Szegedi Tudományegyetem

Bolyai Intézet

Geometria Tanszék

Tranzitív bináris kódok

BSc szakdolgozat

Készítette:
Mezőfi Dávid Csaba
matematika szakos
hallgató

Témavezető:
Dr. Nagy Gábor Péter
egyetemi docens

Szeged

2014

Tartalomjegyzék

Bevezetés **3**

1. Csoportelmélet **4**
1.1. Orbit és stabilizátor 5
1.2. Csoporthatás 6

2. Kódelmélet **8**
2.1. Kódok 10
2.2. Lineáris kódok 11

3. Tranzitív bináris kódok **13**
3.1. A problémák 14
3.2. Az algoritmusok 14

Függelék **20**

Irodalomjegyzék **23**

Nyilatkozat **24**

Bevezetés

A dolgozat a kódelmélet témaköréhez kapcsolódik. A kódelmélet a következő kommunikációs modellel foglalkozik. Egy *küldő* valamilyen csatornán adatokat akar továbbítani egy *fogadónak*. A továbbítás során az adatok egy része esetleg megváltozik. Hogyan tudja a fogadó kiválasztani, hogy mely adatok változtak meg, és hogyan tudja a megváltozott adatokat kijavítani?

A küldő által továbbított **m**-mel jelölt üzenetről a modellünkben feltesszük, hogy az $\mathbb{F}_2^n$ vektortér egy eleme. Csak az olyan hibák javításával foglalkozunk, melyek a szimbólumok megváltozásából adódnak, nem foglalkozunk szimbólumok hozzáadódásából és elhagyásából keletkező hibákkal. A probléma tehát a következő: a továbbítás során **m**-hez hozzáadódik egy **e** hibavektor, s így a fogadó az $\mathbf{x} = \mathbf{m} + \mathbf{e}$ módosult üzenetet kapja. Célunk az **x** dekódolása, azaz a hibavektor, majd pedig **m** meghatározása.

A dolgozat első szakasza a tranzitív bináris kódok tárgyalásához szükséges csoportelméleti ismereteket tartalmazza. A csoporthatás, a pálya (orbit) fogalmának és tulajdonságainak ismerete elengedhetetlen a precíz, pontos tárgyalásmódhoz. A második szakasz a kódelmélet alapvető fogalmai, tételei – minimum távolság, hibajavító képesség, linearitás – közül azokat foglalja magában, amelyek a harmadik szakaszban tárgyaltak maradéktalan megértéséhez szükségesek. A harmadik szakasz három problémára keres megoldást: adott paraméterekkel rendelkező tranzitív bináris kódokat keresünk; az előzőben foglalt feltételek bizonyos mértékű feloldása mellett keresünk megfelelő bináris kódokat; illetve ezek közül csak lineáris kódokat szeretnénk meghatározni. E szakasz mindegyik problémára megoldást nyújt egy-egy algoritmus formájában. A függelék pedig ezen algoritmusok egy használható implementációját mutatja be a GAP [GAP] nyelvén.

1. Csoportelmélet

Az ebben a szakaszban ismertetett fogalmak, állítások és tételek Kiss Emil *Bevezetés az absztrakt algebrába* [BAA] című könyvéből származnak az ott bevezetett jelölésrendszert követve.

1.1. Definíció. Egy X halmazt önmagára képező kölcsönösen egyértelmű leképezéseket (azaz bijekciókat) az X *transzformációinak* nevezzük, ezek halmazát S_X-szel jelöljük.

Megjegyzés. S_X csoportot alkot a kompozíció műveletére nézve.

1.2. Definíció. Legyen X tetszőleges halmaz. Az S_X csoportot a kompozíció műveletére nézve az X halmazon ható *szimmetrikus csoportnak* hívjuk. Az S_X egységelemét id_X, vagy egyszerűen csak id (sőt néha 1) jelöli, ez az identikus leképezés, amely minden elemet önmagába visz. Az $f \in S_X$ inverzét f^{-1}-gyel jelöljük. A kompozíció műveletét szorzásnak hívjuk majd, és gyakran $f \circ g$ helyett fg-t írunk.

1.3. Definíció. Az X véges halmazt önmagára képező bijekciókat az X halmaz permutációinak nevezzük. Ha $|X| = n$, akkor az S_X csoportot a kompozíció műveletére nézve *n-edfokú* szimmetrikus csoportnak nevezzük. Ha $X = \{1, 2, \ldots, n\}$, akkor S_X helyett S_n-et írunk.

1.4. Definíció. Legyen G csoport a $*$ műveletre, és H csoport a $\star$ műveletre. Azt mondjuk, hogy egy $\psi\colon G \to H$ leképezés *csoport-homomorfizmus*, ha művelettartó, vagyis ha tetszőleges $a, b \in G$ esetén

$$\psi(a * b) = \psi(a) \star \psi(b).$$

Ha ψ kölcsönösen egyértelmű is G és H halmazok között, akkor ψ *izomorfizmus*.

1.5. Definíció. Egy G csoportot (vagy más struktúrát) önmagába képező homomorfizmust *endomorfizmusnak* nevezünk. Ha ez kölcsönösen egyértelmű is G-ből G-re, akkor a neve *automorfizmus*. Egy G csoport automorfizmusainak csoportját (a kompozíció műveletére) G *automorfizmuscsoportjának* hívjuk, és $\mathrm{Aut}(G)$-vel jelöljük.

1.6. Definíció. Ha X és Y tetszőleges részhalmazai a G csoportnak, akkor

$$XY = \{xy\colon x \in X,\ y \in Y\}$$

az X és Y *komplexusszorzata*. Ha speciálisan $X = \{a\}$ egy egyelemű halmaz, akkor $\{a\}\,Y$ és $Y\,\{a\}$ helyett egyszerűen aY-t, illetve Ya-t írunk.

1.7. Definíció. Legyen G csoport, $H \leq G$ és $g \in G$. A gH halmazt (a H részcsoport szerinti) *bal oldali mellékosztálynak* nevezzük. Ugyanígy a Hg halmaz *jobb oldali mellékosztály*.

1.1. Orbit és stabilizátor

1.8. Definíció. Legyen X halmaz. Az S_X szimmetrikus csoport részcsoportjait *transzformációcsoportoknak*, illetve (elsősorban véges X) esetén *permutációcsoportoknak* nevezzük. Az X halmaz elemeit néha pontoknak hívjuk.

1.9. Definíció. Legyen G transzformációcsoport az X halmazon. Ha $x \in X$, akkor tekintsük azokat a $g \in G$ elemeket, melyek x-et *fixen hagyják*, azaz $g(x) = x$. Ezek nyilván részcsoportot alkotnak G-ben, melynek neve az x pont G-beli *stabilizátora*, jele G_x Ha $g(x) = x$, akkor mondjuk azt is, hogy *x fixpontja g-nek.* Egy transzformáció *fixpontmentes*, ha nincs fixpontja.

1.10. Definíció. A $G \leq S_X$ transzformációcsoport *tranzitív*, ha X bármely x és y eleméhez létezik olyan $g \in G$, hogy $g(x) = y$.

