Linearne kodovanie

T. K.

May 12, 2010

Uvod (Lekcia 1)

Budeme pokracovat v tom, cim sme zacali v prednaske o kodovani. Strucne si zopakujeme niektore dolezite pojmy, oznacenia a vysledky. Nech A je (konecna) neprazdna mnozina. Prvky z A budeme volat tiez *abecedou* nad A. Konecnu postupnost $\mathbf{a} = a_1 a_2 \cdots a_k$ prvkov z A nazyvame *slovom* nad A. Cislo k volame dlzkou slova $\mathbf{a}$. A^k znamena mnozinu vsetkych slov dlzky k nad A. Dalej, A^* oznacuje mnozinu vsetkych slov nad A. Patri tam aj tzv. prazdne slovo. V pripade, ze A je dvojprvkova mnozina, napr. $A = \{0, 1\}$, tak hovorime o binarnej abecede, resp. o binarnych slovach. Vseobecne, ak $\mathbf{a}, \mathbf{b} \in A^*$, tak mozeme vyrobit nove slovo, tzv. *zlozene*, nasledovne: Nech $\mathbf{a} = a_1 \cdots a_k$, $\mathbf{b} = b_1 \cdots b_n$. Potom

$$\mathbf{a} \mid \mathbf{b} = a_1 \cdots a_k b_1 \cdots b_n.$$

Dostavame sa k prvej dolezitej definicii: Nech A a B su konecne neprazdne mnoziny. Nech

$$\varphi : A \to B^*$$

je proste zobrazenie. Potom φ volame *kodovanim* zdrojovej abecedy A v kodovej abecede B. Mnozinu slov $\{\varphi(A) : a \in A\}$ volame strucne *kodom*. V pripade, ze B je dvojprvkova, tak hovorime, ze φ je *binarny* kod.

Dalsia definicia: Kodovanie zdrojovych znakov $\varphi : A \to B^*$ sa da rozsirit na kodovanie zdrojovych sprav, t.j. slov v abecede A. Presnejsie, $\varphi^* : A^* \to B^*$ je definovane nasledovne:

$$\varphi^*(a_1 \cdots a_k) = \varphi(a_1) \mid \varphi(a_2) \mid \cdots \mid \varphi(a_k),$$

t.j. spravu kodujeme znak po znaku.

Zaverecna definicia: Hovorime, ze kodovanie $\varphi : A \to B^*$ je *jednoznacne dekodovatelne*, ak zo znalosti zakodovanej spravy $\varphi^*(a_1 \cdots a_k)$

mozeme jednoznacne urcit zdrojovu spravu $a_1 \cdots a_k$, t.j. ak $\varphi^* : A^* \to B^*$ je prostym zobrazenim.

Dalsia definicia: ak $\mathbf{b} = b_1 \cdots b_n$ je slovo nad B, tak podslova tvaru b_1, $b_1 b_2$, $\cdots$, $b_1 \cdots b_n$ volame *prefixom* slova **b**. Podobne hovorime aj o *sufixoch*. Teraz sa mozeme este dohodnut na definicii: Kodovanie $\varphi : A \to B^*$ nazyvame *prefixovym*, ak ziadne kodove slovo nie je prefixom ineho kodoveho slova. Plati

Veta 1. Prefixove (sufixove) kodovanie je jednoznacne dekodovatelne. (Preco?)

V dalsom budeme pracovat len s *blokovym* kodovanim, t.j. kodovanie bude mat tvar

$$\varphi : A \to B^n$$

pricom vieme, ze $B^n \subseteq B^*$. Rychle sa preveri, ze blokove kodovanie je prefixove a aj sufixove,

teda kazdy blokovy kod je automaticky jednoznacne dekodovatelny.

Pozrime sa kratko, co znamena *dekodovanie*. Nech teda je zadane jednoznacne dekodovatelnye kodovanie $\varphi : A \to B^*$. Potom (parcialne) zobrazenie

$$\delta : B^* \to \varphi(A)$$

s vlastnostou: $\mathbf{b} \in \varphi(A)$ implikuje $\delta(\mathbf{b}) = \mathbf{b}$. (Pripominame, ze $\varphi(A) \subseteq B^*$.)

Bezpecnostne kody

Podla beznej predstavy funguje v praxi kodovanie nasledovne: Mame povedzme ulohu nastartovat motory vzdialeneho satelitu (rakety). Tento povel musime najprv zakodovat, t.j. prelozit do takej "reci", aby sa to mohlo poslat cez eter ku satelitu. Toto vykonaju inzinieri spolu

s matematikmi. Ked je to hotove, tak zakodovanu spravu poslu pomocou elektromagnetickych vln smerom ku satelitu. Ten ich prijme a iste zariadenie prijatu spravu dekoduje, t.j. znova prelozi do reci pristrojov. Ked je to hotove, tak zapnu sa startery motorov. Kvoli prehladu, si to rozlozime na nasledovne body:

1. Priprava spravy s ulohou nastartovania motorov.

2. Zakodovanie spravy, aby sa mohla poslat cez eter ku satelitu.

3. Prijatie zakodovanej spravy satelitom.

4. Dekodovanie spravy, t.j. jej prelozenie do reci, ktoru "rozumie" satelit.

5. Vykonanie povelu.

Na jednom mieste dochadza casto ku porucham: V bode 3 ciha nebezpecie. Totiz nemame zarucene, ze satelit prijme tu istu zakodovanu spravu, ktoru sme mu poslali. V eteri dochadza ku roznym porucham. Jedna sa hlavne o dva druhy chyb: a) zamena vyslaneho znaku na iny znak, alebo b) vytvorenim znaku, ktory vobec nebol vyslany (porucha synchronizacie). O tychto chybach nebudeme tu hovorit. Pojde len o chyby prveho druhu.

