 1 egységelemmel.
3. Disztribúció: minden $x, y, z \in \mathbb{F}$-re teljesül, hogy $x(y+z) = xy + xz$.

2.2.2. Következmény.

1. Minden testen értelmezve van a kivonás: $a - b = a + (-b)$ és ha $b \neq 0$, az osztás: $a : b = ab^{(-1)}$.
2. Két nem nulla elem szorzata is nem nulla.
3. Mivel az előző definíció 2. pontja csak nem nulla elemekre definiálja a szorzást, a 0 esetében ezt külön meg kell tenni: minden $x \in \mathbb{F}$ esetén $x \cdot 0 = 0$.
4. Minden testben tudunk a nem nulla elemmel egyszerűsíteni: $ax = ay$ köv: $x = y$ (ha $a \neq 0$).

2.2.3. Példa. A valós számok halmaza testet alkot, viszont az egész számok halmaza nem.

2.2.4. Lemma. *Legyen $a, b \in \mathbb{F}$. Ekkor:*

1. $(-1) \cdot a = -a$;
2. $ab = 0$ *csak akkor lehet, ha* $a = 0$ *vagy* $b = 0$.

2.2.1. A $\mathbb{Z}_p$ test

2.2.5. Definíció. Legyenek a, b és $m > 1$ egész számok. Azt mondjuk, hogy a *kongruens* b-vel modulo m, ha $m|(a-b)$. Jelölése: $a \equiv b \ (mod\ m)$.

2.2.6. Definíció. Tetszőleges $m > 1$ egész szám esetén $\mathbb{Z}/(mod\ m)$ -el jelöljük a $\{0, 1, \ldots, m-1\}$ halmazzal és a $+_m$ és $\cdot_m$ műveletekkel definiált algebrai struktúrát, ahol:

- $x +_m y \equiv x + y \ (mod\ m)$
- $x \cdot_m y \equiv x \cdot y \ (mod\ m)$

Vezessük be a következő jelöléseket: $\mathbb{Z}/(mod\ m) = \mathbb{Z}_m$; $+_m = +$; $\cdot_m = \cdot$.

2.2.7. Példa. A 2.1.7 példában már beláttuk, hogy a $\mathbb{Z}_5$ tényleg test.

A $\mathbb{Z}_4$-et vizsgálva viszont már nem kapunk testet, ugyanis $2 \cdot 2 = 0$, azaz nem létezik minden elemnek multiplikatív inverze.

Az előző példából láthatjuk, hogy bizonyos m-ek esetén $\mathbb{Z}_m$ test, bizonyos m-ek esetén pedig nem.

2.2.8. Tétel. *A $\mathbb{Z}_m$ test $\Longleftrightarrow$ m prímszám.*

Bizonyítás. Tegyük fel, hogy m összetett szám, és legyen $m = ab$, ahol $1 < a, b < m$. Ekkor $a \neq 0$ és $b \neq 0$. Viszont $0 = m = ab$ a $\mathbb{Z}_m$ testben. Ez ellentmond a 2.2.4 lemmának, tehát $\mathbb{Z}_m$ nem test.

Most legyen m prím. Bármely nem nulla $a \in \mathbb{Z}_m$ (azaz $0 < a < m$) tudjuk, hogy a és m relatív pramek. Tehát létezik két egész szám, u és v, ahol $0 \leq u \leq m - 1$, hogy $ua + vm = 1$, azaz $ua \equiv 1 \ (mod\ m)$. Tehát $u = a^{-1}$. Emiatt minden elemnek létezik multiplikatív inverze, azaz $\mathbb{Z}_m$ test. $\square$

2.2.9. Definíció. Legyen $\mathbb{F}$ egy test. $\mathbb{F}$ *karakterisztikájának* nevezzük azt a legkisebb p számot, melyre teljesül, hogy $p \cdot 1 = 0$, ahol 1 a multiplikatív egységelem. Ha nem létezik ilyen p, akkor azt mondjuk, hogy az $\mathbb{F}$ karakterisztikája 0.

2.2.10. Példa. A $\mathbb{Q}$,$\mathbb{R}$,$\mathbb{C}$ testek karakterisztikája 0. Ha p prím, $\mathbb{Z}_p$ karakterisztikája p.

2.2.11. Tétel. *Egy test karakterisztikája 0, vagy prímszám.*

Bizonyítás. Az 1 nem lehet a karakterisztika, ugyanis $1 \cdot 1 = 1 \neq 0$.

Tegyük fel, hogy a p karakterisztika összetett szám. Legyen $p = nm$, ahol $1 < n, m < p$. Jelölje $a = n \cdot 1$ és $b = m \cdot 1$. Ekkor:

$$a \cdot b = (n \cdot 1)(m \cdot 1) = \left(\sum_{i=1}^{n} 1\right) \cdot \left(\sum_{i=1}^{m} 1\right) = (mn) \cdot 1 = p \cdot 1 = 0.$$

Az 2.2.4 lemma szerint $a = 0$ vagy $b = 0$, azaz $m \cdot 1 = 0$ vagy $n \cdot 1 = 0$, ami ellentmond a karakterisztika definíciójának, ugyanis találtunk p-nél kisebb számot. $\square$

2.2.12. Definíció. Legyen $\mathbb{K}$ és $\mathbb{F}$ két test, és legyen $\mathbb{F}$ részhalmaza $\mathbb{K}$-nak. Az $\mathbb{F}$ testet a $\mathbb{K}$ *résztestének* nevezzük, ha a $\mathbb{K}$-n értelmezett $+$ és $\cdot$ műveletek megszorítása $\mathbb{F}$-re megegyezik az $\mathbb{F}$ műveleteivel.

2.2.13. Tétel. *A p karakterisztikájú $\mathbb{F}$ véges test p^n elemet tartalmaz egy megfelelő $n \geq 1$ szám esetén.*

Bizonyítás. Válasszunk egy $\alpha_1 \in \mathbb{F}^*$ elemet. Azt állítjuk, hogy $0 \cdot \alpha_1, 1 \cdot \alpha_1, \ldots, (p-1) \cdot \alpha_1$ páronként különbözőek. Valóban, ugyanis, ha $i \cdot \alpha_1 = j \cdot \alpha_1$ valamely $0 \leq i \leq j \leq p-1$, akkor $(j-i) \cdot \alpha_1 = 0$ és $0 \leq j - i \leq p-1$. Mivel $\mathbb{F}$ karakterisztikája p, ezért $j - i = 0$ lehet csak, tehát $i = j$.

Ha $\mathbb{F} = \{0 \cdot \alpha_1, 1 \cdot \alpha_1, \ldots, (p-1) \cdot \alpha_1\}$, készen is vagyunk. Ha nem, választunk egy $\alpha_2 \in \mathbb{F} \backslash \{0 \cdot \alpha_1, 1 \cdot \alpha_1, \ldots, (p-1) \cdot \alpha_1\}$ elemet. Azt állítjuk, hogy $a_1\alpha_1 + a_2\alpha_2$ páronként különbözők minden $0 \leq a_1, a_2 \leq p-1$ esetén. Valóban, ha

$$a_1\alpha_1 + a_2\alpha_2 = b_1\alpha_1 + b_2\alpha_2 \tag{2.1}$$

tetszőleges $0 \leq a_1, a_2, b_1, b_2 \leq p-1$ esetén, akkor $a_2 = b_2$. Különben, (2.1) alapján $\alpha_2 = (b_2 - a_2)^{-1}(a_1 - b_1)\alpha 1$. Ez ellentmond α_2 választásának. Mivel $a_2 = b_2$, ezért a (2.1)-ból közvetlenül következik, hogy $a_1 = b_1$ is teljesül.

Mivel $\mathbb{F}$-nek véges sok eleme van, ezzel a módszerrel választhatunk további $\alpha_1, \ldots, \alpha_n$ számokat, hogy

$$\alpha_i \in \mathbb{F} \backslash \{a_1 \cdot \alpha_1, \ldots, a_{i-1} \cdot \alpha_{i-1} : a_1, \cdots a_{i-1} \in \mathbb{Z}_p\} \text{ minden } 2 \leq i \leq n,$$

és

$$\mathbb{F} = \{a_1 \cdot \alpha_1, \ldots, a_n \cdot \alpha_n : a_1, \cdots a_n \in \mathbb{Z}_p\}.$$

Továbbá azt is meg tudjuk mutatni az eddigiekhez hasonlóan, hogy $a_1 \cdot \alpha_1, \ldots, a_n \cdot \alpha_n$ páronként különbözők minden $a_i \in \mathbb{Z}_p, i = 1, \ldots, n$. Ebből következik, hogy $|\mathbb{F}| = p^n$. $\square$

2.3. Gyűrűk

2.3.1. Definíció. Ha egy algebrai struktúrára a testre vonatkozó axiómák közül csak az nem teljesül, hogy a szorzásra nézve van inverze, akkor a struktúrát egységelemes, kommutatív *gyűrűnek* nevezzük.

2.3.2. Definíció. Legyen R egy gyűrű. Ha S részhalmaza R-nek, amely maga is gyűrű az R-beli műveletekre nézve, akkor azt mondjuk, hogy S *részgyűrűje* R-nek. Ezt $S \leq R$-el jelöljük.

2.3.3. Definíció. Legyen R gyűrű és $X \subseteq R$. Az X által R-ben *generált részgyűrű* a legszűkebb olyan részgyűrűje R-nek, amely X-et tartalmazza. Jele: $\langle X \rangle$.

2.3.4. Definíció. Ha $\varphi : R \to S$ egy gyűrűhomomorfizmus, akkor

a) $Im(\varphi) = \{\varphi(r) | r \in R\} \subseteq S$ a φ *képe*, vagyis a φ függvény értékkészlete.

b) $Ker(\varphi) = \{r \in R : \varphi(r) = 0\} \subseteq R$ a φ *magja*. Itt 0 az S gyűrű nulleleme.

2.3.5. Példa. Legyen $\varphi : \mathbb{Z} \to \mathbb{Z}_n$. Ekkor $Im(\varphi) = \{0, 1, \ldots, n-1\}$ és $Ker(\varphi) = \{a \cdot n | a \in \mathbb{Z}\}$.

2.3.6. Definíció. Egy R gyűrű egy I részhalmazát *balideálnak* nevezzük, ha az összeadásra nézve részcsoport, és minden $a \in I, r \in R$ esetén $ra \in I$. Hasonlóképpen I jobbideál, ha részcsoport, és tetszőleges $a \in I, r \in R$ esetén $ar \in I$.

Az I (kétoldali) *ideál*, ha bal- és jobbideál is egyúttal. Azt, hogy I ideál R-ben, $I \triangleleft R$ jelöli.

2.3.7. Definíció. Egy ideál pontosan akkor *főideál*, ha egy elemmel generálható. Jelölése: $(x) = I$, ahol I ideál.

2.3.8. Definíció. Egy R kommutatív, egységelemes gyűrűt *főideálgyűrűnek* nevezünk, ha minden ideűlja főideál.

2.3.1. Polinomgyűrűk

2.3.9. Definíció. Legyen $\mathbb{F}$ test. Ekkor az

$$\mathbb{F}[x] = \left\{ \sum_{i=0}^{n} a_i x^i : a_i \in \mathbb{F}, n \geq 0 \right\}$$

halmazt $\mathbb{F}$ feletti *polinomgyűrűnek* nevezzük.

2.3.10. Megjegyzés.

1. Az $f(x) = \sum_{i=0}^{n} a_i x^i$ polinomban az n számot a *polinom fokának* nevezzük, jelölése: $deg(f(x)) = n$, ha $a_n \neq 0$.