1.11. Definíció. Legyen $G \leq S_X$ transzformációcsoport és $x \in X$. A $g(x)$ alakú pontok halmazát, ahol g befutja G elemeit, az x *orbitjának* (pályájának) nevezzük, és $G(x)$-szel jelöljük. Az orbit elemszámát az orbit *hosszának* hívjuk.

1.12. Állítás. *Legyen $G \leq S_X$ transzformációcsoport. Ekkor G összes orbitja az X halmaz egy partícióját alkotják.*

Megjegyzés. Az állítást természetesen úgy kell érteni, hogy ha X két elemének az orbitja ugyanaz, akkor ezt az orbitot csak egyszer vesszük be a partíciót alkotó halmazok közé.

Bizonyítás. Definiáljunk egy $\sim_G$ relációt X-en a következőképpen: $x \sim_G y$ akkor és csak akkor, ha van olyan $g \in G$, melyre $g(x) = y$. Tehát két elem akkor áll relációban, ha az egyiket a másikba át lehet vinni G egy elemével. Azonnal látszik, hogy $\sim_G$ ekvivalenciareláció. Valóban, $x \sim_G x$ (hiszen az egységelem x-et önmagába viszi); ha $x \sim_G y$, akkor $y \sim_G x$ (hiszen ha $g(x) = y$, akkor $g^{-1}(y) = x$); végül ha $x \sim_G y$ és $y \sim_G z$, akkor $x \sim_G z$ (hiszen $g(x) = y$ és $h(y) = z$, akkor $(hg)(x) = z$).

Így $\sim_G$ ekvivalenciareláció lévén egy partíciót definiál. Ennél az $x \in X$ elem osztálya nyilvánvalóan pontosan az x elem orbitja lesz. □

1.13. Tétel. *Legyen $G \leq S_X$. Ekkor tetszőleges $x \in X$-re $|G(x)| = |G : G_x|$. Tehát egy pont orbitjának a hossza épp a pont stabilizátorának az indexe, vagy másképp fogalmazva az x orbitjának elemszáma szorozva az x stabilizátorának elemszámával mindig G elemeinek számát adja.*

Bizonyítás. Azt kell belátni, hogy $G(x)$ elemszáma ugyanaz, mint G_x indexe, vagyis a szerinte vett bal oldali mellékosztályok száma. Kölcsönösen egyértelmű megfeleltetést fogunk létesíteni az $G(x)$ elemei és a G_x szerinti bal oldali mellékosztályok között, melynél az $y \in G(x)$ ponthoz azoknak a G-beli elemeknek a halmaza tartozik, amelyek x-et y-ba viszik. Ez a halmaz tényleg G_x szerinti bal oldali mellékosztály. Ha $y_1 \neq y_2$, akkor a hozzájuk tartozó mellékosztályok különbözők, mert ha g közös elemük lenne, akkor $y_1 = g(x) = y_2$ teljesülne. Minden mellékosztályt meg is kapunk ilyen alakban, hiszen ha $g \in G$, akkor $y = g(x) \in G(x)$, és így az y ponthoz gG_x tartozik. □

1.2. Csoporthatás

1.14. Definíció. Azt mondjuk, hogy a G csoport az X halmazon *hat*, ha minden $g \in G$ és $x \in X$ esetén értelmezve van $g * x \in X$ elem úgy, hogy bármely $g, h \in G$ és $x \in X$ esetén

$$g * (h * x) = (gh) * x$$

(azaz G eleminek szorzata kompozícióként hat), és ha 1 jelöli G egységelemét, akkor bármely $x \in X$ estén

$$1 * x = x$$

(vagyis az egységelem identikusan hat X-en).

1.15. Példa. Legyen $\sigma \in S_n$ tetszőleges permutáció, ekkor az $a \in \{1, 2, \ldots, n\}$ elem σ melletti képe $\sigma(a)$. Tetszőleges $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$-re legyen

$$\sigma * \mathbf{x} = \left(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)}\right) \tag{1}$$

(azaz σ szerint permutáljuk a vektor koordinátáit). Ekkor tetszőleges $\sigma, \tau \in S_n$ és $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$ esetén

$$\begin{aligned}\sigma * (\tau * \mathbf{x}) = \sigma * \left(x_{\tau(1)}, x_{\tau(2)}, \ldots, x_{\tau(n)}\right) = \left(x_{\sigma\tau(1)}, x_{\sigma\tau(2)}, \ldots, x_{\sigma\tau(n)}\right) = \\ = (\sigma\tau) * \mathbf{x},\end{aligned}$$

és

$$\mathrm{id} * \mathbf{x} = \left(x_{\mathrm{id}(1)}, x_{\mathrm{id}(2)}, \ldots, x_{\mathrm{id}(n)}\right) = (x_1, x_2, \ldots, x_n) = \mathbf{x},$$

azaz az 1.14-es definíció szerint S_n hat $\mathbb{F}_2^n$-en.

1.16. Definíció. Legyen G az X-en ható csoport. Ekkor az $x \in X$ pont *orbitja* a $g * x$ alakú pontokból áll $(g \in G)$, vagyis azokból, ahová x-et G elemeivel el lehet vinni, jele $G(x)$. Az $x \in X$ *stabilizátora* azokból a $g \in G$ csoportelemekből áll, melyekre $g * x = x$, vagyis amelyek az x-et fixen hagyják, jele G_x. Azt mondjuk, hogy G hatása X-en *tranzitív*, ha csak egyetlen orbit van, azaz ha X bármely két eleme egymásba átvihető G egy elemével.

1.17. Tétel. *Legyen G az X halmazon ható csoport. Ekkor G orbitjai X egy partícióját adják. Tetszőleges $x \in X$ pont orbitjának a hossza épp a pont stabilizátorának az indexe.*

Bizonyítás. Értelmezzük a $\sim_G$ ekvivalenciarelációt az X-en a következőképpen. $x \sim_G y$ akkor és csak akkor, ha van olyan $g \in G$, melyre $g * x = y$. Ennek osztályai nyilván az orbitok lesznek. A második állítás bizonyítása teljesen ugyanaz, mint az 1.13-as tételé: a G_x stabilizátor szerinti bal mellékosztályok elemei x-et ugyanoda, a különböző mellékosztályok elemei pedig különböző helyre viszik, hiszen

$$g' * x = g * x \iff \left(g^{-1}g'\right) * x = x \iff g^{-1}g' \in G_x \iff g' \in gG_x.$$

Ezért az orbitnak ugyanannyi eleme van, mint ahány mellékosztály. □

1.18. Definíció. Legyen G az X halmazon ható permutációcsoport. Azt mondjuk, hogy ez *k-szorosan tranzitív* ($k \geq 1$ egész), ha bármely páronként különböző $x_1, \ldots, x_k \in X$ és szintén páronként különböző $y_1, \ldots, y_k \in X$ pontokhoz van olyan $g \in G$, hogy $g * x_i = y_i$ minden $1 \leq i \leq k$ esetén.