Chceme poukazat na to, ze prijimac spravy moze chyby prveho druhu *objavit* a v priaznivom pripade aj *opravit*. Ako sa to urobi? Vsimnime si jeden trivialny priklad z praxe: Predpokladajme, ze sme vyslali slovenske slovo "opakovanie". Prijali sme vsak divne slovo "opakkvanie". To, co sme prijali nie je "kodovym" slovom, pretoze take slovo sa nenachadza v slovniku slovenskeho jazyka. Teda, objavili sme chybu. V tomto pripade vieme dokonca urobit viacej,

totiz chybu aj napravit. Existuje len jedno slovenske slovo, z ktoreho zmenou jedneho pismena ziskame nase prijate slovo. Je to, ako uz vieme slovo "opakovanie". Ak by sme totiz boli prijali slovo "opakkvanir", tak sme zaregistrovali chybu, ale ju uz nevieme opravit.
Preco? Pozrime sa blizsie na to.

Objavovanie chyb (2. lekcia)

Vo vseobecnosti mame dobry prehlad o kodovych slovach (to su tie slova v slovniku!).

Definicia 1. Nech $\varphi : A \to B^n$ je kod. Ak prijmeme nekodove slovo, tak hovorime, ze sme objavili chybu.

Samozrejme, ak sme prijali kodove slovo, tak bud nedoslo ku chybe, alebo doslo ku chybe, ale sme ju neobjavili. Ku pochopeniu tejto skutocnosti nam pomoze nasledovna definicia

Definicia 2. Hovorime o t-*nasobnej chybe,* ak pocet chybnych (zmenenych) miest v prijatom slove je aspon 1 a nanajvys t. V pripade $t = 1$ hovorime o *jednoduchych* chybach.

Definicia 3. Hovorime, ze kod $\varphi : A \to B^*$ *objavuje* $t-$nasobne chyby, ak pri vyslani kodoveho slova a vzniku t-nasobnej chyby je prijate vzdy *nekodove* slovo.

Definicia 4. Zoberme dve slova $\mathbf{a} = a_1 \cdots a_n$, $\mathbf{b} = b_1 \cdots b_n \in A^n$. Potom definujeme vzdialenost medzi slovami

$$\rho(\mathbf{a}, \mathbf{b}) = \mid \{i : a_i \neq b_i\} \mid .$$

Budeme to volat *Hammingovou* vzdialenostou.

Veta 2. Hammingova vzdialenost je metrikou na mnozine slov A^n.

Definicia 5. Nech $\varphi : A \to B^n$ je blokovy kod. Zrejme $\varphi(A) \subseteq B^n$, kde $\varphi(A)$ je mnozina kodovych slov. Nech

$$d = d_\varphi = \min\{\rho(\mathbf{v}, \mathbf{w}) : \mathbf{v}, \mathbf{w} \in \varphi(A), \mathbf{v} \neq \mathbf{w}\}.$$

Cislo d_φ volame *minimalnou vzdialenostou* kodu φ. (Toto cislo existuje, lebo kodovych slov je konecne vela.)

Veta 3. Nech $\varphi : A \to B^n$ je blokovy kod o minimalnej vzdialenosti d_φ. Potom φ objavuje t-nasobne chyby pre $t < d_\varphi$ a nie je schopny uz objavit vsetky t-nasobne chyby pre $t \geq d_\varphi$.

Priklad 1. Kod celkovej kontroly parity. Predpokladajme, ze mame binarny blokovy kod $\varphi : A \to B^n$. Zrejme $d = d_\varphi \geq 1$. Jednoduchym sposobom prerobime nas kod na iny, u ktoreho jednoduchsie skontrolujeme jednoduche chyby. Nech novy kod ma tvar

$$\psi : A \to B^{n+1}.$$

Definujme kodove slova nasledovne: Ak $\varphi(a) = v_1 \cdots v_n \in \varphi(A)$, tak

$$\psi(a) = v_1 \cdots v_n v_{n+1},$$

pricom $v_{n+1} = 1$, ak pocet cislic "1" medzi $v_1 \cdots v_n$ je neparny pocet. V opacnom pripade polozime $v_{n+1} = 0$. (Hovorime, ze sme dodali kontrolny znak.) Ukazeme, ze $d_\psi \geq 2$. Urobme to nepriamo. Nech $\mathbf{v}, \mathbf{w} \in \psi(a)$. Predpokladajme, ze $\rho(\mathbf{v}, \mathbf{w}) = 1$. Bez ujmy na vseobecnosti mozeme predpokladat, ze $v_1 = 1$, $w_1 = 0$,

$$v_2 = \cdots = v_k = w_2 = \cdots = w_k = 1$$

a sucasne

$$v_{k+1} = \cdots = v_{n+1} = w_{k+1} = \cdots = w_{n+1} = 0.$$

Obe slova maju parny pocet "1". Preto k je parne cislo. Lenze $\mathbf{w}$ ma $k-1$ zloziek rovnych 1, co je neparne cislo. Dostali sme spor. Teda plati $\rho(\mathbf{v}, \mathbf{w}) \geq 2$.