2. Ha $a_n = 1$, akkor a polinomot *normált polinomnak* nevezzük.

3. Az $f(x)$ *irreducibilis*, ha nem bontható fel két, alacsonyabb fokú polinom szorzatára.

2.3.11. Definíció. Legyen $f(x) \in \mathbb{F}[x]$ polinom, ahol $n \geq 1$. Ekkor bármely $g(x) \in \mathbb{F}[x]$ polinomhoz egyértelműen létezik egy $(s(x), r(x))$ páros, ahol $deg(r(x)) < deg(f(x))$, és $g(x) = s(x)f(x)+r(x)$ teljesül. Az $r(x)$ polinomot a $g(x)$ polinom $f(x)$-el vett *maradéknak* nevezzük, jelölése: $r(x) = g(x)\ (mod\ f(x))$.

Sok hasonlóság van a $\mathbb{Z}$ pozitív számok gyűrűje, és az $\mathbb{F}[x]$ polinomgyűrű közt. Mivel $\mathbb{Z}$ és egy $m > 1$ szám segítségével konstruálható az $\mathbb{Z}/(mod\ m) = \mathbb{Z}/(m) = \mathbb{Z}_m$ gyűrű, hasonlóan az $\mathbb{F}[x]/(f(x))$ gyűrű elkészíthető adott $f(x)$ polinom segítségével, ahol $deg(f(x)) \geq 1$.

2.3.12. Definíció. Legyen $\mathbb{F}[x]$ polinomgyűrű, továbbá legyen adott $f(x)$ polinom, ahol $n = deg(f(x)) \geq 1$. Ekkor az $\mathbb{F}[x]/(f(x))$ *faktorgyűrű* a következő:

- $\mathbb{F}[x]/(f(x)) := \left\{\sum_{i=0}^{n-1} a_i \cdot x^i : a_i \in \mathbb{F}, n \geq 1\right\}$
- $g(x) +_f h(x) := (g(x) + h(x)(mod\ f(x)))$
- $g(x) \cdot_f h(x) := (g(x) \cdot h(x)(mod\ f(x)))$

Ezzel a definícióval az $\mathbb{F}[x]/(f(x))$ gyűrűt alkot.

2.3.13. Állítás. *Legyen $f(x)$ polinom $\mathbb{F}[x]$ felett, és legyen $deg(f(x)) \geq 1$. Ekkor $\mathbb{F}[x]/(f(x))$ az előbb definiált összeadásra és szorzásra nézve gyűrűt alkot.*

2.3.14. Állítás. *Minden faktorgyűrű főideálgyűrű.*

2.3.15. Állítás. *Az $\mathbb{F}[x]/(f(x))$ test $\iff f(x)$ irreducibilis.*

2.3.16. Példa. Vizsgáljuk a $\mathbb{Z}_2[x]/(1 + x + x^2) = \{0, 1, x, 1 + x\}$ halmazt a $+_f$ és $\cdot_f$ műveletekkel. Ekkor a 2.3.13 állítás szerint $\mathbb{Z}_2[x]/(1 + x + x^2)$ gyűrű, továbbá, mivel $1 + x + x^2$ irreducibilis $\mathbb{Z}_2$ felett, így test is.

$+_f$	0	1	x	$1+x$
0	0	1	x	$1+x$
1	1	0	$1+x$	x
x	x	$1+x$	0	1
$1+x$	$1+x$	x	1	0

$\cdot_f$	0	1	x	$1+x$
0	0	0	0	0
1	0	1	x	$1+x$
x	0	x	$1+x$	1
$1+x$	0	$1+x$	1	x

2.3.17. Megjegyzés. Ezentúl, ha polinomgyűrűkkel számolunk, a $+_f$ helyett $+$-ot, a $\cdot_f$ helyett $\cdot$-ot írunk.

2.4. Vektorterek

2.4.1. Definíció. Legyen $\mathbb{F}$ test. Az $\mathbb{L}$ halmazt $\mathbb{F}$ feletti *vektortérnek* nevezzük, ha adottak $\mathbb{L}$-en az összeadás $(+)$ és skalárszorzás $(\cdot)$ műveletek, amikre igaz:

1. $\langle \mathbb{L}, + \rangle$ kommutatív csoport.
2. Asszociativitás: $(st)\mathbf{a} = s(t\mathbf{a})$ minden $s, t \in \mathbb{F}$ és $\mathbf{a} \in \mathbb{L}$ esetén.
3. Disztribúciós szabályok: $t(\mathbf{a} + \mathbf{b}) = t\mathbf{a} + t\mathbf{b}$ és $(s + t)\mathbf{a} = s\mathbf{a} + t\mathbf{a}$ ahol $s, t \in \mathbb{F}$ és $\mathbf{a}, \mathbf{b} \in \mathbb{L}$.
4. $1\mathbf{a} = \mathbf{a}$, ahol 1 az $\mathbb{F}$ egységeleme, és $\mathbf{a} \in \mathbb{L}$.

2.4.2. Következmény.

5. $0\mathbf{a} = 0$ minden $\mathbf{a} \in \mathbb{L}$ vektorra.
6. $(-1)\mathbf{a} = -\mathbf{a}$ minden $\mathbf{a} \in \mathbb{L}$ vektorra.
7. $t0 = 0$ minden $t \in \mathbb{F}$ skalárra.

2.4.3. Példa. Az n hosszú bináris szavakra a $\mathbb{Z}_2^n$ halmaz lineáris teret alkot.

2.4.4. Definíció. Legyen $\mathbb{L}$ lineáris tér. Egy nemüres $\mathbb{K}$ részhalmazát *lineáris altérnek* nevezzük, ha az zárt az összeadásra, és a skalárral való szorzásra.

2.4.5. Definíció. Az $a_1, \ldots, a_m$ vektorok *lineáris kombinációjának* nevezzük az $t_1 a_1 + \cdots + t_m a_m$ vektorokat, ahol $t_1, \ldots, t_m$ skalárok.

2.4.6. Állítás. *Legyenek $a_1, \ldots, a_m$ vektorok az $\mathbb{L}$ lineáris térben. Ekkor ezeknek a vektoroknak az összes lineáris kombinációja alteret alkot $\mathbb{L}$-ben, ami a legkisebb olyan altér, ami az említett vektorokat tartalmazza. Nevezzük ezt az $a_1, \ldots, a_m$ vektorok által generált altérnek.*

2.4.7. Megjegyzés. Az $a_1, \ldots, a_m$ vektorok kifeszítik az $\mathbb{L}$ lineáris teret, ha minden $\mathbb{L}$-beli vektor megkapható az $a_1, \ldots, a_m$ vektorok lineáris kombinációjából.

2.4.8. Definíció. Legyen $\mathbb{L}$ lineáris tér:

1. A nem nulla $a_1, \ldots, a_m$ vektorok *lineárisan függetlenek*, ha egyik sem lineáris kombinációja a többinek.

2. Az $\mathbb{L}$ *bázisán* azokat a lineárisan független vektorokat értjük, amelyek kifeszítik $\mathbb{L}$-t, azaz minden $a \in \mathbb{L}$ egyértelműen felírható báziselemek lineáris kombinációjaként. Ha $\mathbb{L}$-nek van egy véges bázisa, akkor véges dimenziós lineáris térnek nevezzük.

2.4.9. Megjegyzés. Legyen $\mathbb{F}$ egy r elemű test. Ekkor minden $\mathbb{F}$ felett értelmezett k-dimenziós lineáris térnek r^k eleme van.

3. fejezet

Lineáris kódok, blokk-kódok

Az első fejezetben definiáltuk a kódábécét, kódokat, távolságot, a hibajelzést és -javítást, és megvizsgáltuk az ismétlőkódokat. Ebben a fejezetben először korlátokat állítunk a kódokra, majd mutatunk egy olyan kódolási eljárást, ami az ismétlőkódoknál élesebb eredményt ad ezekre a korlátokra. Ebben a fejezetben a [7] és [8] szakirodalmakat használtam fel.

A továbbiakban a kódokat alkotó szimbólumok legyenek a 0 és 1, az összeadás a bináris összeadás, a szorzás a bináris szorzás.

3.1. Korlátok

Az 1. fejezetben láttuk, hogy kellően hosszú ismétlő kódokkal nagy mennyiségű hibát tudunk javítani, ám a valóságban nem áll rendelkezésünkre tetszőlegesen nagy kódhosszúság. Ebben a fejezetben először megnézzük, hogy adott n kódhosszúság és d_{min} kódtávolság esetén legfeljebb mekkora kódot lehet konstruálni.

3.1.1. Tétel (Singleton-korlát)**.** *Egy M kódszóból álló, n hosszú és d_{min} kódtávolságú kódra $M \leq q^{n-d_{min}+1}$.*

Bizonyítás. Legyen k olyan természetes szám, melyre

$$q^{k-1} < M \leq q^k.$$

Mivel a $k-1$ különböző sorozatok száma q^{k-1}, ezért $q^{k-1} < M$ miatt létezik két olyan $\mathbf{c}$ és $\mathbf{c}'$ kódszó, melyek az első $k-1$ koordinátában megegyeznek. Ezekre

$$d(\mathbf{c}, \mathbf{c}') \leq n - (k-1) = n - k + 1,$$

következésképpen

$$d_{min} \leq n - k + 1,$$

azaz

$$M \leq q^k \leq q^{n-d_{min}+1}.$$

□

3.1.2. Következmény. Ha $M = q^k$, vagyis k hosszú üzenetből készül az n hosszú kódszó $n-k$ hosszú redundáns rész hozzáadásával, akkor azt mondjuk, hogy az ilyen kódunk (n, k) paraméterű. Ezt a kódot *blokk-kódnak* is nevezzük. Ebben az esetben a Singleton-korlát alakja: $d_{min} \leq n - k + 1$.

3.1.3. Definíció. Ha egy kód éles a Singleton-korlátra, azt mondjuk, hogy a kód *maximális távolságú*, vagy MDS (maximum distance separable).

3.1.4. Tétel (Hamming-korlát)**.** *Ha egy (n, k) paraméterű kód t hibát tud javítani, akkor*

$$\sum_{i=0}^{t} \binom{n}{i} (q-1)^i \leq q^{n-k}. \tag{3.1}$$

Bizonyítás. A bizonyításhoz a 1.3.18 definícióban bevezetett Hamming gömböket használjuk fel. Egy kódszó közepű gömb álljon azokból a vektorokból, melyek a kódszóból legfeljebb t hibával keletkeznek. $(q-1)^i$ olyan vektor van, amely az adott kódszótól pontosan i koordinátában tér el, ezért egy ilyen gömb $\sum_{i=0}^{t} \binom{n}{i}(q-1)^i$ vektort tartalmaz. A kód akkor tud t hibát javítani, ha a q^k darab kódszó körüli gömbök diszjunktak (lásd 1.3.20 tétel). Ekkor viszont az összes gömbben lévő vektorok száma kisebb vagy egyenlő, mint q^n. □

3.1.5. Megjegyzés. Az (3.1) egyenlet a kód létezésének szükséges feltétele.

3.1.6. Definíció. Azokat a kódokat, ahol a Hamming-korlát élesen teljesül, *perfekt kódoknak* nevezzük.

Írjuk fel a Hamming-korlátot bináris esetben:

$$\sum_{i=0}^{t} \binom{n}{i} \leq 2^{n-k}.$$

3.2. Lineáris kódok bevezetése

3.2.1. Definíció. Egy C kód *lineáris*, ha a C lineáris tér, azaz, ha minden $\mathbf{c}, \mathbf{c}' \in C$ esetén $\mathbf{c} + \mathbf{c}' \in C$ is teljesül.

3.2.2. Megjegyzés. A C lineáris térre kimondható a lineáris függetlenség, és a bázis definíciója.

3.2.3. Definíció. A $g_1, g_2, \ldots, g_j \in C$ vektorok lineárisan függetlenek, ha $\alpha_i \in \{0, 1\}$ mellett

$$\sum_{i=1}^{j} \alpha_i g_i = 0$$

csak úgy állhat elő, ha $\alpha_i = 0$ minden $i = 1, 2, \ldots, j$-re.