1.19. Állítás. *Az n-edfokú alternáló csoport (A_n) $n \geq 3$ esetén $(n-2)$-tranzitív az $\{1, 2, , \ldots, n\}$ halmazon.*

Bizonyítás. Ha adottak a páronként különböző $x_1, \ldots, x_{n-2}$ és az ugyancsak páronként különböző $y_1, \ldots, y_{n-2}$ pontok, akkor jelölje a két kimaradó pontot x_{n-1} és x_n, illetve y_{n-1} és y_n. Az S_n-ben két olyan permutáció van, amely $i \leq n-2$ esetén mindegyik x_i pontot pontosan y_i-be viszi. Az egyiknél x_{n-1} az y_{n-1}-be megy, ezt jelölje f. A másiknál x_{n-1} az y_n-be megy, és ezt jelölje g. Nyilván $f = g \circ (x_{n-1}, x_n)$. Ezért f és g közül pontosan az egyik lesz páros permutáció.

□

2. Kódelmélet

Az ebben a szakaszban ismertetett fogalmak, állítások és tételek Petteri Kaski – Patric R. J. Östergard *Classification Algorithms for Codes and Designs* [CACD], valamint Kiss György – Szőnyi Tamás *Véges geometriák* [VG] című könyvéből származnak.

Legyen $\mathbf{x}$ az $\mathbb{F}_2^n$ vektortér egy tetszőleges eleme. Kódelméleti kontextusban az $\mathbf{x}$ vektort *szónak* nevezzük. Az $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ komponenseit *koordinátáknak* nevezzük, míg az $x_i \in \mathbb{F}_2$ értékeket $(i \in \{1, 2, \ldots, n\})$ *koordinátaértékeknek* nevezzük. Ha nem okoz félreértést, akkor az $\mathbf{x}$ szót gyakran csak $x_1 x_2 \cdots x_n$-nel jelöljük.

Célunk, hogy az $\mathbb{F}_2^n$ vektorteret metrizálhassuk – a lehető legegyszerűbb módon. Ezt a következőképpen tehetjük meg.

2.1. Lemma. *A* $d\colon \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{N}_0$ *leképezés, melyre*

$$d(\mathbf{x}, \mathbf{y}) := |\{i \in \{1, 2, \ldots, n\} : x_i \neq y_i\}|, \tag{2}$$

metrika az $\mathbb{F}_2^n$ *vektortéren.*

Bizonyítás. Legyen $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}_2^n$ tetszőleges. d pontosan akkor metrika az $\mathbb{F}_2^n$ vektortéren, ha a következő három feltétel mindegyikének eleget tesz:

1. $d(\mathbf{x}, \mathbf{y}) \geq 0$, valamint $d(\mathbf{x}, \mathbf{y}) = 0$ pontosan akkor, ha $\mathbf{x} = \mathbf{y}$;
2. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$;
3. $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$.

Az 1-es feltétel nyilván teljesül, hiszen $d(\mathbf{x}, \mathbf{y}) \geq 0$ (2)-ből adódóan. Valamint egyrészt, ha $d(\mathbf{x}, \mathbf{y}) = 0$, akkor (2) szerint $\mathbf{x}$ és $\mathbf{y}$ koordinátái rendre megegyeznek, azaz $\mathbf{x} = \mathbf{y}$. Másrészt, ha $\mathbf{x} = \mathbf{y}$, akkor $\mathbf{x}$ és $\mathbf{y}$ koordinátái rendre megegyeznek, így (2) szerint $d(\mathbf{x}, \mathbf{y}) = 0$.

A 2-es feltétel triviálisan teljesül.

A 3-as feltétel igazolásához definiáljuk minden $i \in \{1, 2, \ldots, n\}$-re a $\delta_i\colon \mathbb{F}_2^n \times \mathbb{F}_2^n \to \{0, 1\}$ leképezést úgy, hogy

$$\delta_i(\mathbf{x}, \mathbf{y}) := \begin{cases} 0, & \text{ha } x_i = y_i; \\ 1, & \text{ha } x_i \neq y_i. \end{cases}$$

Nyilván $d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \delta_i(\mathbf{x}, \mathbf{y})$. Legyen $i \in \{1, 2, \ldots, n\}$ tetszőleges.

Ha $\delta_i(\mathbf{x}, \mathbf{z}) = 0$, azaz $x_i = z_i$, akkor

- vagy $x_i = y_i = z_i$, és így

$$\delta_i(\mathbf{x}, \mathbf{z}) = 0 \leq 0 + 0 \leq \delta_i(\mathbf{x}, \mathbf{y}) + \delta_i(\mathbf{y}, \mathbf{z});$$

- vagy $x_i \neq y_i$, és $y_i \neq z_i$, és így

$$\delta_i(\mathbf{x}, \mathbf{z}) = 0 \leq 2 = 1 + 1 \leq \delta_i(\mathbf{x}, \mathbf{y}) + \delta_i(\mathbf{y}, \mathbf{z}),$$

ahonnan $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$.

Ha $\delta_i(\mathbf{x}, \mathbf{z}) = 1$, azaz $x_i \neq z_i$ akkor

- vagy $x_i = y_i$, és $y_i \neq z_i$, így

$$\delta_i(\mathbf{x}, \mathbf{z}) = 1 \leq 0 + 1 \leq \delta_i(\mathbf{x}, \mathbf{y}) + \delta_i(\mathbf{y}, \mathbf{z});$$

- vagy $x_i \neq y_i$, és így

$$\delta_i(\mathbf{x}, \mathbf{z}) = 1 \leq \delta_i(\mathbf{x}, \mathbf{y}) + \delta_i(\mathbf{y}, \mathbf{z}) = \begin{cases} 1 + 0 = 1, & \text{ha } y_i = z_i; \\ 1 + 1 = 2, & \text{ha } y_i \neq z_i, \end{cases}$$

ahonnan $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$.

Ezzel beláttuk az állítást. □

2.2. Definíció. A 2.1-es lemmában definiált d leképezést az $\mathbb{F}_2^n$ vektortéren vett *Hamming-távolságnak* nevezzük.

Megjegyzés. Két vektor Hamming-távolsága nem más, mint azon koordináták száma, amelyekben a két vektor eltér egymástól.

Így az $\mathbb{F}_2^n$ vektortér a d metrikával már egy $(\mathbb{F}_2^n, d)$ metrikus teret alkot. Egy metrikus térben az adott középpontú és adott sugarú gömböket célszerű definiálni.

2.3. Definíció. A $\mathbf{c} \in \mathbb{F}_2^n$ *középpontú* $r \in \mathbb{N}_0$ *sugarú gömbnek* az

$$S_r(\mathbf{c}) = \{\mathbf{x} \in \mathbb{F}_2^n \colon\ d(\mathbf{c}, \mathbf{x}) \leq r\}$$

halmazt nevezzük.