Priklad 2. Opakovaci kod. Nech $n \geq 2$. Mozeme vytvorit nasledovny pozoruhodny kod $\varphi : B \to B^n$ definovany vztahom

$$\varphi(b) = b_1 \cdots b_n \in B^n,$$

kde $b = b_1 = \cdots = b_n$. Volame ho *opakovacim* kodom. Rychle sa presvedcime, ze $d_\varphi = n$. (Vidime, ze pre kazde n mame kod s minimalnou vzdialenostou rovnou n.) Neskorsie si uvedieme sposob, ako mozeme pomocou opakovacieho kodu jednoduchym sposobom zvacsovat minimalnu vzdialenost kodov.

Opravovanie chyb

Definicia 6. Hovorime, ze kod $\varphi : A \to B^n$ *opravuje* t-nasobne chyby, ak pri vyslani kodoveho slova $\mathbf{v} \in \varphi(A)$ a prijateho slova $\mathbf{w} \in B^n$ s vlastnostou $\rho(\mathbf{v}, \mathbf{w}) \leq t$ plati

$$\rho(\mathbf{v}, \mathbf{w}) < \rho(\mathbf{x}, \mathbf{w})$$

pre kazde $\mathbf{x} \in \varphi(A)$ a $\mathbf{x} \neq \mathbf{v}$.

Veta 4. Blokovy kod minimalnej vzdialenosti d opravuje t-nasobne chyby pre vsetky

$$t < d/2.$$

Konecne polia

Najprv musime nieco zopakovat o konecnych poliach. Klasicky vysledok z algebry hovori, ze okruh zvyskovych tried $Z_n = (Z_n; +, .)$ je polom prave vtedy, ked n je prvocislo. Takychto poli mame nekonecne vela: Z_2, $Z_3, \cdots$, $Z_p, \cdots$. To nie je vsetko. Plati este

Veta 5. Nech F je konecne pole charakteristiky p. Potom $\mid F \mid = p^k$. (Vieme, ze p je prvocislo.)

Veta 6. Nech p je dane prvocislo a nech n je lubovolne prirodzene cislo. Potom existuje pole F, pre ktore plati: $\mid F \mid = p^n$.

Veta 7. Nech F a E su konecne polia. Potom $F \cong E$ prave vtedy, ked $\mid F \mid = \mid E \mid$.

Veta 8. Nech F je pole. Nech $F[x]$ je okruh polynomov v neurcitej x nad polom F. Nech $p(x) \in F[x]$ je ireducibilny polynom nad F. Potom $F[x]/(p(x))$ je pole, ktore je izomorfne s nadpolom pola F. Specialne, nech $p(x) \in Z_p[x]$ je ireducibilny polynom stupna n. Potom

$$\mid Z_p[x]/(p(x)) \mid = p^n.$$

Linearne kody (3. lekcia)

Ak F znamena (konecne) pole, tak $\mathbf{F}^n$ bude oznacovat mnozinu vsetkych slov $\mathbf{a} = a_1 \cdots a_n$

dlzky n nad polom F. Samozrejme, na slova sa mozeme divat ako na vektory, t.j. usporiadane $n-$tice prvkov z F. Dalej vidime, ze na $\mathbf{F}^n$ mame definovanu binarnu operaciu $+$ nasledovne:

$$\mathbf{a} + \mathbf{b} = (a_1 + b_1) \cdots (a_n + b_n).$$

Rychle sa presvedcime (urobte to !), ze $(\mathbf{F}^n; +)$ je komutativna grupa. Mozeme ist dokonca dalej. Da sa jednoducho ukazat (preverte to!), ze dvojica $(F; \mathbf{F}^n)$ je konecny vektorovy priestor, ak nasobenie skalarmi definujeme nasledovne:

$$c\mathbf{a} = (ca_1) \cdots (ca_n)$$

pre kazde $c \in F$. Taktiez vidime, ze konecny priestor automaticky znamena konecno-rozmerny vektorovy priestor.

V dalsom budeme stale predpokladat, ze F je konecne pole. Budeme sa zaoberat kodami tvaru $\varphi : A \to \mathbf{F}^n$ v zmysle nasledujucej definicie.