3.2.4. Definíció. A $g_1, g_2, \ldots, g_j \in C$ vektoroka C lineáris tér egy bázisát alkotják, ha lineárisan függetlenek, továbbá minden $\mathbf{c} \in C$ vektor egyértelműen előállítható

$$\sum_{i=1}^{k} u_i g_i$$

alakban, ahol $u_i = \{0, 1\}$ minden $i = 0, 1, \ldots, k$-ra.

3.2.5. Következmény. Az előző egyenlőség mátrix alakban: $\mathbf{c} = \mathbf{uG}$, ahol $\mathbf{u} = (u_1, \ldots u_k)$, $\mathbf{G}$ pedig a bázisvektorokból, mint sorvektorokból álló mátrix. Tehát egy k-dimenziós és egy n-dimenziós vektort rendelünk egymáshoz lineáris transzformációval, kölcsönösen egyértelmű módon.

Azt mondjuk, hogy az $\mathbf{u}$ üzenethez a $\mathbf{c}$ kódszó tartozik. A k-dimenziós $\mathbf{u}$ vektorokkal 2^k féle üzenetet fejezhetünk ki, és ezeket kódoljuk a C kóddal.

3.2.6. Definíció. Az üzenetekhez a kódszavakat a $\mathbf{G}$ mátrix segítségével rendeljük hozzá, vagyis a $\mathbf{G}$ mátrix jelöli ki az n-dimenziós vektortérnek a kódot jelentő C alterét. A $\mathbf{G}$ mátrixot a C kód *generátormátrixának* nevezzük.

Ha a kódot, mint halmazt vizsgáljuk, akkor ugyanazokat a kódszavakat többféle $\mathbf{G}$ generátormátrix is megadhatja. Most nézzünk egy olyan szabályt, amivel egyértelmű lesz a $\mathbf{G}$ és C közti összefüggés.

3.2.7. Definíció. Egy (n, k) paraméterű lineáris kód *szisztematikus*, ha a k hosszúságú üzenetet egészítjük ki az $n - k$ redundáns karakterrel.

Szisztematikus kód esetén a $\mathbf{G}$ generátormátrix a következő alakot veszi fel:

$$\mathbf{G} = (\mathbf{I}_k, \mathbf{B}),$$

ahol $\mathbf{I}_k$ a $k \times k$ méretű egységmátrix, $\mathbf{B}$ pedig a $k \times (n-k)$ méretű mátrix.

Az $\mathbf{u}$ üzenethez tartozó $\mathbf{c}$ kódszó:

$$\mathbf{c} = (u_1, \ldots, u_k, c_{k+1}, \ldots, c_n). \tag{3.2}$$

3.2.8. Megjegyzés. Az (3.2) egyenlet első k koordinátájából álló szegmensét *üzenetszegmensnek*, az utolsó $n-k$ koordinátájából állót pedig *paritásszegmensnek* nevezzük.

A hibajavító kódolásban és dekódolásban a $\mathbf{G}$ generátormátrix mellett van még egy fontos mátrix, amit a következőképp definiálunk:

3.2.9. Definíció. Legyen $\mathbf{H}$ egy $n-k$ sorból és n oszlopból álló mátrix. Ha

$$\mathbf{H}\mathbf{c}^T = 0$$

akkor és csak akkor teljesül, ha $\mathbf{c} \in C$, akkor $\mathbf{H}$-t a C kód *paritásellenőrző mátrixának* (röviden paritásmátrixának) nevezzük.

3.2.10. Megjegyzés. A $\mathbf{H}$ paritásmátrix segítségével meg tudjuk mondani, hogy egy szó kódszó-e.

A következő tételekben kapcsolatot vonunk a generátormátrix és a paritásmátrix között.

3.2.11. Tétel. *Ha* $\mathbf{G}$ *és* $\mathbf{H}$ *ugyanazon* C *lineáris kód generátormátrixa illetve paritásmátrixa, akkor*

$$\mathbf{H}\mathbf{G}^T = 0.$$

Bizonyítás. Jelölje Q^k a k hosszú bináris sorozatok halmazát. Ekkor minden $\mathbf{u} \in Q^k$-hoz létezik $\mathbf{c} \in C$ amire $\mathbf{c} = \mathbf{u}\mathbf{G}$. Ugyanakkor $\mathbf{c} \in C$ miatt $\mathbf{H}\mathbf{c}^T = 0$ azaz

$$\mathbf{H}\mathbf{c}^T = \mathbf{H}(\mathbf{u}\mathbf{G})^T = \mathbf{H}\mathbf{G}^T\mathbf{u}^T = 0.$$

Az utolsó egyenlőség pedig csak úgy állhat fönn minden $\mathbf{u} \in Q^k$-ra, ha $\mathbf{H}\mathbf{G}^T = 0$. $\square$

3.2.12. Megjegyzés. Ha $\mathbf{H}$ kielégíti az $\mathbf{H}\mathbf{G}^T = 0$ egyenletet, akkor még nem biztos, hogy $\mathbf{H}$ paritásmátrix.

3.2.13. Lemma. *Ha egy* $\mathbf{c} = (c_1, \ldots, c_k, c_{k+1}, \ldots, c_n)$ *vektorra* $\mathbf{H}\mathbf{c}^T = 0$, *akkor* $\mathbf{c}$ *kódszó.*

Bizonyítás. Az előző tétel alapján

$$0 = \mathbf{H}\mathbf{c}^T = (\mathbf{A}, \mathbf{I}_{n-k})\mathbf{c}^T = (-\mathbf{B}^T, \mathbf{I}_{n-k})\mathbf{c}^T$$

miatt

$$-(c_1, c_2, \ldots, c_k)\mathbf{B} + (c_{k+1}, \ldots, c_n) = 0,$$

tehát

$$(c_{k+1}, \ldots, c_n) = (c_1, c_2, \ldots, c_k)\mathbf{B}$$

következésképp

$$(c_1, \ldots, c_k, c_{k+1}, \ldots, c_n) = (c_1, c_2, \ldots, c_k)(\mathbf{I}_k, \mathbf{B}),$$

tehát $\mathbf{c}$ tényleg kódszó. $\square$

3.2.14. Tétel. *Minden lineáris kódnak van paritásmátrixa.*

Bizonyítás. Először szisztematikus esetben látjuk be a tételt. Legyen

$$\mathbf{G} = (\mathbf{I}_k, \mathbf{B})$$

alakú, keressük $\mathbf{H}$-t

$$\mathbf{H} = (\mathbf{A}, \mathbf{I}_{n-k})$$

alakban. Az előző tétel szerint

$$\mathbf{H}\mathbf{G}^T = (\mathbf{A}, \mathbf{I}_{n-k})(\mathbf{I}_k, \mathbf{B}) = \mathbf{A} + \mathbf{B}^T = 0.$$

Azaz

$$\mathbf{A} = -\mathbf{B}^T$$

kell teljesüljön. Azt kell megmutatni, hogy az így kapott $\mathbf{H}$ mátrix tényleg paritásmátrix, ugyanis az előző megjegyzésben láttuk, hogy ez nem garantált. A paritásmátrix definíciója miatt tudjuk, hogy $\mathbf{H}\mathbf{c}^T = 0$ minden $\mathbf{c} \in C$ kódszóra. A másik irányt a 3.2.13 lemmában láttuk be. Ezzel igazoltuk a tételt szisztematikus esetre.

A tetszőleges esethez azt mutatjuk meg, hogy minden lineáris kód általános értelemben szisztematikus. Egy lineáris kódot általános értelemben szisztematikusnak nevezünk, ha létezik egy $\mathbf{G}$ generátormátrixa és $i_1, \ldots, i_k$ egész számok úgy, hogy $\mathbf{G}$ i_j-edik oszlopa a j-edik koordinátában 1, a többi koordinátában 0. Egy ilyen $\mathbf{G}$ mátrix oszlopcseréinek egy

sorozatával egy szisztematikus mátrixot kapunk, ahol elő tudjuk állítani a paritásmátrixot, majd az előző oszlopcserék inverzével az eredeti kód paritásmátrixát kapjuk meg. Az itt definiált általános értelemben szisztematikus mátrixokhoz a Gauss-elimináció módszerével juthatunk. A Gauss-elimináció azért alkalmazható, mert $\mathbf{G}$ oszlopai bázist alkotnak, és a transzformációk után is bázist fognak alkotni a kód linearitása miatt. $\square$

Még az 1.3.12 következményben láttuk, hogy a d_{min} meghatározása $\frac{|C|(|C|-1)}{2}$ lépést vesz igénybe. Vezessünk be egy új fogalmat, aminek a segítségével d_{min} gyorsabban meghatározható.

3.2.15. Definíció. Egy $\mathbf{c}$ vektor *súlya* a koordinátái között lévő nem nulla elemek száma, jelölése $w(\mathbf{c})$.

3.2.16. Definíció. Egy C kód *minimális súlyán* a

$$w_{min} = \min_{\substack{\mathbf{c}\in C \\ \mathbf{c}\neq 0}} w(\mathbf{c})$$

számot értjük.

3.2.17. Tétel. *Ha C lineáris kód, akkor a kódtávolsága megegyezik a minimális súlyával, azaz $d_{min} = w_{min}$.*

Bizonyítás. A d_{min} definíciójából, a w_{min} definíciójából, és a C kód linearitásából következik, hogy

$$d_{min} = \min_{\mathbf{c}\neq\mathbf{c}'} d(\mathbf{c},\mathbf{c}') = \min_{\mathbf{c}\neq\mathbf{c}'} w(\mathbf{c}-\mathbf{c}') = \min_{\mathbf{c}''} w(\mathbf{c}'') = w_{min}.$$

$\square$

Mivel a w_{min} kiszámolható $|C|-1$ lépésben, így a minimális súly fogalmának segítségével egy gyorsabb módot találtunk d_{min} meghatározására.

3.3. Hibajavítás szindrómákkal

Az előzőek segítségével most nézzünk egy konkrét dekódolási eljárást. A felhasznált algebrai hátteret a 2.1.24 pontban kimondott, és bizonyított Lagrange-tétel szolgáltatja.

A módszer ismertetése előtt bevezetünk egy definíciót:

3.3.1. Definíció. Az $\mathbf{s} = \mathbf{e}\mathbf{H}^T$ mennyiséget *szindrómának* nevezzük.

Legyen az adott kódszó $\mathbf{c}$, a vett szó $\mathbf{v}$. Az $\mathbf{e} = \mathbf{v} - \mathbf{c}$ vektort *hibavektornak* nevezzük. Az eddigi tételek alapján könnyen látszik, hogy $\mathbf{H}\mathbf{v}^T = \mathbf{H}(\mathbf{c}+\mathbf{e})^T = \mathbf{H}\mathbf{c}^T + \mathbf{H}\mathbf{e}^T = \mathbf{H}\mathbf{e}^T$, vagyis $\mathbf{H}\mathbf{v}^T$ értéke csak a hibavektortól függ. A szindróma dekódolás az $\mathbf{s}^T = \mathbf{H}\mathbf{v}^T = \mathbf{H}\mathbf{e}^T$ szindróma kiszámítása, és az ennek alapján történő hibavektor becsléseivel történik, amit $\mathbf{v}$-ből levonva megkapjuk a $\mathbf{c}$-re vonatkozó becslésünket.