2.4. Definíció. Legyen $\mathbf{x} \in \mathbb{F}_2^n$, és legyen

$$\mathrm{wt}(\mathbf{x}) := |\{i \in \{1, 2, \ldots, n\} : x_i \neq 0\}|.$$

Ekkor $\mathrm{wt}(\mathbf{x})$-et az $\mathbf{x}$ szó *Hamming-súlyának* nevezzük.

Megjegyzés. Jelöljük $\mathbf{0}$-val a csupa 0 koordinátájú szót $\mathbb{F}_2^n$-ben. Ekkor nyilván tetszőleges $\mathbf{x} \in \mathbb{F}_2^n$-re $\mathrm{wt}(\mathbf{x}) = d(\mathbf{x}, \mathbf{0})$.

2.1. Kódok

2.5. Definíció. A $C \subseteq \mathbb{F}_2^n$ nem üres részhalmazt *bináris kódnak* nevezzük. C elemei a *kódszavak*, a kód *hossza* n.

2.6. Definíció. Legyen $C \subseteq \mathbb{F}_2^n$ kód, és $2 \leq |C|$. Legyen

$$d(C) := \min \{d(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in C,\ \mathbf{x} \neq \mathbf{y}\}.$$

Ekkor $d(C)$-t a C kód *minimum távolságának* nevezzük.

Megjegyzés. Az $|C| = 1$ esetben szokás a $d(C) = +\infty$ konvenciót alkalmazni.

2.7. Definíció. Legyen t pozitív egész szám. A $C \subseteq \mathbb{F}_2^n$ kódot $(2 \leq |C|)$ *t-hibajavító kódnak* nevezzük, ha bármely két különböző $\mathbf{x}, \mathbf{y} \in C$ kódszó esetén

$$d(\mathbf{x}, \mathbf{y}) \geq 2t + 1. \tag{3}$$

Megjegyzés. (3) ekvivalens azzal, hogy $d(C) \geq 2t + 1$.

2.8. Definíció. Legyen $C \subseteq \mathbb{F}_2^n$ egy t-hibajavító kód. A C kódot *perfekt kódnak* nevezzük, ha tetszőleges $\mathbf{x} \in \mathbb{F}_2^n$ szóhoz létezik tőle legfeljebb t távolságra lévő $\mathbf{c} \in C$ kódszó.

2.9. Lemma. *Ha C t-hibajavító kód, akkor tetszőleges $\mathbf{x} \in \mathbb{F}_2^n$ szóhoz legfeljebb egy olyan $\mathbf{c} \in C$ kódszó létezik, amelyre $d(\mathbf{c}, \mathbf{x}) \leq t$.*

Bizonyítás. Mivel C t-hibajavító kód, ezért a háromszög-egyenlőtlenségből következik, hogy a kódszavak körüli t-sugarú gömbök diszjunktak, ami éppen állításunk átfogalmazása. □

Megjegyzés. Azaz, ha C minimum távolsága $d(C)$, akkor C legfeljebb

$$t = \left\lfloor \frac{d(C) - 1}{2} \right\rfloor$$

hiba javítására képes, ahol $\lfloor \cdot \rfloor$ az alsó egészrészt jelöli.

A 2.9-es lemma megvilágítja a t-hibajavító kód elnevezést. Ha a küldő csak kódszavakat továbbít, és a továbbítás során legfeljebb t hiba lép fel, akkor a fogadó dekódolni tudja az eredeti üzenetet. Ugyanis a kapott üzenet pontosan egy kódszó köré írt t sugarú gömbben lesz benne, s ennek a gömbnek a középpontja volt az eredeti üzenet.

2.2. Lineáris kódok

Ha egy kódot a gyakorlatban akarunk alkalmazni, akkor tárolnunk kell a kódszavakat, meg kell határoznunk, a kód minimum távolságát, a dekódolásnál pedig ki kell számolnunk a kapott üzenetnek a kódszavaktól való távolságát. Ezeket a műveleteket a kódok egy speciális osztálya esetén sokkal egyszerűbben tudjuk elvégezni, mint az általános esetben.

2.10. Definíció. Ha a $C \subseteq \mathbb{F}_2^n$ kód lineáris altere az $\mathbb{F}_2^n$ vektortérnek, azaz $C \leq \mathbb{F}_2^n$, akkor C-t *lineáris kódnak* nevezzük. Ha C dimenziója éppen k, akkor lineáris $[n,k]$-kódnak nevezzük.

Ha $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k$ a C egy bázisa, akkor azt a $k \times n$-es $\mathbf{G}$ mátrixot, melynek i-edik sora a $\mathbf{c}_i$ vektor $(i = 1, 2, \ldots, k)$, C *generátormátrixának* nevezzük.

Lineáris kódok esetén a a kód minimum távolságának meghatározása egyszerű.

2.11. Lemma. *Egy lineáris kód minimum távolsága megegyezik a legkisebb súlyú nem nulla kódszó súlyával.*

Bizonyítás. Ha C lineáris kód, akkor $\mathbf{0} \in C$. Ezért

$$\begin{aligned} d(C) &= \min\{d(\mathbf{x},\mathbf{y}) : \mathbf{x},\mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}\} \leq \\ &\leq \min\{d(\mathbf{x},\mathbf{0}) : \mathbf{x} \in C, \mathbf{x} \neq \mathbf{0}\} = \mathrm{wt}(\mathbf{x}), \end{aligned}$$

tehát a minimum távolság nem nagyobb a legkisebb súlyú nem nulla kódszó súlyánál.

Másrészt ha $\mathbf{x}$ és $\mathbf{y}$ olyan kódszavak, melyekre $d(C) = d(\mathbf{x},\mathbf{y})$, akkor a linearitás miatt $\mathbf{x} - \mathbf{y} \in C$. Mivel

$$\mathrm{wt}(\mathbf{x}-\mathbf{y}) = d(\mathbf{x}-\mathbf{y},\mathbf{0}) = d(\mathbf{x}-\mathbf{y},\mathbf{y}-\mathbf{y}) = d(\mathbf{x},\mathbf{y}) = d(C),$$

ezért van olyan kódszó, amelynek súlya éppen a minimum távolság. □

2.12. Definíció. Legyen $C \subseteq \mathbb{F}_2^n$ egy tetszőleges kód. C *duális kódja* a

$$C^\perp = \{\mathbf{x} \in \mathbb{F}_2^n : \langle \mathbf{x}, \mathbf{c} \rangle = 0 \ \forall \mathbf{c} \in C\},$$

ahol $\langle \cdot, \cdot \rangle$ a skaláris szorzatot jelöli.

Lineáris algebrából ismert, hogy ha C lineáris $[n,k]$-kód, akkor $C^\perp$ lineáris $[n, n-k]$-kód. Ennek a $C^\perp$ kódnak egy $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$ generátormátrixát a C kód *paritásellenőrző mátrixának* nevezzük.

A „jó" kódok azok, amelyek viszonylag sok hibát tudnak kijavítani. Lineáris kódok esetén a hibajavító képesség és a kód dimenziója azonban csak egymás rovására növelhető. Erről szól a következő, a kódelméletben alapvető egyenlőtlenség.