Definicia 7. Nech F je konecne pole. Potom proste zobrazenie $\varphi : A \to \mathbf{F}^n$ volame *linearnym kodom*, ak mnozina kodovych slov

$$K = \varphi(A) \subseteq \mathbf{F}^n$$

je podpriestorom vektoroveho priestoru $(F; \mathbf{F}^n)$. Dalej, ak sme presnejsi, hovorime, ze φ je linearnym $(n, k)-$kodom, ak

$$\dim(\varphi(A)) = k.$$

Zvlastna situacia nastane, ked $k = 0$ alebo $k = n$. Vtedy hovorime o *trivialnom* kode. Vacsinou sa budeme zaoberat netrivialnymi kodami. Dalsia zvlastnost nastane, ked $F = Z_2$. Bude to vlastne binarny linearny kod. Pripomenieme si este jednu dolezitu vetu z linearnej algebry. K tomu potrebujeme oznacenia. Obycajne znamena $\mathtt{A}$ nejaku maticu typu $m \times n$, t.j. ktora ma m riadkov a n stlpcov. Specialne, $\mathtt{a}$ moze znamenat riadkovu maticu, resp. slovo, dlzky n,

co sme doteraz zapisovali aj ako $\mathbf{a}$. Znakom $\mathtt{A}^T$ zapisujeme transponovanu maticu ku $\mathtt{A}$, t.j. ak v $\mathtt{A}$ navzajom vymenime riadky za odpovedajuce stlpce.

Veta 9. Nech

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = 0$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = 0$$

$$\cdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = 0.$$

je system linearnych homogennych rovnic nad polom F s maticou sustavy $\mathtt{A}$. Potom riesenia tohto systemu tvoria podpriestor vektoroveho priestoru $(F; \mathbf{F}^n)$. Dalej, ak $h(\mathtt{A}) = r$ (=hodnost matice sustavy), tak $n - r = k$ je dimenziou podpriestoru rieseni.

Horeuvedeny system rovnic mozeme zapisat aj pomocou matic. Vyzera to nasledovne:

$$\mathtt{A}\mathtt{x}^T = \mathtt{0}^T.$$

Priklad 3. Nech $F = Z_2$. Uvazujme o systeme linearnych homogennych rovnic

$$x_1 + \cdots + x_n = 0.$$

Jeho riesenim su vsetky slova (vektory) $\mathtt{v} = v_1 \cdots v_n \in Z_2^n$ s vlastnostou $v_1 + \cdots + v_n = 0$. Teda sa jedna o kod celkovej kontroly parity, ktory je linearnym $(n, n-1)$-kodom.

Priklad 4. Nech F je konecne pole. Opakovaci kod dlzky n je linearny $(n, 1)$-kod nad F a da sa opisat nasledovnym homogennym systemom linearnych rovnic

$$x_1 + (-1)x_2 = 0$$

$$x_1 + (-1)x_3 = 0$$

$$\ldots$$

$$x_1 + (-1)x_n = 0.$$

Priklad 5. "Koktavy" kod (pre $n = 6$) a pole F je (6,3)-linearnym kodom a da sa opisat systemom rovnic

$$x_1 + (-1)x_2 = 0$$

$$x_3 + (-1)x_4 = 0$$

$$x_5 + (-1)x_6 = 0.$$

Priklad 6. Nie kazdy kod je linearny. Napr. kod "2 z 5" nie je linearnym kodom, pretoze tam nepatri nulove slovo. Pripominame, ze kod "2 z 5" vyzera nasledovne: $A = \{0, 1 \cdots, 9\}$ a $1 \mapsto 11000$, $2 \mapsto 10100$, $3 \mapsto 10010$, $4 \mapsto 10001$, $5 \mapsto 01001$, $6 \mapsto 00101$, $7 \mapsto 00011$, $8 \mapsto 00110$, $9 \mapsto 01100$, $0 \mapsto 01010$.

Generujuca matica

Ukazeme, ze linearny kod mozeme charakterizovat dvoma maticami: generujucou a kontrolnou. Dalej ukazeme, ako sa z jednej matice da urcit prislusna druha matica. Vsimnime si najprv jednu dolezitu vlastnost linearnych kodov. Ak $\varphi : A \to \mathbf{F}^n$, je linearny kod, tak $\varphi(A)$ je podpriestorom vektoroveho priestoru $(F; \mathbf{F}^n)$. Pretoze φ je proste zobrazenie, tak plati:

$$| A | = | \varphi(A) | .$$

Dalej, vieme ze $\dim(\varphi(A)) = k$. Teda, $\varphi(A)$ ma aspon jednu k-prvkovu bazu $\mathrm{g}_1, \cdots, \mathrm{g}_k$. Pretoze $\mathrm{g}_i \in \mathbf{F}^n$, tak slova g_i vieme prepisat nasledovne:

$$\mathrm{g}_1 = g_{11} g_{12} \cdots g_{1n},$$

$$\mathrm{g}_2 = g_{21} g_{22} \cdots g_{2n},$$

$$\cdots$$

$$\mathrm{g}_k = g_{k1} g_{k2} \cdots g_{kn}.$$

Z tychto hodnot mozeme vytvorit maticu $\mathtt{G} = (g_{ij})$ typu $k \times n$.

Definicia 8. Horeuvedenu maticu $\mathtt{G} = (g_{ij})$ budeme volat *generujucou* maticou linearneho kodu φ.

Poznamka Generujuca matica nie je jednoznacne urcena. Pretoze existuje viacej baz podpriestoru $\varphi(A)$, tak preto existuje aj viacej generujucich matic. Vsetky generujuce matice kodu φ maju nasledovne vlastnosti:

(i) pocet riadkov je $k = \dim(\varphi(A))$ a n je pocet stlpcov;

(ii) riadky generujucej matice su linearne nezavisle (ako vektory);

(iii) kazdy riadok matice je kodovym slovom;

(iv) kazde kodove slovo kodu φ je linearnou kombinaciou riadkov generujucej matice.