A dekódolási séma a dekódolási táblától függ, amely az össze lehetséges vehető szavakat sorolja fel. A dekódolási táblát a Lagrange-tétel szerint a mellékosztályokra való felbontás felhasználásával szerkesztjük meg. Mivel a kódszavak az összes kapható szavak halmazának egy részcsoportját alkotják, a standard elrendezési táblázatot a következőképp készíthetjük el:

Minden sort a hozzá tartozó $\mathbf{s}$ szindrómával fogunk kezdeni. Az első sorba kiírjuk az $\mathbf{s}^{(0)}$ szindrómát, majd utána írjuk a C kód szavait, a 0-val kezdve (ugyanis ez azonos az $\mathbf{e} = 0$ hibavektor mellékosztályával):

$\mathbf{s}^{(0)}$	$\mathbf{e}^{(0)} = 0$	$\mathbf{c}^{(1)}$	$\mathbf{c}^{(2)}$	$\dots$	$\mathbf{c}^{(q^k-1)}$

Ezután egy olyan $\mathbf{e}$ hibamintát választunk, ami nem írható fel $\mathbf{e} = \mathbf{e}^{(0)} + \mathbf{c}$ alakban, ahol $\mathbf{e}^{(0)}$ az első sor mellékosztályát generálja, azaz nem tartoznak ugyanazon mellékosztályba. Ilyen $\mathbf{e}$ biztosan létezik, (lást a 2.1.23 lemma) mert ha nem létezne, az első sor, azaz a C kód lefedné az összes lehetséges szót ($C = Q^n$), viszont ez ellentmondana annak, hogy a redundancia érdekében nem minden szó kódszó.

Ezzel az $\mathbf{e}$ hibamintával alkotott mellékosztály elemeit leírjuk az első sor alá olyan sorrendben, hogy az első elem legyen a legkisebb súlyú (ezt nevezzük *mellékosztály-vezetőnek*):

$\mathbf{s}^{(0)}$	$\mathbf{e}^{(0)} = 0$	$\mathbf{c}^{(1)}$	$\mathbf{c}^{(2)}$	$\dots$	$\mathbf{c}^{(q^k-1)}$
$\mathbf{s}^{(1)}$	$\mathbf{e}^{(1)}$	$\mathbf{c}^{(1)} + \mathbf{e}^{(1)}$	$\mathbf{c}^{(2)} + \mathbf{e}^{(1)}$	$\dots$	$\mathbf{c}^{(q^k-1)} + \mathbf{e}^{(1)}$

Ezután válasszunk olyan hibamintát, ami az előző két sorban nem szerepel, és ismételjük ezeket a lépéseket, amíg lehetséges. Így a standard elrendezési táblázat végleges alakja:

$\mathbf{s}^{(0)}$	$\mathbf{e}^{(0)} = 0$	$\mathbf{c}^{(1)}$	$\mathbf{c}^{(2)}$	$\dots$	$\mathbf{c}^{(q^k-1)}$
$\mathbf{s}^{(1)}$	$\mathbf{e}^{(1)}$	$\mathbf{c}^{(1)} + \mathbf{e}^{(1)}$	$\mathbf{c}^{(2)} + \mathbf{e}^{(1)}$	$\dots$	$\mathbf{c}^{(q^k-1)} + \mathbf{e}^{(1)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{s}^{(q^{n-k}-1)}$	$\mathbf{e}^{(q^{n-k}-1)}$	$\mathbf{c}^{(1)} + \mathbf{e}^{(q^{n-k}-1)}$	$\mathbf{c}^{(2)} + \mathbf{e}^{(q^{n-k}-1)}$	$\dots$	$\mathbf{c}^{(q^k-1)} + \mathbf{e}^{(q^{n-k}-1)}$

A Lagrange tétel alapján egyik szó sem fordul elő kétszer a táblázatban. Egy soron belül ez rögtön következik az első sor felírásából és a 2.1.21 lemmából. Különböző sorok esetén tegyük fel, hogy $\mathbf{e}^{(i)} + \mathbf{c}^{(j)} = \mathbf{e}^{(k)} + \mathbf{c}^{(m)}$, ahol $i \neq k$. Mivel ebből $\mathbf{e}^{(i)} = \mathbf{e}^{(k)} + \mathbf{c}^{(m)} - \mathbf{c}^{(j)} = \mathbf{e}^{(k)} + \mathbf{c}^{(n)}$ következik, ahol $\mathbf{c}^{(j)}, \mathbf{c}^{(n)}, \mathbf{c}^{(m)} \in C$, ezért $\mathbf{e}^{(i)}$-nek is az $\mathbf{e}^{(k)}$ mellékosztály-vezetőjű sorban kell lennie, ami ellentmond a kiindulási feltételünknek.

A szindrómák bevezetése, és a standard elrendezési táblázat felírása után következzen maga a hibajavítás:

Az $\mathbf{e}^{(i)}$, $i = 1, 2, \ldots, q^{n-k} - 1$ mellékosztály-vezetőket *javítható hibamintáknak* nevezzük, ugyanis ha a $\mathbf{v}$ fogadott szó szindrómája $\mathbf{s}^{(i)}$, akkor a $\mathbf{c}' = \mathbf{v} - \mathbf{e}^{(i)}$ kódszóra döntünk. A szindróma dekódolásának ezt a módját *táblázatos dekódolásnak* nevezzük.

Nézzünk ezzel a módszerrel egy példát:

3.3.2. Példa. Vegyük a

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

generátormátrix segítségével generálható C kódot. Ekkor C szisztematikus, továbbá $(5, 2)$ bináris kód, tehát $|C| = q^k = 2^2 = 4$ kódszóból áll, és elemei a

$$C = \{(0,0,0,0,0);\ (1,0,1,1,1);\ (0,1,1,0,1);\ (1,1,0,1,0)\}$$

kódszavak. A $\mathbf{G}$ mátrix és a 3.2.14 tétel segítségével számoljuk ki a $\mathbf{H}$ paritásmátrixot:

$$\mathbf{G} = \left[\begin{array}{cc|ccc} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{array}\right] \Rightarrow \left[\begin{array}{ccc|cc} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{array}\right] = \mathbf{H}.$$

Legyen $\mathbf{v} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \end{bmatrix}$, ahol $\mathbf{v}$ egy fogadott kód.

Számoljuk ki az $\mathbf{s}$ szindrómát a következő módon: Első lépésben a szindróma definíciója segítségével meghatározzuk a szindróma transzponáltját, majd az így kapott vektort transzponáljuk:

$$\mathbf{s}^T = (\mathbf{e}\mathbf{H}^T)^T = \mathbf{H}\mathbf{e}^T = \mathbf{H}\mathbf{v}^T = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

azaz $\mathbf{s} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$.

Most a standard elrendezési táblázatot felírjuk a fejezetben leírt módon, majd kikeressük belőle a $\mathbf{c}'$ becsült kódszót. A felírás lépéseit nem részletezem ismét, csak az eredményt adom meg.

3.1. táblázat. Standard elrendezési táblázat

000	00000	11010	01101	10111
100	10000	01010	11101	00111
010	01000	10010	00101	11111
011	00110	11100	01011	10001
001	00100	11110	01001	10011
101	00001	11011	01100	10110
110	01100	10110	00001	11011
111	00010	11000	01111	10101

Azaz az $\mathbf{s} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$ szindrómához tartozó hibaminta: $\mathbf{e} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}$, ennek segítségével pedig $\mathbf{c}' = \mathbf{v} - \mathbf{e} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \end{bmatrix}$. Kikeresve a 3.1 táblázatból a $\mathbf{c}'$ elemet, meghatározhatjuk, melyik kódszó került elküldésre, annak alapján, hogy melyik kódszó szerepel a $\mathbf{c}'$ oszlopának első sorában. Ez a $\mathbf{c} = 11010$ kódszó lesz.

3.4. Hamming-kódok

Az 1940-es években a kódolást még csak hibajelzésre használták, Richard Hamming készítette el az első, máig használt kódokat, amik már egy hiba javítására is alkalmasak voltak, a Hamming-kódot, és a kiterjesztett Hamming-kódot. Ezen eredményeit az 1950-ben kiadott Error Detecting and Error Correcting Codes című cikkében közölte. Ehhez az alfejezethez a [9] és [10] forrásokat használtam fel.

A Hamming-kódok lineáris kódok, és egy Hamming-kódot precízen q *ábécéjű* (n, k) *Hamming-kódnak* nevezünk, ahol q az $\mathbb{F}_q$ alaptest elemszáma. Továbbra is $q = 2$-vel számolunk, ám az eredményeket általános q-ra is kimondjuk majd, ugyanis fontos jelezni, hogy a Hamming-kódok nem csak bináris esetben használhatók.

A hibajavításra r redundáns bitet akarunk felhasználni, (azaz $r = n - k$) és így szeretnénk minél nagyobb k-t elérni, hogy minél több üzenetünk lehessen. Mivel r csak n-től és k-tól függ, a feladatot úgy tudnánk átfogalmazni, hogy fix n esetén keressük a maximális k-t, vagy ami ezzel ekvivalens, hogy fix k esetén keressük a minimális n-t.

Ha n hosszú kódszóban legfeljebb 1 hiba történhet, az azt jelenti, hogy az összes $\mathbf{e}$ hibavektor vagy minden koordinátájában 0, vagy pontosan egy koordinátájában 1, a többiben 0. Ez a 3.3.1 pontban definiáltak miatt akkor teljesül, ha az $\mathbf{s}$ szindróma minden koordinátájában 0, vagy megegyezik $\mathbf{H}$ paritásmátrix valamelyik oszlopával. Mivel rögzített r esetén minél nagyobb k elérése a cél, így mind a $2^r - 1$ lehetőséget használni fogjuk. Az $r = n - k$ miatt itt fennáll az

$$n = 2^{n-k} - 1 \tag{3.3}$$

összefüggés.

3.4.1. Következmény. Az előző egyenlőséget, és r értékét felhasználva:

$$k = n - r = 2^{n-k} - 1 - r = 2^r - 1 - r.$$

Ezzel megvan a kód mérete. A folyamat jobb megértése érdekében nem a paritásmátrix felírása lesz a következő lépés, hanem megvizsgáljuk, hogy az alapokról építkezve hogyan tudjuk alkalmazni ezt a fajta kódolást, míg végül a paritásmátrixot is felírjuk. Most, hogy tudjuk a kódszavak méretét, meg kell mondani, hogy a kódszavakban hol helyezkedjenek el az üzenet- és a paritásszimbólumok, és a paritásszimbólumok milyen értékeket vegyenek fel. Először adjuk meg a paritásszimbólumok értékeit. A paritásmátrix feladata megadni a hiba helyét a kódszóban, így, ha egy pozíció bináris ábrázolásában a jobb szélső bit 1, akkor ennek a hibája esetén az első paritásszimbólum értéke hibás lesz. A különböző számok bináris formája, amik kielégítik ezt a feltételt:

$1 = 1$

$3 = 11$

$5 = 101$

$7 = 111$

$9 = 1001$

Hasonló módszerrel a második paritásszimbólum akkor fog hibázni, ha a bináris ábrázolásban a jobb szélről számolva a második bit 1, azaz a

$2 = 10$

$3 = 11$

$6 = 110$
$7 = 111$
$10 = 1010$
$11 = 1011$
számok esetén. Ez a módszer így folytatódik tovább, amíg le nem fedjük a pozíciókat n-ig.

Ezután azt kell még kiválasztani, hogy hol helyezkedjenek el a paritásszimbólumok. Erre a célra az $1, 2, 4, 8, \ldots$ pozíciókat választjuk, ugyanis így az ellenőrző pozíciók függetlenek lesznek egymástól. A többi helyre kerülnek az üzenetszimbólumok. Az eddigieket a 3.2 tábla szemlélteti.