2.13. Tétel (Singleton-korlát). *Ha valamely C lineáris $[n, n-k]$-kód minimum távolsága d, akkor*

$$d \leq k + 1.$$

Bizonyítás. A 2.11-es lemma alapján elegendő megmutatnunk, hogy C-ben van olyan nem nulla kódszó, melynek súlya legfeljebb $k+1$. Ha C generátormátrixa $\mathbf{G} \in \mathbb{F}_2^{(n-k)\times n}$, akkor elemi transzformációkkal előállíthatunk $\mathbf{G}$-ből egy $\mathbf{G}'$ mátrixot, amely szintén generálja C-t, és alakja

$$\mathbf{G}' = \begin{pmatrix} \mathbf{I} & \mathbf{G}^* \end{pmatrix},$$

ahol $\mathbf{I} \in \mathbb{F}_2^{(n-k)\times(n-k)}$ egységmátrix, és $\mathbf{G}^* \in \mathbb{F}_2^{(n-k)\times k}$. Miután $\mathbf{G}^*$ tetszőleges sorában legfeljebb k nem nulla elem van, ezért $\mathbf{G}$ tetszőleges sorának megfelelő kódszó súlya legfeljebb $k+1$. □

2.14. Definíció. Ha egy lineáris $[n, n-k]$-kód minimum távolsága d, és teljesül a

$$d = k + 1$$

egyenlőség, akkor a kódot *MDS-kódnak* nevezzük.

A 2.13-as tétel szerint tehát adott dimenzió esetén az MDS-kódok minimum távolsága a lehető legnagyobb, illetve adott kódszóhossz és adott minimum távolság esetén az MDS-kódok a lehető legnagyobb dimenziójú lineáris kódok. Vagyis az MDS-kódok több szempontból is optimálisak.

3. Tranzitív bináris kódok

Legyen n természetes szám és $\sigma \in S_n$ tetszőleges permutáció. Jelölje az $a \in \{1, 2, \ldots, n\}$ σ melletti képét $\sigma(a)$. Az 1.15-ös példa szerint S_n hat $\mathbb{F}_2^n$-en, és tetszőleges $(x_1, x_2, \ldots, x_n) = \mathbf{x} \in \mathbb{F}_2^n$-re

$$\sigma * \mathbf{x} = \left(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)}\right),$$

és jelöljük $\sigma * \mathbf{x}$-et $\sigma(\mathbf{x})$-szel. Hasonlóan, ha $C \subseteq \mathbb{F}_2^n$ kód, akkor a

$$\{\sigma(\mathbf{x}) : \mathbf{x} \in C\}$$

halmazt jelöljük $\sigma(C)$-vel.

3.1. Definíció. Legyen $C \subseteq \mathbb{F}_2^n$ tetszőleges kód. Azt mondjuk, hogy a $\sigma \in S_n$ permutáció *automorfizmusa* a C kódnak, ha

$$C = \sigma(C) = \{\sigma(\mathbf{x}) : \mathbf{x} \in C\}.$$

A C kód automorfizmusainak halmazát $\mathrm{Aut}(C)$-vel jelöljük.

Megjegyzés. Nyilvánvalóan $\mathrm{Aut}(C) \subseteq S_n$.

3.2. Állítás. *Legyen $C \subseteq \mathbb{F}_2^n$ tetszőleges kód. Ekkor* $\mathrm{Aut}(C)$ *részcsoportot alkot S_n-ben.*

Bizonyítás. Tekintsük az $1 \in S_n$ elemet. Definíció szerint

$$1(C) = \{1(\mathbf{x}) : \mathbf{x} \in C\} = \{\mathbf{x} : \mathbf{x} \in C\} = C,$$

amiből következik, hogy $1 \in \mathrm{Aut}(C)$, tehát $\mathrm{Aut}(C)$ nem üres.

Legyenek $\pi, \sigma, \tau \in \mathrm{Aut}(C)$ tetszőleges permutációk. Mivel $\mathrm{Aut}(C) \subseteq S_n$, ezért $(\pi\sigma)\tau = \pi(\sigma\tau)$. Ebből pedig nyilván adódik, hogy

$$((\pi\sigma)\tau)(C) = (\pi(\sigma\tau))(C).$$

Fentebb láttuk, hogy $1 \in S_n$-re $1(C) = C$, azaz $1 \in \mathrm{Aut}(C)$. Tehát tetszőleges $\sigma \in \mathrm{Aut}(C)$-re $1\sigma = \sigma 1 = \sigma$.

Legyen $\sigma \in \mathrm{Aut}(C)$ tetszőleges permutáció. Ekkor egyértelműen létezik $\sigma^{-1} \in S_n$ úgy, hogy $\sigma\sigma^{-1} = \sigma^{-1}\sigma = 1$. Mivel $\sigma \in \mathrm{Aut}(C)$, ezért $C = \sigma(C)$. Ekkor

$$\sigma^{-1}(C) = \sigma^{-1}(\sigma(C)) = \left\{\sigma^{-1}(\sigma(\mathbf{x})) : \mathbf{x} \in C\right\} =$$
$$= \left\{\sigma^{-1}\sigma(\mathbf{x}) : \mathbf{x} \in C\right\} = \{\mathbf{x} : \mathbf{x} \in C\} = C,$$

azaz $\sigma^{-1} \in \mathrm{Aut}(C)$.

Ezzel beláttuk az állítást. □

3.3. Definíció. Legyen $C \subseteq \mathbb{F}_2^n$ tetszőleges kód. C-t *tranzitív kódnak* mondjuk, ha $\mathrm{Aut}(C)$ tranzitívan hat $\{1, 2, \ldots, n\}$-en.

Megjegyzés. Ez nyilván azzal ekvivalens, hogy $\mathrm{Aut}(C)$ tranzitív részcsoport S_n-ben.

3.1. A problémák

Legyenek n, d $(1 \leq d \leq n)$ természetes számok és $G \leq S_n$ tranzitív részcsoport.

1. Probléma. Határozzuk meg azokat a $C \subseteq \mathbb{F}_2^n$ n-hosszúságú, d minimum távolságú tranzitív bináris kódokat, amelyekre $\mathrm{Aut}(C) = G$!

Ha meg tudjuk határozni azokat a C kódokat, amelyekre $d(C) \geq d$, és $\mathrm{Aut}(C) \geq G$, akkor az eredeti feladat már megoldható, hiszen ezen kódok között szerepelnek az 1-es probléma megoldásai is.

2. Probléma. Határozzuk meg azokat a $C \subseteq \mathbb{F}_2^n$ n-hosszúságú, legalább d minimum távolságú bináris kódokat, amelyekre $\mathrm{Aut}(C) \geq G$!

A kódelméleti szakaszban említettek miatt (jobb tárolhatóság, kezelhetőség, stb.) gyakran a fenti kódok közül csak a lineárisak érdekesek számunkra, természetesen így is megfogalmazható lenne a feladat.