Informacne znaky (4. lekcia)

Z doterajsich prikladov sme uz zistili (opakovaci kod, koktavy kod alebo kod celkovej kontroly parity), ze z dodanim vhodnych znakov ku existujucim slovam, mozeme zlepsit kvalitu prijatych sprav. To nas vedie k tomu, ze znaky v kodovych slovach delime na *informacne* (tie, ktore mozeme lubovolne urcit) a na *kontrolne* (tie, ktore su uplne urcene informacnymi znakmi). U blokovych (linearnych) kodov to mozeme este upresnit v nasledujucom zmysle

Definicia 9. Nech $\varphi(A) = K \subseteq B^n$ je blokovy (linearny) kod. Ak existuje take cislo k a bijektivne zobrazenie $\psi : B^k \to K$, tak hovorime,

ze kod K ma k *informacnych* a $n-k$ *kontrolnych* znakov. Zobrazenie ψ volame *kodovanim informacnych znakov*.

Definicia 10. Blokovy kod $K \subseteq B^n$ volame *systematickym*, ak existuje take cislo $k < n$, ze zobrazenie $\psi : B^k \to K$ definovane vztahom

$$\psi : v_1 \cdots v_k \mapsto v_1 \cdots v_k v_{k+1} \cdots v_n$$

je kodovanim informacnych znakov. Volame ho *systematickym*.

Vsimnime si, ze u systematickeho kodu je poslednych $n-k$ znakov kontrolnych. Koktavy kod nepatri do triedy systematickych kodov. Dolezity je aj nasledovny pojem

Definicia 11. Nech $\varphi : A \to B^n$ je blokovy kod. Nech k je pocet informacnych znakov. Potom cislo

$$R = k/n$$

volame *informacnym pomerom* kodu φ.

Veta 10. Nech $\varphi : A \to \mathbf{F}^n$ je linearny (n, k)-kod a nech $\mathbf{b}_1, \cdots, \mathbf{b}_k$ je baza $\varphi(A) = K$. Potom φ ma k informacnych znakov a zobrazenie $\psi : \mathbf{F}^k \to \mathbf{F}^n$ definovane predpisom

$$\psi : u_1 \cdots u_k \mapsto u_1\mathbf{b}_1 + \cdots + u_k\mathbf{b}_k$$

je kodovanim informacnych znakov a sucasne je linearne.

Dokaz. Kedze φ je linearny (n, k)-kod, tak K ma k-prvkovu bazu a lubovolny prvok $\mathbf{u} \in K$ sa da podla nasej bazy jednoznacne zapisat v tvare, aky je uvedeny pri tvorbe zobrazenia ψ. Teraz uz rychle preverime, ze ψ je proste linearne zobrazenie vektorovych priestorov. Preverte to!

Poznamka. Poucenie z predchadzajucej vety: Pri linearnom kodovani $\varphi : A \to \mathbf{F}^n$ mnozinu A

mozeme vzdy povazovat za $\mathbf{F}^k$ pre vhodne k, kde k udava pocet informacnych znakov, alebo co je to iste, $k = \dim(\varphi(A))$. Ak $A = \mathbf{F}^k$, tak pri linearnom kode $\varphi : \mathbf{F}^k \to \mathbf{F}^n$ ziskame bazu v K jednoducho: $\varphi(10\cdots0), \cdots, \varphi(0\cdots01)$.

V dalsom vidime, ze pri vybere (konstrukcii) linearneho kodu sa snazime dosiahnut toho, aby pocet informacnych znakov k bol velky (t.j. aby redundancia bola mala) a sucasne, aby minimalna vzdialenost kodu d bola velka, t.j. aby kod mohol objavovat a opravovat vela chyb. Tieto poziadavky su rozporne a preto hladame vhodny kompromis. Samozrejme, ze su este dolezite otazky realizacie nasich poziadaviek (t.j. vhodnych algoritmov na objavovanie a opravovanie chyb). Na tazkosti poukazuje uz aj nasledujuce tvrdenie

Veta 11. Nech d znamena minimalnu vzdialenost systematickeho kodu $\varphi : A \to B^n$. Potom

$$d \leq n - k + 1.$$

Dokaz. Podla definicie systematickeho kodu vieme, ze existuje k s vlastnostou: mame take bijektivne zobrazenie $\psi : B^k \to \varphi(A)$, ze

$$\psi : v_1 \cdots v_k \mapsto v_1 \cdots v_k v_{k+1} \cdots v_n.$$

Vyberme si slovo $\mathbf{v} = v_1 \cdots v_{k-1} \in B^{k-1}$. Nech K_0 je mnozina vsetkych slov z $\varphi(A)$, ktore maju za prefix slovo $\mathbf{v}$. Potom pre minimalnu vzdialenost d_0 slov z K_0 plati

$$d_0 \leq n - (k-1) = n - k + 1.$$

Pretoze $K_0 \subseteq \varphi(A)$, tak mame $d \leq d_0$.