3.2. táblázat. Paritásbitek, és összefüggéseik

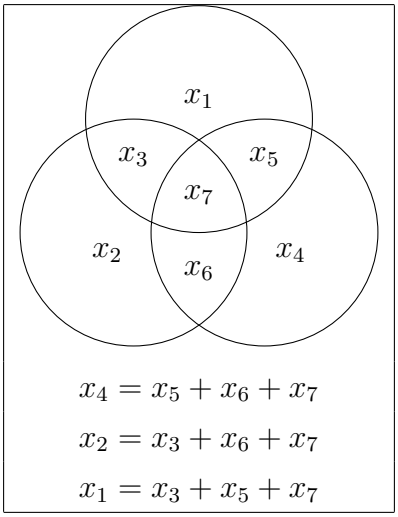
Paritásbit indexe	Paritásbit pozíciója	Paritásbitek által ellenőrzött pozíciók
1	1	$1, 3, 5, 7, 9, 11, 13, 15, 17, \ldots$
2	2	$2, 3, 6, 7, 10, 11, 14, 15, 18, \ldots$
3	4	$4, 5, 6, 7, 12, 13, 14, 15, 20, \ldots$
4	8	$8, 9, 10, 11, 12, 13, 14, 15, 24, \ldots$
$\vdots$	$\vdots$	$\vdots$

3.4.2. Következmény. A Hamming-kódok blokk-kódok.

Az eddigiek alapján nézzünk meg egy konkrét esetet, és vizsgáljuk meg annak a paritásszimbólumait. Legyen a példánkban $r = 3$. Ekkor a vizsgált kód, az $(n, k) = (2^r - 1, 2^r - 1 - r) = (7, 4)$ Hamming-kód a (3.3) egyenlőséget felhasználva.

A paritásbitek értékének meghatározása egyenletekkel vagy halmazok segítségével látható a 3.3 számú ábrán, ahol x_i a szó i. koordinátájának értéke.

3.3. táblázat. A $(7,4)$ Hamming-kód paritásbitjeinek meghatározása



Ekkor a kódszók a $\begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{bmatrix}$ alakot veszik fel. Nézzünk egy gyors példát a paritásbitek kiszámolására.

3.4.3. Példa. Legyen a kódolandó szavunk az $\begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}$, ekkor a kódszó meghatározása paritásbitenként:

$$\begin{array}{lllllll} [x_1 & x_2 & 1 & x_4 & 0 & 1 & 1] \\ [x_1 & x_2 & 1 & 0 & 0 & 1 & 1] \\ [x_1 & 1 & 1 & 0 & 0 & 1 & 1] \\ [\ 0 & 1 & 1 & 0 & 0 & 1 & 1] \end{array}$$

Most, hogy láttuk egy kódszó felépülését, nézzük meg, hogyan lehet velük hibát javítani.

3.4.4. Példa. Legyenek

$$a = x_4 + x_5 + x_6 + x_7$$

$$b = x_2 + x_3 + x_6 + x_7$$

$$c = x_1 + x_3 + x_5 + x_7,$$

és tegyük fel, hogy egy hiba történt az üzenetátvitel közben. Ekkor abc, mint egy szám bináris interpretációja fogja megadni, hogy hányadik koordinátában történt a hiba. Konkrét számokkal számolva, ha $\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$ volt a küldött szó, és $\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$ a fogadott, akkor $a = 1$, $b = 0$, $c = 1$, azaz $abc = 101$, ami az 5 bináris megfeleltetése, tehát az ötödik koordinátában történt a hiba.

A kódok egy általánosabb elkészítéséhez a paritásmátrixokat fogjuk felhasználni. Már láttuk a k maximalizálásakor, hogy, 3.3.1 következtében, $\mathbf{H}$ minden lehetséges $2^r - 1$ oszlopát fel kell használnunk, hogy az összes lehetséges legfeljebb 1 hibát javíthassuk, így a paritásmátrix felírása $r = 3$ esetre:

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{3.4}$$

Egy konkrét Hamming-kód megadása általában a paritásmátrixa felírásával történik.

Ebben az alfejezetben mindig feltételeztük, hogy legfeljebb 1 hiba történt a kódszóban.

3.4.5. Tétel. *Bináris Hamming-kódok kódszói közti minimális távolság* 3.

Bizonyítás. Tegyük fel, hogy $\mathbf{u}, \mathbf{v} \in C$, ahol C egy Hamming-kód, és ellenőrzőmátrixa $\mathbf{H}$. Ekkor C linearitása miatt $\mathbf{u} - \mathbf{v} \in C$ is teljesül. Ha $d_{min} = 1$, akkor $\mathbf{H}(\mathbf{u} - \mathbf{v})^T$ eredménye a $\mathbf{H}$ egy oszlopa. Ekkor $\mathbf{H}$ egyik oszlopa sem 0, viszont, mivel $\mathbf{u} - \mathbf{v}$ kódszó, így $\mathbf{H}(\mathbf{u} - \mathbf{v})^T = 0$, ami ellentmondás. Ha $d_{min} = 2$, akkor $\mathbf{H}(\mathbf{u} - \mathbf{v})^T = 0$ úgy teljesülhet, ha $\mathbf{H}$-nak van 2 lineárisan összefüggő oszlopa, ez viszont ellentmond annak, hogy $\mathbf{H}$ oszlopai bázist alkotnak, tehát $d_{min} \geq 3$ bármely $\mathbf{u}, \mathbf{v} \in C$ kódszóra. Minden bináris Hamming-kód paritásmátrixának van három lineárisan összefüggő oszlopa, azaz beláttuk, hogy d_{min} el is érheti a 3-at. $\square$

3.4.6. Következmény. A 1.3.14 tétel szerint egy kód t-hibajavító $\Longleftrightarrow 2t < d_{min}$. Ez az egyenlőtlenség $d_{min} = 3$ esetén csak $t = 1$-re teljesül. Emiatt lesz a bemutatott Hamming-kód 1 hibajavító.

Most mutassuk meg, hogy az eddigiek nemcsak bináris esetben teljesülnek, a Hamming-kódok egy tetszőleges q prímelemű ábécé esetén is képesek 1 hibát javítani. A $\mathbf{H}$ mátrix elkészítése (ami megadja a Hamming-kódot) hasonlóan történik, mint $q = 2$ esetben. Adott r szám esetén válasszunk egy nem nulla r hosszú vektort $\mathbb{F}_q$ elemeivel. Ezután válasszunk egy újabb r hosszú vektort úgy, hogy az lineárisan független legyen az elsőtől. Legyen ez a $\mathbf{H}$ második oszlopa. Összesen $q^r - 1$ lehetséges nem nulla r hosszú vektor létezik, de figyelni kell, hogy minden vektor $\mathbf{H}$-ba történő beválasztásával további $q - 1$ vektor beválasztásának lehetőségétől esünk el, ugyanis ez a szám a nemnulla többszörösök száma, minden nemnulla vektor esetében. Tehát a paritásmátrix sorainak száma $(q^r - 1)/(q - 1)$.

Az $r = n - k$ alapján definiáltuk a $[(q^r - 1)/(q - 1), (q^r - 1)/(q - 1) - r] = [(q^{n-k} - 1)/(q - 1), k]$, q elemű ábécével rendelkező Hamming-kódokat.

3.4.7. Példa. Egy $[4,2]$-es, $\mathbb{F}_3$ felett értelmezett Hamming-kód paritásmátrixa:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 1 & 1. \end{bmatrix}$$

Fontos megjegyezni, hogy nem ez az egyetlen elkészíthető $[4,2]$-es, $\mathbb{F}_3$ feletti kód, például a

$$\mathbf{H} = \begin{bmatrix} 0 & 2 & 1 & 2 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

egy másik kódot reprezentál.

3.4.8. Tétel. *A Hamming-kódok perfekt,* 1 *hibajavító kódok bármely q prím esetén.*

Bizonyítás. Azt már láttuk az előző tételben, hogy a Hamming-kódok 1 hibajavító kódok. Még azt kell ellenőriznünk, hogy

$$\sum_{i=0}^{t} \binom{n}{i} (q-1)^i \leq q^{n-k}$$

egyenlőtlenségben teljesül-e az egyenlőség. Ehhez alakítsuk át az egyenletet úgy, hogy átszorzunk q^k-val, azaz

$$q^k \cdot \sum_{i=0}^{t} \binom{n}{i} (q-1)^i \leq q^n.$$

Ekkor a bal oldal értéke (ahol $t = 1$ az előzőek miatt)

$$\begin{aligned} q^k(1 + n(q-1)) &= q^k \left(1 + \frac{q^{n-k}-1}{q-1}(q-1)\right) \\ &= q^k(1 + (q^{n-k} - 1)) \\ &= q^k(q^{n-k}) \\ &= q^n. \end{aligned}$$

Vagyis a Hamming-kódok perfekt, 1 hibajavító kódok. □

Végül tegyünk említést a kiterjesztett Hamming-kódokról is, már csak bináris esetben. Visszatérve a korábban bemutatott $[7,4]$ Hamming-kódhoz, vegyünk hozzá egy új, x_0 paritásbitet, hogy

$$x_0 = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7,$$

vagyis a nyolc szám összege 0. Ekkor a kód 8 hosszú, ahol az üzenet továbbra is 4 hosszú. Ezt a kódot $[8,4]$ *kiterjesztett bináris Hamming-kódnak* nevezzük. Ennek a paritásmátrixa megkapható a $[7,4]$ Hamming-kód paritásmátrixa segítségével úgy, hogy a bal oldalhoz hozzáveszünk egy csupa 0 oszlopot, majd a mátrixhoz hozzáveszünk egy csupa 1 sort. A (3.4) paritásmátrixot felhasználva, a $[8,4]$ Hamming-kód paritásmátrixa:

$$\left[\begin{array}{c|ccccccc} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right].$$

4. fejezet

Ciklikus kódok

Ennek a fejezetnek az lesz a célja, hogy bevezessük a bináris ciklikus kódokat, és hatékony módszert adjunk a velük való kódolásra és dekódolásra. A fejezethez a [1], [12], [13] és [14] forrásokat használtam.

4.1. Bináris ciklikus kódok

4.1.1. Definíció. Legyen C lineáris (n, k) blokk-kód. Legyen $\mathbf{u}$ egy üzenet, $\mathbf{c} \in C$ pedig az $\mathbf{u}$-hoz tartozó kódszó. Ekkor minden $\mathbf{c}$-hez hozzárendelhető egy $\mathbf{c}(x)$ *polinom*, úgy, hogy

$$\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}) \Rightarrow \mathbf{c}(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1},$$

ahol az x bitpozíció-operátor segít meghatározni a c_i helyét $\mathbf{c}$-ben $c_i x^i$-ként, ahol $0 \leq i < n$.

4.1.2. Definíció. Egy C lineáris blokk-kód *ciklikus*, ha minden kódszavának lineáris eltoltja is kódszó, azaz ha zárt a kódszavak lineáris eltolására:

$$\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}) \in C \Leftrightarrow \mathbf{c}^{(1)} = (c_{n-1}, c_0, c_1, \ldots, c_{n-2}) \in C.$$

Az $\mathbb{F}^n$ test elemeit az n-nél alacsonyabb fokú polinomokkal azonosítjuk (együtthatósorozatok). Ezek a poliomok a *mod* $(x^n - 1)$ maradékosztály polinomjainak, azaz az $\mathbb{F}[x]/(x^n - 1)$ gyűrű elemeinek tekinthetők.