3. Probléma.

1. Határozzuk meg azokat a $C \subseteq \mathbb{F}_2^n$ n-hosszúságú, d minimum távolságú lineáris tranzitív bináris kódokat, amelyekre $\mathrm{Aut}(C) = G$!
2. Határozzuk meg azokat a $C \subseteq \mathbb{F}_2^n$ n-hosszúságú, legalább d minimum távolságú lineáris bináris kódokat, amelyekre $\mathrm{Aut}(C) \geq G$!

A fenti problémák megoldásainak algoritmusait a következő alszakasz mutatja be.

3.2. Az algoritmusok

Az algoritmusok ismertetése előtt szükségünk van néhány fogalomra.

3.4. Definíció. Legyen (V, E) véges egyszerű gráf és $w\colon V \to \mathbb{N}$ egy súlyfüggvény a gráf csúcsain. A $K \subseteq V$ csúcshalmazt *klikknek* vagy *teljes részgráfnak* nevezzük, ha bármely $u, v \in K$ $(u \neq v)$ csúcsra $\{u, v\} \in E$. A K klikk *súlya*

$$w(K) = \sum_{v \in K} w(v).$$

Megjegyzés. Legyen $v \in V$ tetszőleges. Speciálisan a $\{v\} \subseteq V$ részhalmazt is klikknek tekintjük.

3.5. Definíció. Legyenek n, d természetes számok, $1 \leq d \leq n$, legyen $G \leq S_n$ részcsoport és $\mathbf{x} \in \mathbb{F}_2^n$ tetszőleges elem. Tekintsük G-nek az 1.15-ös példa szerinti hatását $\mathbb{F}_2^n$-en. Ekkor az $\mathbf{x}$ elem

$$G(\mathbf{x}) = \{g(\mathbf{x}) : g \in G\}$$

pályáját (orbitját) *d-pályának* nevezzük, ha bármely $\mathbf{y}, \mathbf{z} \in G(\mathbf{x})$ $(\mathbf{y} \neq \mathbf{z})$ esetén $d(\mathbf{y}, \mathbf{z}) \geq d$.

Megjegyzés. Az, hogy bármely $\mathbf{y}, \mathbf{z} \in G(\mathbf{x})$ $(\mathbf{y} \neq \mathbf{z})$ esetén $d(\mathbf{y}, \mathbf{z}) \geq d$ nyilván ekvivalens azzal, hogy $d(G(\mathbf{x})) \geq d$. Ebből a megfogalmazásból következik, hogy az egy elemű pályák tetszőleges $d \in \{1, 2, \ldots, n\}$-re d-pályák (lásd a 2.6-os definíciót követő megjegyzést).

Legyenek n, d $(1 \leq d \leq n)$ természetes számok és $G \leq S_n$ tranzitív részcsoport rögzítettek. Tekintsük G pályáit (orbitjait) $\mathbb{F}_2^n$-ben, jelöljük a pályák halmazát $\mathcal{P}$-vel.

Legyen $\mathcal{V} \subseteq \mathcal{P}$ a d-pályák halmaza, azaz

$$\mathcal{V} = \{P\colon\ d(P) \geq d \text{ és } P \in \mathcal{P}\}.$$

$\mathcal{V}$ elemei lesznek a gráfunk csúcsai. Legyen a $w\colon \mathcal{V} \to \mathbb{N}$ súlyfüggvény olyan, hogy bármely $P \in \mathcal{V}$-re

$$P \mapsto w(P) = |P|,$$

azaz a P d-pálya, mint csúcs súlya legyen a pálya számossága.

Legyen az $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ élhalmaz olyan, hogy bármely két különböző $P, Q \in \mathcal{V}$-re $\{P, Q\} \in \mathcal{E}$ pontosan akkor, ha bármely $\mathbf{x} \in P$ és bármely $\mathbf{y} \in Q$ esetén $d(\mathbf{x}, \mathbf{y}) \geq d$.

Megjegyzés. Mivel P és Q is d-pályák, ezért az előző azzal ekvivalens, hogy $\{P, Q\} \in \mathcal{E}$ pontosan akkor, ha $d(P \cup Q) \geq d$.

3.6. Definíció. Legyenek n, d természetes számok, $1 \leq d \leq n$, legyen $G \leq S_n$ részcsoport. Tekintsük az előzőekben konstruált $(\mathcal{V}, \mathcal{E})$ véges egyszerű gráf klikkjeinek $\mathfrak{K}$ halmazát. Legyen $\mathcal{K} \in \mathfrak{K}$ tetszőleges klikk. A

$$C_{\mathcal{K}} = \bigcup_{P \in \mathcal{K}} P$$

kódot a $\mathcal{K} \in \mathfrak{K}$ *klikkhez tartozó kódnak* nevezzük.

3.7. Állítás. *Legyenek n, d természetes számok, $1 \leq d \leq n$, legyen $G \leq S_n$ részcsoport. Tetszőleges, a 3.6-ban definiált $C_{\mathcal{K}}$ kódra*

1. $|C_{\mathcal{K}}| = w(\mathcal{K})$;
2. $d(C_{\mathcal{K}}) \geq d$ *és*
3. $G \leq \mathrm{Aut}(C)$.

Bizonyítás. Legyen $C_{\mathcal{K}}$ tetszőleges, a 3.6-os definíciót kielégítő kód.

1. Mivel bármely két különböző $P, Q \in \mathcal{V}$-re $P \cap Q = \emptyset$, ezért
$$|C_{\mathcal{K}}| = \left| \bigcup_{P \in \mathcal{K}} P \right| = \sum_{P \in \mathcal{K}} |P| = \sum_{P \in \mathcal{K}} w(P) = w(\mathcal{K}).$$

2. Ha a $C_{\mathcal{K}}$ kódnak egyetlen eleme van, akkor a bevezetett konvenciót követve $d(C_{\mathcal{K}}) = \infty$, azaz $d(C_{\mathcal{K}}) \geq d$ triviálisan teljesül.

   Ha $C_{\mathcal{K}}$-nak legalább két eleme van, akkor definíció szerint $d(C_{\mathcal{K}}) \geq d$ pontosan akkor, ha
$$\forall\, \mathbf{x}, \mathbf{y} \in C_{\mathcal{K}},\ \mathbf{x} \neq \mathbf{y}\colon\ d(\mathbf{x}, \mathbf{y}) \geq d. \tag{4}$$
   Mivel
$$C_{\mathcal{K}} = \bigcup_{P \in \mathcal{K}} P,$$
   és P-k mind d-pályák, ezért elég különböző $P, Q \in \mathcal{K}$ d-pályákból származó $\mathbf{x} \in P$-kre és $\mathbf{y} \in Q$-ra ellenőrizni a 4-es egyenlőtlenséget. Az $\mathcal{E}$ élhalmaz konstrukciójakor viszont csak azokat a $\{P', Q'\}$ csúcspárokat választottuk be $\mathcal{E}$-be, amelyekre $d(P' \cup Q') \geq d$.