Je zaujimave si vsimnut, ze systematicky linearny kod ma za generujucu maticu

$$\mathtt{G} = (\mathtt{E} \mid \mathtt{C})$$

typu $k \times n$, pricom $\mathtt{E}$ je jednotkova matica stupna k a $\mathtt{C}$ je maticou typu $k \times (n-k)$. (Pozri poznamku za vetou 10.)

Definicia 12. Blokove kody K a K_1 sa nazyvaju *ekvivalentne*, ak existuje taka permutacia $\pi \in S_n$, ze

$$v_1 \cdots v_n \in K \;\Leftrightarrow\; v_{\pi(1)} \cdots v_{\pi(n)} \in K_1.$$

Veta 12. Kazdy linearny kod je ekvivalentny s nejakym systematickym linearnym kodom.

Dokaz. Pouzijeme znamu metodu z 1. rocnika: uprava matice pomocou elementarnych riadkovych operacii. Pripominame, ze su tri zakladne operacie: I. vymena riadkov, II. vynasobenie riadku nenulovym prvkom z pola F a III. vynasobeny i-riadok pripocitame ku j-temu riadku. Pomocou riadkovych operacii vieme nasu generujucu maticu `G` upravit na tzv. schodikovy tvar. Presnejsie: 1. kazdy riadok je bud nulovy, alebo nie. V druhom pripade existuje prvy $a_{ij} \neq 0$, t.j. j je minimalne. Ziadame $a_{ij} = 1$ (=pivotny prvok); 2. v tom stlpci,

v ktorom existuje pivotny prvok, existuju inac len nulove prvky; 3. najprv idu nenulove riadky a po nich nulove; 4. ak s_i a s_j su stlpcove indexy pivotnych prvkov z i-teho a j-teho riadku, tak $s_i < s_j$. Pretoze $h(\mathtt{G}) = k$, tak kazdy riadok novej matice je nenulovy. Kedze mame k nenulovych riadkov, tak mame k pivotnych prvkov. V ziadnom stlpci nelezia dva pivotne prvky. Teda, pivotne prvky lezia v stlpcoch

$$1 \leq j_1 < \cdots < j_k \leq n.$$

Postarame sa o to, aby pivotne prvky (co su 1-ky) lezali v prvych k stlpcoch. Urobime to pomocou vymeny stlpcov (transpozicii). T.j. vymenime j_1-vy stlpec s 1. stlpcom atd. Zlozenim vsetkych potrebnych transpozicii ziskame permutaciu π, ktoru potrebujeme urobit na stlpcoch. Tak zaverom dostaneme novu generujucu maticu

$$\mathtt{G}_1 = (\mathtt{E} \mid \mathtt{C}),$$

ktora je ekvivalentna s $\mathtt{G}$. Urobte podrobnejsi dokaz!

Priklad 7. Binarny kod celkovej kontroly parity dlzky 4 ma generujucu maticu

$$\mathtt{G} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Priklad 8. Nech K je ternarny kod s abecedou $Z_3 = \{0, 1, 2\}$ a dlzky 6, v ktorom 3-ti znak sluzi ku kontrole prvych dvoch a siesty znak zase ku kontrole 4. a 5. znaku, t.j. plati

$$a_1 + a_2 = a_3,\ a_4 + a_5 = a_6.$$

K je linearny, pretoze sa da opisat systemom linearnych rovnic

$$x_1 + x_2 + 2x_3 = 0$$

$$x_4 + x_5 + 2x_6 = 0$$

Tento kod ma 4 informacne znaky. Generujucou maticou je

$$\mathtt{G} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Kontrolna matica (5. lekcia)

Definicia 13. Matica $\mathtt{H}$ nad konecnym polom F sa nazyva *kontrolnou* maticou linearneho kodu $K \subseteq \mathbf{F}^n$, ak plati

$$\mathtt{H}\mathbf{v}^T = 0^T \quad \text{prave vtedy, ked} \quad \mathbf{v} \in K.$$

Inac povedane, $\mathtt{H}$ je kontrolnou maticou linearneho kodu $K \subseteq \mathbf{F}^n$, ak K je podpriestor rieseni homogenneho systemu rovnic

$$\mathtt{H}\mathbf{x}^T = 0^T.$$

(Samozrejme 0 je nulove slovo dlzky $n-k$.)

Ideme sa pozriet na vztah medzi generujucou a kontrolnou maticou jedneho a toho isteho linearneho kodu.

Definicia 14. Nech $(F, \mathbf{F}^n)$ je vektorovy priestor nad konecnym polom F. Nech $\mathbf{u}, \mathbf{v} \in \mathbf{F}^n$. Potom definujeme *skalarny* sucin vektorov (slov) nasledovne:

$$\mathbf{u}_*\mathbf{v} = u_1v_1 + \cdots + u_nv_n,$$

pricom $\mathbf{u} = u_1 \cdots u_n$, a $\mathbf{v} = v_1 \cdots v_n$.

Lema 1. Pre skalarny sucin plati:

(i) $\mathbf{u}_*\mathbf{v} = \mathbf{v}_*\mathbf{u}$;

(ii) $\mathbf{u}_*(\mathbf{v}+\mathbf{w}) = \mathbf{u}_*\mathbf{v} + \mathbf{u}_*\mathbf{w}$;

(iii) $(c\mathbf{u})_*\mathbf{v} = c(\mathbf{u}_*\mathbf{v}) = \mathbf{u}_*(c\mathbf{v})$.