4.1.3. Lemma. *A polinomok esetében egy ciklikus eltolás az x bitpozíció-operátorral való mod $(x^n - 1)$ szorzást jelenti, tehát*

$$\mathbf{c}(x) \in C \Leftrightarrow \mathbf{c}^{(1)}(x) = x\mathbf{c}(x) \ mod \ (x^n - 1) \in C.$$

Bizonyítás. Legyen $\mathbf{c}(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$, ekkor $x\mathbf{c}(x) = c_{n-1}(x^n - 1) + \mathbf{c}^{(1)}(x)$, ahonnan az állítás következik, mert $deg(\mathbf{c}^{(1)}(x)) \leq n - 1$. $\square$

4.1.4. Definíció. A $\mathbf{c}(x)$ polinomot *kódszópolinomnak*, vagy *kódpolinomnak* nevezzük.

4.1.5. Példa. Legyen $n = 7$. Ekkor a $\mathbf{c} = (0101011)$ vektor ciklikus eltoltja a $\mathbf{c}^{(1)} = (1010101)$. Polinomokkal ábrázolva $\mathbf{c}(x) = x + x^3 + x^5 + x^6$ és

$$\begin{aligned}\mathbf{c}^{(1)}(x) &= x\mathbf{c}(x) = x^2 + x^4 + x^6 + x^7 \ mod\ (x^7 + 1)\\ &= x^2 + x^4 + x^6 + x^7 + (x^7 + 1)\\ &= 1 + x^2 + x^4 + x^6.\end{aligned}$$

4.1.6. Tétel. *A $C \subseteq \mathbb{F}[x]/(x^n - 1)$ lineáris kód ciklikus $\Leftrightarrow$ C ideál az $\mathbb{F}[x]/(x^n - 1)$ gyűrűben.*

Bizonyítás. $\Leftarrow$: C zárt a ($mod\ (x^n - 1)$ vett) szorzásra x-szel, ezért minden t konstansra az x^t-vel való szorzásra is. Mivel C zárt az összeadásra, az előzőkből következik, hogy zárt a polinomokkal való szorzásra is.

$\Rightarrow$: Mivel C lineáris tér, így az elemei csoportot alkotnak az összeadásra. Még kell, hogy tetszőleges $\mathbf{c} \in C$ és $\mathbf{v}(x) \in \mathbb{F}[x]/(x^n - 1)$ esetén $\mathbf{c} \cdot \mathbf{v} \in C$. Ez is teljesül, ugyanis $\mathbf{c}(x) \cdot \mathbf{v}(x)$ a $\mathbf{c}(x)$ ciklikus eltoltjainak lineáris kombinációja. $\square$

4.1.1. Generátor- és paritáspolinomok

4.1.7. Definíció. Minden ciklikus kódhoz egyértelműen tartozik egy $\mathbf{g}(x)$ *generátorpolinom*, aminek a többszörösei a kódszavak. Mivel a $\mathbf{g}(x)$ generátorpolinom osztja $(x^n - 1)$-et, a $\mathbf{g}(x)$ meghatározásához bontsuk fel $(x^n - 1)$-et irreducibilis tagok szorzatára:

$$(x^n - 1) = \phi_1(x) \dots \phi_l(x).$$

Ezek alapján a generátorpolinom előáll

$$\mathbf{g}(x) = \prod_{j \in J \subset \{1,2,\dots,l\}} \phi_j(x)$$

alakban.

4.1.8. Megjegyzés. Mivel bináris esetben kivonás és összeadás közt nem teszünk különbséget, és innentől kezdve bináris esetet feltételezünk, így $(x^n - 1)$ helyett $(x^n + 1)$-et fogunk írni.

4.1.9. Példa. Az $n = 7$ esetben az irreducibilis polinomok szorzatára bontás:

$$x^7 + 1 = (x + 1)(x^3 + x + 1)(x^3 + x^2 + 1)$$

4.1.10. Definíció. A C bináris ciklikus (n, k) kód esetén $k = n - deg[\mathbf{g}(x)]$, ahol C linearitása miatt bármely k lineárisan független vektor előállíthatja a *generátormátrixot*. Erre egy egyszerű választás a $\mathbf{g}(x), x\mathbf{g}(x), \ldots, x^{k-1}\mathbf{g}(x)$ vektorok, ám ez esetben nem szisztematikus a kódolás.

Állítsunk elő egy szisztematikus kódolást. Legyen

$$\mathbf{m} = (m_0, m_1, \ldots, m_{k-1}) \leftrightarrow \mathbf{m}(x) = m_0 + m_1 x + \cdots + m_{k-1}x^{k-1}$$

az üzenetvektor és az üzenetpolinom. Toljuk el jobbra $n - k$ pozícióval:

$$x^{n-k}\mathbf{m}(x) = m_0 x^{n-k} + m_1 x^{n-k+1} + \cdots + m_{k-1}x^{n-1}.$$

Az ennek megfeleltethető vektor a $(0, 0, \ldots, 0, m_0, m_1, \ldots, m_{k-1})$, ahol $n - k$ darab 0-t követnek az $\mathbf{m}$-et alkotó bitek. Ezután osszuk el $x^{n-k}\mathbf{m}(x)$-et a $\mathbf{g}(x)$ generátorpolinommal, a következő módon:

$$x^{n-k}\mathbf{m}(x) = \mathbf{q}(x)\mathbf{g}(x) + \mathbf{d}(x).$$

Így megkaphatjuk a $\mathbf{q}(x)$-el jelölt *hányadost*, és a $\mathbf{d}(x)$-el jelölt *maradékot*, ahol $deg[\mathbf{d}(x)] < n - k$, tehát $\mathbf{d}(x) = [x^{n-k}\mathbf{m}(x)\ mod(\mathbf{g}(x))]$. A $\mathbf{d}(x)$ a fokából következően a következő vektort jelenti: $(d_0, d_1, \ldots, d_{n-k-1}, 0, 0, \ldots, 0)$.

Az eddigiek alapján, a

$$x^{n-k}\mathbf{m}(x) - \mathbf{d}(x) = \mathbf{q}(x)\mathbf{g}(x)$$

egyenlet bal oldalán lévő kifejezés egy kódszó, (ugyanis a $\mathbf{g}(x)$ többszöröse) így a kódszó vektoros alakja:

$$x^{n-k}\mathbf{m}(x) - \mathbf{d}(x) \leftrightarrow (-d_0, -d_1, \ldots, -d_{n-k-1}, m_0, m_1, \ldots, m_{k-1}). \tag{4.1}$$

Az utolsó k szimbólum az üzenetszimbólumok, míg az első $n - k$ szimbólum a paritás-szimbólumok, tehát egy szisztematikus kódolást kaptunk.

4.1.11. Példa. Ebben a példában egy másik módszert mutatunk szisztematikus kódolásra. Legyen C $(7, 4)$ paraméterű Hamming kód $d = 3$ minimális távolsággal. Ekkor

$\mathbf{g}(x) = x^3 + x + 1$, és

$$\begin{aligned} x^3 \ mod(x^3 + x + 1) &= x + 1 = \mathbf{d}_0(x), \\ x^4 \ mod(x^3 + x + 1) &= x^2 + x = \mathbf{d}_1(x), \\ x^5 \ mod(x^3 + x + 1) &= x^2 + x + 1 = \mathbf{d}_2(x), \\ x^6 \ mod(x^3 + x + 1) &= x^2 + 1 = \mathbf{d}_3(x). \end{aligned}$$

Ezek alapján a szisztematikus generátormátrix:

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix},$$

így egy $\mathbf{u}(x)$ üzenet szisztematikus kódolása: $\mathbf{c}(x) = x^{n-k}\mathbf{u}(x) + [x^{n-k}\mathbf{u}(x) \ mod(\mathbf{g}(x))]$.

4.1.12. Definíció. Definiáljuk a $\mathbf{h}(x)$ *paritásellenőrző polinomot* a $\mathbf{g}(x)$ segítségével a következőképp:

$$\mathbf{g}(x)\mathbf{h}(x) = x^n + 1$$

Innen a paritásellenőrző polinom megkapható a

$$\mathbf{h}(x) = (x^n + 1)/\mathbf{g}(x) = h_0 + h_1 x + \cdots + h_k x^k$$

polinomosztás segítségével. Ekkor a $\mathbf{H}$ paritásmátrix a $\mathbf{h}(x)$ megfelelő ciklikus eltoltjaival kapható meg.

4.1.13. Példa. Vegyük a $d = 3$ minimális távolságú, $(7, 4)$ paraméterű Hamming kódot, ahol $\mathbf{g}(x) = x^3 + x + 1$. Ekkor $\mathbf{h}(x) = (x^7 + 1)/(x^3 + x + 1) = x^4 + x^2 + x + 1$, és a ciklikus eltoltak segítségével kapott paritásmátrix:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

4.1.14. Megjegyzés. A 3.2.14 tétel alapján a $\mathbf{H}$ könnyen megadható szisztematikus alakban, ha a $\mathbf{G}$ mátrixot a fent levezetett módszerrel határozzuk meg.

4.1.2. Szindróma

Az előző alfejezetben láttuk, hogy ha $\mathbf{c}$ egy kódszó, akkor a neki megfeleltetett polinom és a paritáspolinom szorzata 0. Ebből következik, hogy

$$\mathbf{c}(x)\mathbf{h}(x) = \mathbf{m}(x)\mathbf{g}(x)\mathbf{h}(x) = \mathbf{m}(x)(x^n + 1) = 0 \ mod(x^n + 1),$$

ahol $\mathbf{m}(x)$ az üzenetpolinom.

4.1.15. Definíció. Egy $\mathbf{u}$ vektorról el tudjuk dönteni, hogy kódszó-e, annak alapján, hogy $\mathbf{u}(x)\mathbf{h}(x) = 0 \ mod(x^n + 1)$ teljesül-e.

Mivel a ciklikus kódok lineáris blokk-kódok, itt is definiálhatjuk a szindróma fogalmát.

4.1.16. Definíció. Adjuk meg az $\mathbf{s}(x)$ szindrómapolinomot a következő alakban:

$$\mathbf{s}(x) = \mathbf{u}(x)\mathbf{h}(x) \ mod(x^n + 1),$$

ahol $\mathbf{s}(x) = 0$ pontosan akkor, ha $\mathbf{u}(x)$ egy kódszó.

Az eddigi definíciók alapján a szindrómával történő hibajavítás ugyanúgy működik, mint az előző fejezetben, ugyanis csak a jelölések változtak, az elmélet ugyanaz maradt.

4.2. Műveletek elvégzése áramkörök segítségével

A ciklikus kódok használatát indokolja, hogy számítógépek gyorsan el tudják végezni velük a kódolási folyamatot. Ebben az alfejezetben olyan áramköröket vizsgálunk, amik polinomosztással hajtják végre azokat a lépéseket, amikről eddig a fejezetben szó volt. Itt csak a bináris esettel foglalkozunk.

4.2.1. Szisztematikus kódolás

Legyen $\mathbf{g}(x) = 1 + g_1x + \cdots + g_{n-k-1}x^{n-k-1} + x^{n-k}$ a generátorpolinom, $\mathbf{m}(x) = m_0 + m_1x + \cdots + m_{k-1}x^{k-1}$ pedig egy üzenetpolinom. Hogy szisztematikus kódolást kaphassunk, először kiszámoljuk az $x^{n-k}\mathbf{m}(x)$ szorzatot. Ezután osztunk $\mathbf{g}(x)$-el, hogy megkaphassuk $\mathbf{d}(x)$ maradékot. Végül a $x^{n-k}\mathbf{m}(x) - \mathbf{d}(x)$ kiszámításával megkapjuk az $\mathbf{m}(x)$ üzenethez tartozó szisztematikus kódszót a (4.1) alapján.