   $\mathcal{K}$ klikk volta miatt azonnal adódik, hogy tetszőleges két különböző $P, Q \in \mathcal{K}$ d-pályákra $d(P' \cup Q') \geq d$, azaz tetszőleges $\mathbf{x} \in P$-re és tetszőleges $\mathbf{y} \in Q$-ra teljesül a 4-es egyenlőtlenség, így $d(C_{\mathcal{K}}) \geq d$.

3. Legyen $g \in G$ tetszőleges. Mivel
$$C_{\mathcal{K}} = \bigcup_{P \in \mathcal{K}} P,$$
   és P-k a G csoport pályái, ezért
$$g(C_{\mathcal{K}}) = g\left(\bigcup_{P \in \mathcal{K}} P \right) = \bigcup_{P \in \mathcal{K}} g(P) = \bigcup_{P \in \mathcal{K}} P = C_{\mathcal{K}}.$$
   Azaz tetszőleges $g \in G$-re $g \in \mathrm{Aut}(C_{\mathcal{K}})$, amiből következik, hogy $G \leq \mathrm{Aut}(C_{\mathcal{K}})$.

Ezzel beláttuk az állítást. □

A fenti konstrukció pszeudokódját láthatjuk az 1-es algoritmusban. Az így előállított kódokról a 3.7-es állításban láttuk, hogy biztosan megoldásai a 2-es problémának. Lehet-e más megoldás is?

1. algoritmus. A 2-es probléma megoldása

Bemenet. $1 \leq d \leq n$, $G \leq S_n$ tranzitív részcsoport.
Kimenet. Bármely $C_{\mathcal{K}} \in \mathcal{C}$-re $|C_{\mathcal{K}}| = w(\mathcal{K})$, $d(C_{\mathcal{K}}) \geq d$ és $\mathrm{Aut}(C_{\mathcal{K}}) \geq G$.

procedure TRANSITIVEBINARYCODES(n, d, G)
  $\mathcal{P} \leftarrow G$ *pályáinak halmaza* $\mathbb{F}_2^n$*-ben*
  $\mathcal{V} \leftarrow \emptyset$ ▷ Csúcshalmaz inicializálása.
  for $P \in \mathcal{P}$ **do**
    if $d(P) \geq d$ **then** ▷ d-pályák beválasztása a csúcshalmazba.
      $\mathcal{V} \leftarrow \mathcal{V} \cup \{P\}$
    end if
  end for
  $\mathcal{E} \leftarrow \emptyset$ ▷ Élhalmaz inicializálása.
  for $P, Q \in \mathcal{V}$ $(P \neq Q)$ **do**
    if $d(P \cup Q) \geq d$ **then** ▷ Élek behúzása.
      $\mathcal{E} \leftarrow \mathcal{E} \cup \{\{P, Q\}\}$
    end if
  end for
  $\mathfrak{K} \leftarrow (\mathcal{V}, \mathcal{E})$ *gráf klikkjeinek halmaza*
  $\mathcal{C} \leftarrow \emptyset$ ▷ A kódok halmazának inicializálása.
  for $\mathcal{K} \in \mathfrak{K}$ **do**
    $C_{\mathcal{K}} \leftarrow \emptyset$ ▷ A $\mathcal{K}$ klikknek megfelelő kód inicializálása.
    for $P \in \mathcal{K}$ **do**
      $C_{\mathcal{K}} \leftarrow C_{\mathcal{K}} \cup P$
    end for
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{C_{\mathcal{K}}\}$
  end for
  return $\mathcal{C}$
end procedure

3.8. Állítás. *A 2-es probléma megoldásai pontosan azok a kódok, amelyek a fenti konstrukcióval előállnak (az 1-es algoritmus által előállítottak).*

Bizonyítás. Azt, hogy minden, az 1-es algoritmussal előállított kód valóban megoldása a 2-es problémának, már láttuk a 3.7-es állítás bizonyítása-

kor. Most azt kell belátnunk, hogy bármely megoldás előállítható az 1-es algoritmussal.

Legyen a C bináris kód megoldása a 2-es problémának, azaz $\mathrm{Aut}(C) \geq G$ és $d(C) \geq d$. Mivel $\mathrm{Aut}(C) \geq G$, ezért bármely $g \in G$-re $g(C) = C$. Ezt megfogalmazhatjuk pályákkal is:

$$\bigcup_{\mathbf{x} \in C} G(\mathbf{x}) = C.$$

Tekintsük ezen pályák közül az összes különbözőt, legyen k ilyen, és jelöljük őket $P_1, P_2, \ldots, P_k$-val (a pályák diszjunktak). Így az előzővel ekvivalens

$$\bigcup_{i=1}^{k} P_i = C$$

megfogalmazáshoz jutunk.

$d(C) \geq d$ pontosan akkor teljesül, ha bármely $1 \leq i \leq k$-ra $d(P_i) \geq d$, és tetszőleges $1 \leq i, j \leq k$ $(i \neq j)$ esetén $d\left(P_i \cup P_j\right) \geq d$. Azaz C olyan d-pályák uniójaként áll elő, hogy tetszőleges P, Q két különböző ilyen pályára $d(P \cup Q) \geq d$. Mivel az 1-es algoritmus éppen az ilyen tulajdonságú kódokat állítja elő, ezért az állítás valóban igaz. □

Ezt követően az eredeti feladat, az 1-es probléma megoldása már egyszerű: az 1-es algoritmus $\mathcal{C}$ kimenetéből – ami a 2-es problémában keresett kódok mindegyikét tartalmazza – kell meghatároznunk azokat, amelyekre a minimum távolság pontosan d, valamint automorfizmus-csoportjuk éppen G. Ezen algoritmus pszeudokódja látható a 2-as algoritmusban.

2. algoritmus. Az 1-es probléma megoldása

Bemenet. Az 1-es algoritmus $\mathcal{C}$ kimenete. d és G az ott szereplő d-vel és G-vel azonos.
Kimenet. Bármely $C \in \mathcal{F}$-re $d(C) = d$ és $\mathrm{Aut}(C) = G$.

```
1: procedure TBCFILTER(C, d, G)
2:     F ← ∅                          ▷ A megfelelő kódok halmazának inicializálása.
3:     for C ∈ C do
4:         if d(C) = d és Aut(C) = G then          ▷ A kiválasztás.
5:             F ← F ∪ {C}
6:         end if
7:     end for
8:     return F
9: end procedure
```

A 3-as probléma megoldása mondhatni mindennapos feladat: adott kódok közül kell kiválogatnunk a lineárisakat. Attól függően, hogy az 1-es vagy a 2-es probléma megoldásai között szereplő lineáris kódokat szeretnénk meghatározni, a 2-es algoritmus $\mathcal{F}$ vagy az 1-es algoritmus $\mathcal{C}$ kimenetéből kell kiszűrni a lineáris kódokat. A megoldást szemlélteti a 3-as algoritmusban látható pszeudokód.