Poznamka. Jedna sa o tzv. degenerovany skalarny sucin, t.j. moze platit $\mathbf{u}_*\mathbf{u} = 0$ pre nejake $\mathbf{u} \neq \mathbf{0}$. Vezmime napr. $\mathbf{u} = 0101$ nad Z_2.

Definicia 15. Nech $K \subseteq \mathbf{F}^n$ je podpriestor. Potom

$$K^\perp = \{\mathbf{v} \in \mathbf{F}^n : \mathbf{u}_*\mathbf{v} = 0 \quad \text{pre vsetky} \quad \mathbf{u} \in K\}$$

a $K^\perp$ volame *dualnym* podpriestorom ku K.

Lema 2. Nech K je k-rozmerny podpriestor $\mathbf{F}^n$. Potom $K^\perp$ je tiez podpriestorom priestoru $\mathbf{F}^n$ a plati: $\dim(K^\perp) = n - k = n - \dim(K)$.

Dokaz. Nech $\mathbf{g}_1, \cdots, \mathbf{g}_k$ je baza K. Predpokladajme, ze

$$\mathbf{g}_i = g_{i1} \cdots, g_{in}$$

je tvar tychto vektorov v $\mathbf{F}^n$ pre $i = 1, \cdots, k$. (Ak K je linearny kod, tak $\mathtt{G} = (g_{ij})$ je generujucou maticou pre K.) Potom plati:
$\mathbf{x} = x_1 \cdots x_n \in K^\perp$ prave vtedy, ked

$$(\mathbf{g}_1)_* \mathbf{x} = \cdots = (\mathbf{g}_k)_* \mathbf{x} = 0.$$

Ak prepiseme posledne vztahy, tak dostaneme homogenny linearny system rovnic

$$g_{11}x_1 + \cdots + g_{1n}x_n = 0$$

$$\cdots$$

$$g_{k1}x_1 + \cdots + g_{kn}x_n = 0.$$

Teda, $\mathbf{x} \in K^\perp$ prave vtedy, ked $\mathbf{x}$ je riesenim $\mathtt{G}\mathbf{x}^T = \mathbf{0}^T$. Pretoze $h(\mathtt{G}) = k$, tak

$$\dim(K^\perp) = n - k.$$

Zaroven sme dokazali este

Lema 3. Ak K je linearny kod s generujucou maticou $\mathtt{G}$, tak $\mathtt{G}$ je kontrolnou maticou dualneho kodu $K^{\perp}$.

Priklad 9. Nech $F = Z_2$. Nech $K = \{00000, 11111\} \subseteq \mathbf{F}^5$ je opakovaci kod. Potom $\mathbf{u} \in K^{\perp}$ prave vtedy, ked

$$u_1 + u_2 + u_3 + u_4 + u_5 = 0.$$

Teda, $K^{\perp}$ je kodom celkovej kontroly parity.

Veta 13. Dualny kod $K^{\perp}$ linearneho (n, k)-kodu K je $(n, n-k)$-kodom, t.j. $\dim(K^{\perp}) = n-k$. Dalej, generujuca matica kodu K je kontrolnou maticou kodu $K^{\perp}$ a obratene.

Dosledok. Kazdy linearny kod ma kontrolnu maticu.

Dokaz vety. Velku cast vety sme uz dokazali v lemach 1-3. Vzhladom na komutativnost skalarneho sucinu (Lema 1) vidime, ze

$$K \subseteq K^{\perp\perp}.$$

Tvrdime, ze $K = K^{\perp\perp}$. Z lemy 2 vyplyva

$$\dim(K^{\perp\perp}) = n - \dim(K^{\perp}) =$$

$$= n - (n - k) = k = \dim(K).$$

Pretoze $\dim(K) = \dim(K^{\perp\perp})$ a K je konecnorozmerny (!!), tak $K = K^{\perp\perp}$. (Preco?)

Poznamka. Upozornujeme, ze dokaz tvrdenia $K = K^{\perp\perp}$ plati len pre konecnorozmerne K. Vo vseobecnom pripade plati len $K \subseteq K^{\perp\perp}$. Kontrapriklady sa najdu vo funkcionalnej analyze.

Dokaz dosledku. Generujuca matica kodu $K^{\perp}$ je kontrolnou maticou kodu $K^{\perp\perp}$. Dokazali sme vsak $K = K^{\perp\perp}$. Teda generujuca matica kodu $K^{\perp}$ je kontrolnou maticou kodu K.

Ukazeme este, ako sa ku generujucej matici (v specialnom tvare) vyrobi odpovedajuca kontrolna matica a obratene.

Veta 14. Linearny kod K s generujucou maticou $\mathtt{G} = (\mathtt{E} \mid \mathtt{B})$ typu $k \times n$ ma kontrolnu maticu $\mathtt{H} = (-\mathtt{B}^T \mid \mathtt{E}_1)$ typu $(n-k) \times n$, kde $\mathtt{B}^T$ je transponovana matica ku $\mathtt{B}$ a $\mathtt{E}$ ci $\mathtt{E}_1$ su prislusne jednotkove matice.