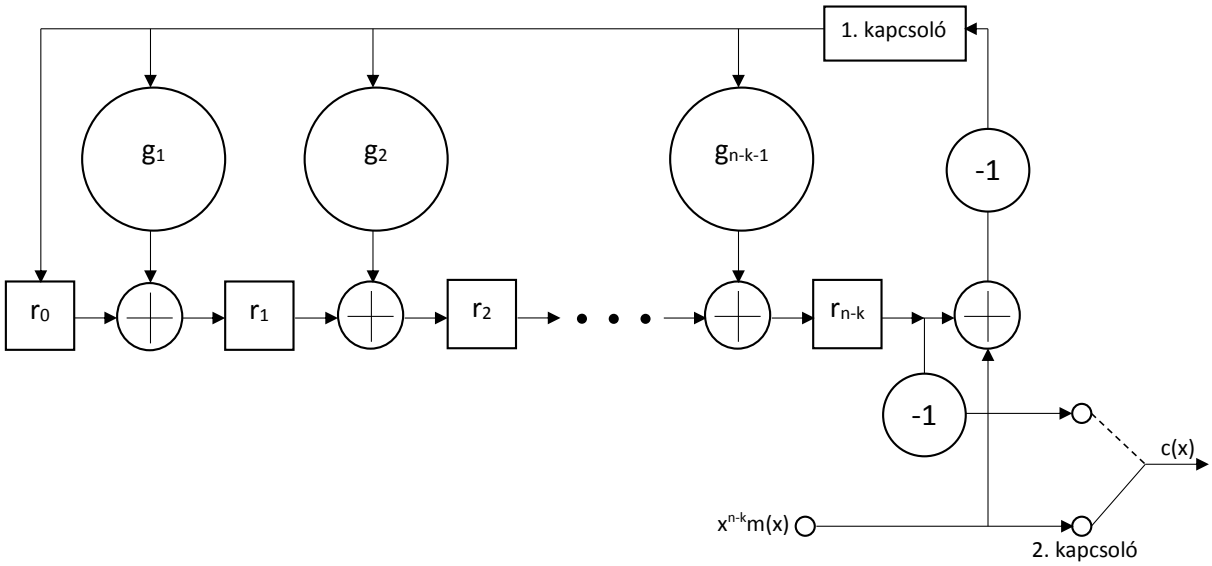
A 4.1 ábrán látható áramkör ezeket a lépéseket hajtja végre. Az ábrán a $\oplus$ a mod 2 összeadást, az $r_0, r_1, \ldots, r_{n-k-1}$ a billenőáramkörökből felépülő regisztereket, a $g_1, \ldots, g_{n-k-1}$ pedig a generátorpolinom együtthatóit jelentik. Az áramkör működése:

1. Az első k lépésben az 1. kapcsoló zárva van, és a 2. kapcsoló az alsó pozícióra mutat, így az $m_{k-1}, \ldots, m_0$ bitek egyszerre lépnek be az áramkörbe, és az outputra.

2. A k-adik üzenetbit elküldése után az 1. kapcsoló kinyílik, és a 2. kapcsoló a felső pozícióra mutat.

3. A rendszer még $n-k$ eltolást hajt végre, hogy a paritásszimbólumokat is eljuttassa az outputra.

4. Összesen n eltolás történt, és az outputon az $x^{n-k}\mathbf{m}(x) - \mathbf{d}(x)$ polinom szerepel.

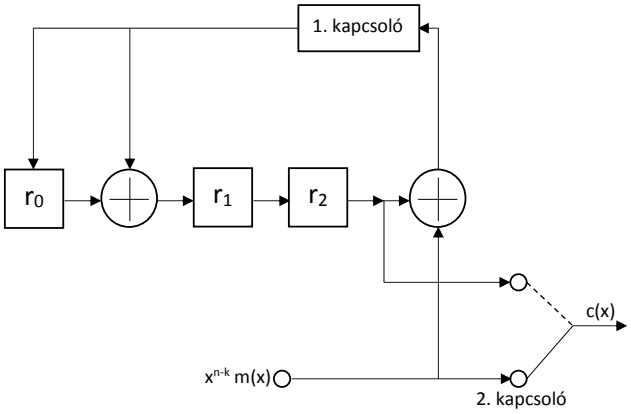


4.1. ábra. Szisztematikus kódoláshoz használt áramkör

4.2.1. Példa. Nézzük a $[7, 4]$ bináris Hamming-kódot generáló $\mathbf{g}(x) = 1 + x + x^3$ polinomot. Az ehhez tartozó áramkör a 4.2 ábrán látható.

4.1. táblázat.

Üzenet	Eltolás előtt			Eltolás után		
m	r_0	r_1	r_2	$r_0 = r_2 + m$	$r_1 = r_0 + r_2 + m$	$r_2 = r_1$
1	0	0	0	1	1	0
1	1	1	0	1	0	1
0	1	0	1	1	0	0
1	1	0	0	1	0	0
–	1	0	0	–	–	–



4.2. ábra. Az $\mathbf{g}(x) = 1 + x + x^3$-hoz tartozó áramkör

Az $\mathbf{m} = (1, 0, 1, 1)$ üzenetvektor áramkörrel vett kódolásának lépéseit a 4.1 táblázat mutatja.

Tehát az outputon lévő kódszó a következő:

$$\mathbf{c}(x) = (1, 0, 0, 1, 0, 1, 1)$$

4.2.2. Dekódolás szindrómával

4.2.2. Tétel. *Legyen* $\mathbf{s}(x)$ *az* $\mathbf{u}(x)$*-hez tartozó szindróma, vagyis* $\mathbf{u}(x) = \mathbf{q}(x)\mathbf{g}(x) + \mathbf{s}(x)$*. Legyen* $\mathbf{u}^{(1)}(x)$ *az* $\mathbf{u}(x)$ *jobb oldali ciklikus eltoltja, és a hozzá tartozó szindróma* $\mathbf{s}^{(1)}(x)$*. Ekkor* $\mathbf{u}(x)$ $mod(x^n - 1)$ *ciklikus eltoltjainak szindrómái az* $\mathbf{s}(x)$ $mod(g(x))$ *ciklikus eltoltjai.*

Bizonyítás. Ha $\mathbf{u}(x) = u_0 + u_1 x + \cdots + u_{n-1}x^{n-1}$, akkor ciklikus eltoltja $\mathbf{u}^{(1)}(x) = u_{n-1} + u_0 x + \cdots + u_{n-2}x^{n-1}$, ami felírható a következőképp:

$$\mathbf{u}^{(1)}(x) = x\mathbf{u}(x) - u_{n-1}(x^n - 1).$$

Felhasználva a maradékos osztást, és azt, hogy $x^n - 1 = \mathbf{g}(x)\mathbf{h}(x)$, azt kapjuk, hogy

$$\mathbf{q}^{(1)}(x)\mathbf{g}(x) + \mathbf{s}^{(1)}(x) = x[\mathbf{q}(x)\mathbf{g}(x) + \mathbf{s}(x)] - u_{n-1}\mathbf{g}(x)\mathbf{h}(x),$$

ahol $\mathbf{s}^{(1)}(x)$ az $\mathbf{u}^{(1)}(x)$-nek a $\mathbf{g}(x)$-szel vett hányadosának maradéka. Az előző egyenlőséget átrendezve:

$$x\mathbf{s}(x) = [\mathbf{q}^{(1)}(x) + u_{n-1}\mathbf{h}(x) - x\mathbf{q}(x)]\mathbf{g}(x) + \mathbf{s}^{(1)}(x).$$

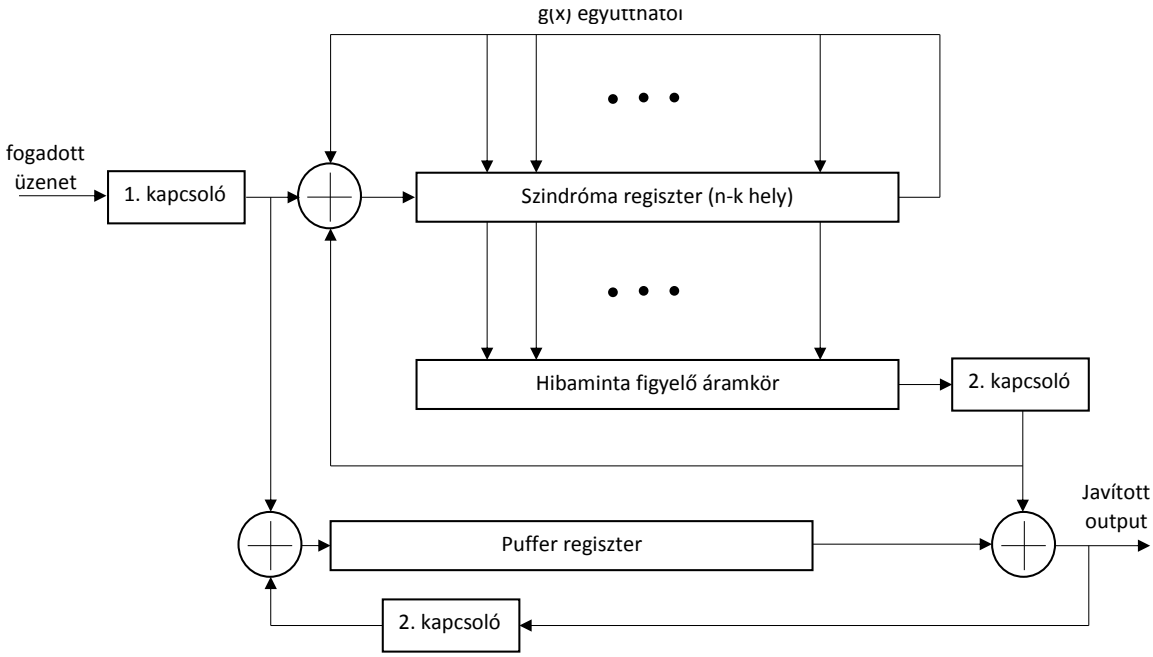
Tehát $\mathbf{s}^{(1)}(x)$ az $x\mathbf{s}(x)/\mathbf{g}(x)$ hányados maradéka. $\square$

Ezek alapján csak egy $\mathbf{s}$ szindrómát kell kiszámolnunk egy $\mathbf{e}$ hibamintához, és annak ciklikus eltoltjaihoz, ami miatt a szindróma tábla n sorral rövidebb lehet. Továbbá a szükséges eltolásokat ki tudjuk számolni azzal az áramkörrel, amivel az $\mathbf{s}$ szindrómát meghatároztuk. Így készítsük el a következő áramkört, ami a hibamintához tartozó szindróma kiszámításának segítségével képes hibajavításra. Ennek a neve *Meggitt-dekódoló.*

Az 4.3 ábrán látható áramkör működésének lépései:

- Az 1. kapcsoló zárva van, a 2. kapcsoló nyitva, a szindróma regiszterek pedig mind 0-ra vannak állítva. A fogadott kódszó az első n eltolás alatt egyszerre belép a puffer regiszterbe és a szindróma regiszterbe. Az n-edik lépés után, a szindróma regiszter tartalmazza az $\mathbf{u}(x)$ fogadott kódszóhoz tartozó $\mathbf{s}(x)$ szindrómát.

- Az 1. kapcsoló nyitva van, a 2. kapcsoló pedig zárva. A hibaminta figyelő áramkör $e_{n-1} = 1$-et ad vissza, ha hibát észlel a fogadott kódszó legmagasabb bitpozíciójában a szindróma segítségével, tehát $\mathbf{e}(x) = x^{n-1}$. Ha 0 a visszatérő érték, akkor a legmagasabb bitpozícióban nincs hiba. Az ezáltal módosított fogadott szó polinomalakban: $\mathbf{u}_1(x) = u_0 + u_1 x + \cdots + u_{n-2}x^{n-2} + (u_{n-1} + e_{n-1})x^{n-1}$. Az $\mathbf{u}_1(x)$ ciklikus eltoltja $\mathbf{u}_1^{(1)}(x) = (u_{n-1} + e_{n-1}) + u_0 x + \cdots + u_{n-2}x^{n-1}$. Az $\mathbf{u}_1^{(1)}(x)$-hez tartozó $\mathbf{s}_1^{(1)}(x)$ szindróma az $\mathbf{u}_1^{(1)}(x)/\mathbf{g}(x)$ hányados maradéka. Mivel az $x\mathbf{u}(x)$ maradéka az $\mathbf{s}^{(1)}(x)$ szindróma, és az xx^{n-1} maradéka 1, így a módosított fogadott szóhoz tartozó szindróma előáll $\mathbf{s}_1^{(1)}(x) = \mathbf{s}^{(1)}(x) + 1$ alakban. Így a szindróma regiszter módosítható, hogy jelezze az $\mathbf{u}(x)$ módosításait úgy, hogy a regiszter bal oldalán hozzáadunk egy 1-est. A módosított érték kikerül az outputra, és újra belép a rendszerbe a 2. kapcsolón keresztül.