3. algoritmus. A 3-as probléma megoldása

Bemenet. A 2-es algoritmus $\mathcal{F}$ kimenete.
Kimenet. Bármely $C \in \mathcal{L}$-re C lineáris kód, $d(C) = d$ és $\mathrm{Aut}(C) = G$.

```
1: procedure LinearTBCs(F)
2:     L ← ∅                          ▷ A lineáris kódok halmazának inicializálása.
3:     for C ∈ F do
4:         if C altér F_2^n-ben then          ▷ Lineáris kódok beválasztása.
5:             L ← L ∪ {C}
6:         end if
7:     end for
8:     return L
9: end procedure
```

Illetve ha az 1-es algoritmus $\mathcal{C}$ kimenetére szeretnénk alkalmazni, akkor:

Bemenet. Az 1-es algoritmus $\mathcal{C}$ kimenete.
Kimenet. Bármely $C \in \mathcal{L}$-re C lineáris kód, $d(C) \geq d$ és $\mathrm{Aut}(C) \geq G$.

```
1: procedure LinearTBCs(C)
2:     L ← ∅                          ▷ A lineáris kódok halmazának inicializálása.
3:     for C ∈ C do
4:         if C altér F_2^n-ben then          ▷ Lineáris kódok beválasztása.
5:             L ← L ∪ {C}
6:         end if
7:     end for
8:     return L
9: end procedure
```

Függelék

Az előző szakaszban leírt algoritmusok implementációja látható itt GAP-ben [GAP] megvalósítva. A GRAPE csomag [GRAPE] segítségével hatékonyan lehet klikkeket meghatározni egy gráfban, míg a GUAVA csomag [GUAVA] biztosítja a kódelméleti függvényeket.

Az implementációban a 2-es algoritmus egy logikai kapcsolóval hívható meg a TransitiveBinaryCodes függvényen belül: **true** érték esetén a függvény a 2-es algoritmus kimenetével tér vissza, míg **false** esetén az 1-es algoritmus kimenetével. A LinearTBCs függvény pedig a 3-as algoritmust kódolja: a TransitiveBinaryCodes tetszőleges kimenetére meghívva csak a lineáris kódok listájával tér vissza.

```
1  LoadPackage("guava");
2  LoadPackage("grape");
3  Unbind(n);
4  Unbind(d);
5  Unbind(group);
6  Unbind(filterQ);
7  TransitiveBinaryCodes:=function(n, d, group, filterQ)
8     local orbits, vertices, orbit, code, n_vertices,
        adjacency_m, vertex_1, vertex_2, vertex_12, i, j
        , graph, x, y, cliques, codes, clique, C,
        filtered_codes;
9     LoadPackage("guava");
10    LoadPackage("grape");
11    if (1 <= d) and (d <= n) and IsTransitive(group)
        then
12       orbits:=Orbits(group, GF(2)^n, Permuted);
13       vertices:=[];
14       for orbit in orbits do
15          if Size(orbit) = 1 then
16             Add(vertices, orbit);
17          else
18             code:=ElementsCode(orbit, GF(2));
19             if MinimumDistance(code) >= d then
20                Add(vertices, orbit);
21             fi;
22             Unbind(code);
23          fi;
24       od;
```

```
25        Unbind(orbit);
26        Unbind(orbits);
27        n_vertices:=Size(vertices);
28        adjacency_m:=NullMat(n_vertices, n_vertices);
29        for vertex_1 in vertices do
30           for vertex_2 in vertices do
31              if vertex_1 <> vertex_2 then
32                 vertex_12:=Concatenation(vertex_1,
                      vertex_2);
33                 code:=ElementsCode(vertex_12, GF(2));
34                 if MinimumDistance(code) >= d then
35                    i:=Position(vertices, vertex_1);
36                    j:=Position(vertices, vertex_2);
37                    adjacency_m[i][j]:=1;
38                    Unbind(j);
39                    Unbind(i);
40                 fi;
41                 Unbind(code);
42                 Unbind(vertex_12);
43              fi;
44           od;
45           Unbind(vertex_2);
46        od;
47        Unbind(vertex_1);
48        graph:=Graph(Group( () ), [1..n_vertices],
             OnPoints, function(x, y) return adjacency_m[x
             ][y] = 1; end, true);
49        cliques:=[];
50        for i in [1..n_vertices] do
51           cliques:=Concatenation(cliques,
                CompleteSubgraphs(graph, i));
52        od;
53        Unbind(i);
54        Unbind(graph);
55        Unbind(n_vertices);
56        codes:=[];
57        for clique in cliques do
58           C:=[];
59           for i in clique do
60              C:=Concatenation(C, vertices[i]);
61           od;
```

```
62            Unbind(i);
63            code:=ElementsCode(C, GF(2));
64            Unbind(C);
65            Add(codes, code);
66            Unbind(code);
67         od;
68         Unbind(clique);
69         Unbind(cliques);
70         if filterQ then
71            filtered_codes:=[];
72            for code in codes do
73               if Size(code) <> 1 then
74                  if (MinimumDistance(code) = d) and (
                       Stabilizer(SymmetricGroup(n), code,
                       PermutedCode) = group) then
75                     Add(filtered_codes, code);
76                  fi;
77               fi;
78            od;
79            Unbind(i);
80            return filtered_codes;
81         else
82            return codes;
83         fi;
84      else
85         return 0;
86      fi;
87   end;
88   Unbind(codes);
89   LinearTBCs:=function(codes)
90      local linear_codes, code;
91      LoadPackage("guava");
92      linear_codes:=[];
93      for code in codes do
94         if IsLinearCode(code) then
95            Add(linear_codes, code);
96         fi;
97      od;
98      Unbind(code);
99      return linear_codes;
100  end;
```

Hivatkozások

[CACD] Petteri Kaski – Patric R. J. Östergard, *Classification Algorithms for Codes and Designs*, Springer-Verlag, Berlin, 2006.

[VG] Kiss György – Szőnyi Tamás, *Véges geometriák*, Polygon, Szeged, 2001.

[BAA] Kiss Emil, *Bevezetés az absztrakt algebrába*, kiadó, hol, mikor.

[GAP] The GAP Group, *GAP – Groups, Algorithms, and Programming, Version 4.7.4*; 2014, (`http://www.gap-system.org`).

[GUAVA] Reinald Baart – Tom Boothby – Jasper Cramwinckel – Joe Fields – David Joyner – Robert Miller – Eric Minkes – Erik Roijackers – Lea Ruscio – Cen Tjhai, *GUAVA – a GAP package, Version 3.12*; 2012, (`http://www.southernct.edu/~fields/Guava/`).

[GRAPE] Leonard H. Soicher, *GRAPE – A GAP package, Version 4.6.1*; 2012, (`http://www.maths.qmul.ac.uk/~leonard/grape/`).

Nyilatkozat

Alulírott Mezőfi Dávid Csaba kijelentem, hogy a szakdolgozatban foglaltak saját munkám eredményei, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel. Tudomásul veszem, hogy szakdolgozatomat a Szegedi Tudományegyetem könyvtárában a kölcsönözhető könyvek között helyezik el, és az interneten is nyilvánosságra hozhatják.

Szeged, 2014. május 9. .....
aláírás