4.3. ábra. Meggitt-dekódoló

- Az előző lépés ismétlődik addig, amíg a puffer regiszter összes eleme kikerül az outputra. A folyamat végén a puffer regiszter tartalmazza a javított szót, azaz az eredetileg elküldött kódszót.

4.2.3. Példa. Vizsgáljuk ismét a $\mathbf{g}(x) = 1 + x + x^3$ polinommal generált kódot. A 4.2 táblázat tartalmazza a javítható hibákat, a hozzájuk tartozó szindrómákat, és ezek polinomokban felírt alakját.

Ebből a táblázatból le tudjuk olvasni, hogy az utolsó bithelyen lévő hibához, az $\mathbf{e}(x) = x^6$-hoz tartozó szindróma a $\mathbf{s}(x) = 1 + x^2$. A példához tartozó áramkörben, ami a 4.4 ábrán látható, emiatt egy 3-inputos és kapu, ahol a középső input megfordul fogja végezni a hibajavítást. Amikor a hibaminta figyelő áramkör felismeri a fent említett hibamintát, a puffer regiszterből kilépő bitnek, és a szindróma regiszterbe belépő bitnek a kiegészítettje jut el.

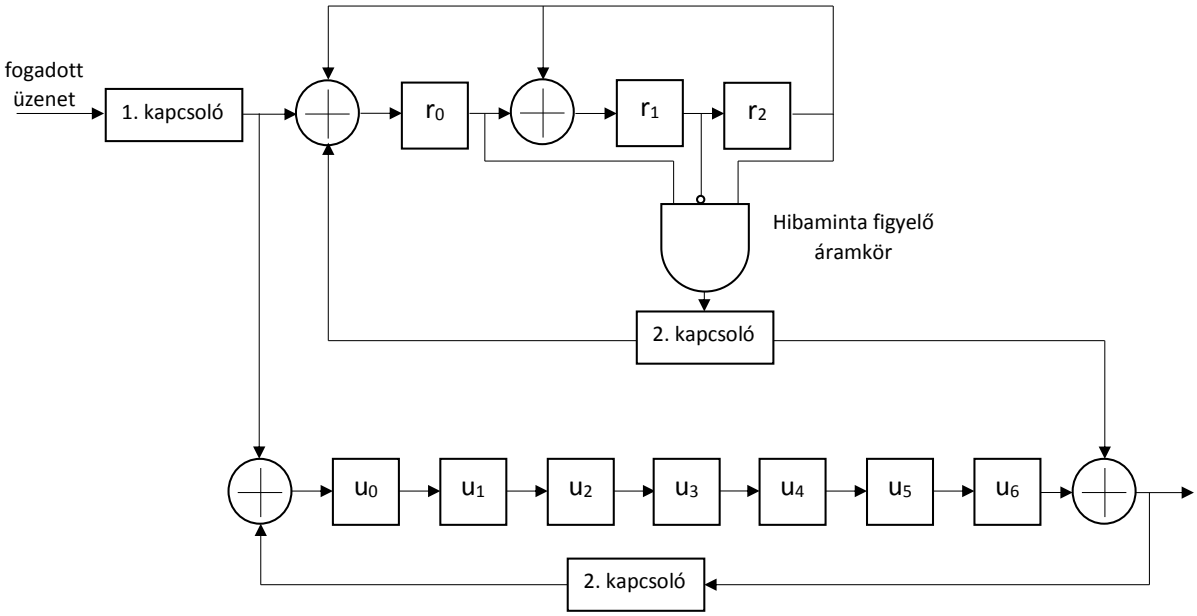
Tegyük fel, hogy a fogadott szó az $\mathbf{u}(x) = 1 + x + x^3 + x^5 + x^6$ polinommal írható le. Ahogyan ez belép az áramkörbe, az $\mathbf{s}(x) = x$ jelenik meg a szindróma regiszterben. A 12. eltolásnál megjelenik az $\mathbf{s}^{(5)}(x) = 1 + x^2$ polinom a szindróma regiszterben, ami azt jelenti, hogy hiba van a puffer regiszter jobb oldali bitjében. A bithez hozzáadjuk a komplementerét, ezzel javítva a hibát. A következő szindróma a 0 polinom lesz, ami

4.2. táblázat.

Hiba	Hibapolinom	Szindróma	Szindrómapolinom
0000000	$\mathbf{e}(x) = 0$	000	$\mathbf{s}(x) = 0$
1000000	$\mathbf{e}(x) = 1$	100	$\mathbf{s}(x) = 1$
0100000	$\mathbf{e}(x) = x$	010	$\mathbf{s}(x) = x$
0010000	$\mathbf{e}(x) = x^2$	001	$\mathbf{s}(x) = x^2$
0001000	$\mathbf{e}(x) = x^3$	110	$\mathbf{s}(x) = 1 + x$
0000100	$\mathbf{e}(x) = x^4$	011	$\mathbf{s}(x) = x + x^2$
0000010	$\mathbf{e}(x) = x^5$	111	$\mathbf{s}(x) = 1 + x + x^2$
0000001	$\mathbf{e}(x) = x^6$	101	$\mathbf{s}(x) = 1 + x^2$

azt jelenti, nincs több hiba a szóban. A javított szó, ami az eredetileg elküldött kódszó: $\mathbf{c}(x) = 1 + x^3 + x^5 + x^6$. Ehhez a kódszóhoz tartozó üzenetpolinom az (4.1) alapján a $\mathbf{c}$ utolsó 4 bitjéből áll elő: $\mathbf{m}(x) = 1 + x^2 + x^3$, tehát $\mathbf{m} = (1011)$.

A 4.4 ábrán látható Meggit-dekódoló műveletei lépésenként lebontva az 4.3 táblázatban láthatók. A hibás bitet aláhúzás jelöli.



4.4. ábra. Egy [7, 4] Hamming-kód dekódoló áramköre

4.3. táblázat.

Lépések	Input	Szindróma regiszter	Puffer regiszter
1.	1	100	1000000
2.	1	110	1100000
3.	0	011	0110000
4.	1	011	1011000
5.	0	111	0101100
6.	$\underline{1}$	001	$\underline{1}$010110
7.	1	010	1$\underline{1}$01011
8.		001	11$\underline{1}$0101
9.		110	111$\underline{1}$010
10.		011	0111$\underline{1}$01
11.		111	101111$\underline{0}$
12.		101	010111$\underline{1}$
13.		000	0010111
14.		000	1001011

Irodalomjegyzék

[1] TODD K. MOON, *Error Correction Coding Mathematical Methods and Algorithms*, A JOHN WILEY & SONS, INC., PUBLICATION, 2005 (5 - 9, 62 - 80, 113 - 154)

[2] M. PURSER, *Introduction to Error Correcting-Codes*, ARTECH HOUSE, INC., 1995 (95 - 97)

[3] JIRI ADAMEK, *Foundation of coding*, A JOHN WILEY & SONS, INC., PUBLICATION, 1991 (25 - 31, 39 - 58, 91 - 114)

[4] KISS EMIL, *Bevezetés az algebrába*, Typotex, 2007 (40 - 45, 141 - 336)

[5] MARK WILDON, *ERROR CORRECTING CODES MT361/MT461/MT5461*, 2012 (1 - 20) (http://www.ma.rhul.ac.uk/ uvah099/Maths/Codes11/MT3612011Notes.pdf)

[6] SAN LING and CHAOPING XING, *Choding theory a first course*, Cambridge University Press, 2004 (17 - 36)

[7] GYÖRFI L., GYŐRI S., VAJDA I, *Információ és kódelmélet*, Typotex Kiadó, 2000 (186 - 197)

[8] G. BIRKHOFF, T. C. BARTEE, *A modern algebra a számítógéptudományban*, Műszaki könyvkiadó, 1974 (196 - 215)

[9] R. W. HAMMING, *Error Detecting and Error Correcting Codes*, The Bell System Technical Journal Vol. XXIX, 1950

[10] JAMES FIEDLER *Hamming Codes*, 2004,

(https://orion.math.iastate.edu/linglong/Math690F04/HammingCodes.pdf)

[11] DEREK J. S. ROBINSON, *An Introduction to Abstract Algebra*, De Gruyter, 2015 (254 - 266)

[12] ROBERT H. MORELOS-ZARAGOZA, *The Art of Error Correcting Coding*, A JOHN WILEY & SONS, INC., PUBLICATION, 2002 (33 - 57)

[13] CHRIS SCHMITT, DONNY HUBENER & LAMIN DUMBUYA *Cyclic Codes*, Kansas University, 2000

[14] YUNGHSIANG S. HAN *Cyclic Codes*, Graduate Institute of Communication Engineering, National Taipei University Taiwan

A. függelék

A 4.2 alfejezetben lévő példa számítógépes kiszámítása

A fejezetben bemutatott script MATLAB-ban íródott.

```
% Adjunk meg egy generátorpolinomot, például: g(x)=1+x+x^3
g = [1 1 0 1];
% Adjunk meg egy üzenetpolinomot, például:  m(x)=1+x^2+x^3
m = [1 0 1 1];

n = length(g)+length(m)-1;
k = length(m);

% m(x)*x^{m-k}
c = [zeros(1,n-k) m];
% kvóciens és maradék meghatározása: (x^6+x^5+x^3)/(x^3+x+1)
[~,d] = deconv(c(end:-1:1),g(end:-1:1));
% bináris esetet vizsgálunk
d = mod(d,2);

% kódszó elkészítése
for j = 1:(n-k)
    c(j) = d(end+1-j);
end
```

```
% 1 hiba hozzáadása
e = zeros(1,n);
e(ceil(rand(1)*n)) = 1;
u = mod(c+e,2);

% és-kapu elkészítése:
[~,and_g] = deconv([1 0 0 0 0 0 0],g(end:-1:1));
and_g = mod(and_g(end:-1:k+1),2);

% Meggit:
% Az 1. kapcsoló zárva, a 2. kapcsoló nyitva
puffer = [];
s = zeros(1,n-k);
% Egy ciklus egy órajelet szimulál
for j = 1:n
    s_tmp = s;
    s(1) = mod(u(end+1-j) + s_tmp(end),2);
    for z = 2:n-k
        if g(z) ~= 0
            s(z) = mod(s_tmp(z-1) + s_tmp(end),2);
        else
            s(z) = s_tmp(z-1);
        end
    end
    puffer = [u(end+1-j) puffer];
end

% Az 1. kapcsoló kinyílik, a 2. kapcsoló bezárul
idx = 0;
for j = 1:n
    s_tmp = s;
    if s_tmp == and_g
        s(1) = mod(s_tmp(end)+1,2);
```

```
    else
        s(1) = s_tmp(end);
    end

    for z = 2:n-k
        if g(z) ~= 0
            s(z) = mod(s_tmp(z-1) + s_tmp(end),2);
        else
            s(z) = s_tmp(z-1);
        end
    end

    if s_tmp == and_g
        puffer = [mod(puffer(end)+1,2), puffer(1:end-1)];
    else
        puffer = [puffer(end), puffer(1:end-1)];
    end
end

m_rec = puffer(k:end);
```