Dokaz. Nech $K_0 \subseteq \mathbf{F}^n$ je podpriestor rieseni systemu homogennych linearnych rovnic

$$\mathtt{H}\mathbf{x}^T = 0^T.$$

Najprv overime

$$\mathtt{H}\mathtt{G}^T = (-\mathtt{B}^T \mid \mathtt{E}_1) \begin{pmatrix} \mathtt{E} \\ \mathtt{B}^T \end{pmatrix} =$$

$$= -\mathtt{B}^T\mathtt{E} + \mathtt{E}_1\mathtt{B}^T = -\mathtt{B}^T + \mathtt{B}^T = 0$$

pre blokove matice. Urobte si to detailne! Ukazali sme vlastne, ze riadky matice $\mathtt{G}$ su rieseniami horeuvedeneho systemu rovnic, teda riadky matice $\mathtt{G}$ patria do K_0. Pretoze K_0 je podpriestor, tak obsahuje aj vsetky linearne kombinacie riadkov matice $\mathtt{G}$. Teda $K \subseteq K_0$. Tvrdime,

ze

$$\dim(K) = \dim(K_0).$$

Vieme, ze $\dim(K) = k$, lebo $\mathtt{G}$ je typu $k \times n$. Matica $\mathtt{H}$ je typu $(n-k) \times n$, co znamena, ze $h(\mathtt{H}) = n-k$, lebo stupen $\mathtt{E}_1$ je $n-k$. Z algebry vieme, ze

$$\dim(K_0) = n - h(\mathtt{H}) = n - (n-k) = k.$$

Z rovnosti dimenzii dostaneme rovnako ako v dokaze predchadzajucej vety uz $K = K_0$, co je koniec dokazu.

Cvicenie 1. Nech φ a φ_1 su linearne (n,k)-kody nad polom F. Nech φ je systematicky a nech φ je ekvivalentny s φ_1. Nech $\mathtt{H}$ je kontrolna matica kodu φ. Ako vyzera kontrolna matica $\mathtt{H}_1$ ku kodu φ_1? (Odpoved: Nech $\pi \in S_n$ je permutacie uskutocnujuca prechod od φ ku φ_1. Pouzijeme permutaciu π na stlpce matice $\mathtt{H}$.)

Poznamka. Veta 14 a cvicenie 1 nam hovoria, ako vo vseobecnosti pocitame kontrolnu maticu ku generujucej matici. V obratenom poradi to pracuje podobne: z kontrolnej matice vyrobime generujucu.

Standardne dekodovanie (6. lekcia)

Najprv sa pozrieme na nove moznosti pri objavovani chyb prenosu sprav.

Definicia 14. Pod *Hammingovou vahou* slova $\mathbf{v} = v_1 \cdots v_n \in \mathbf{F}^n$ rozumieme pocet nenulovych znakov slova. Znacime to ako

$$\|\mathbf{v}\| = \|v_1 \cdots v_n\| = \mid \{i : v_i \neq 0\} \mid .$$

Lema 4. Nech $\varphi : A \to \mathbf{F}^n$ je linearny kod s minimalnou vzdialenostou d_φ. Potom

(i) $\rho(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|$ a

(ii) $d_\varphi = \min\{\|\mathbf{v}\| : \mathbf{v} \in \varphi(A) - \{0\}\}$.

Dokaz. Podmienka (i) je zrejma. V pripade (ii) oznacme

$$d' = \min\{\|\mathbf{v}\| : \mathbf{v} \in \varphi(A) - \{0\}\}.$$

Zrejme existuju $\mathbf{u}, \mathbf{v} \in \varphi(A)$ take, ze $d_\varphi = \rho(\mathbf{u}, \mathbf{v})$. Potom $d_\varphi = \|\mathbf{u} - \mathbf{v}\|$. Z toho vyplyva $d' \leq d_\varphi$. Obratene, nech $d' = \rho(\mathbf{w}, \mathbf{0})$ pre vhodne nenulove kodove slovo $\mathbf{w}$. Odtial, $d_\varphi \leq d'$, co dava $d_\varphi = d'$.

Veta 15. Linearny kod objavuje t-nasobne chyby prave vtedy, ked kazdych t stlpcov jeho kontrolnej matice je linearne nezavislych (=LN).

Dokaz. Predpokladajme, ze mame do cinenia s linearnym kodom $K = \varphi(A) \subseteq \mathbf{F}^n$. Dalej, nech $\mathtt{H}$ je kontrolna matica kodu K. Nakoniec,

nech d_0 je najvacsie (prirodzene) cislo t s vlastnostou: kazdych t stlpcov matice $\mathtt{H}$ je LN. Tvrdime:

$$d_\varphi = d_0 + 1.$$

Nech $\mathbf{w} \in K$ je nenulove slovo s vahou $d = d_\varphi$ (Lema 4). Potom existuju indexy

$$1 \leq i_1 < i_2 < \cdots < i_d \leq n$$

tak, ze $w_{i_j} \neq 0$ a ostatne $w_i = 0$. Z toho vyplyva

$$\mathtt{H}\mathbf{w}^T = 0^T,$$

lebo $\mathbf{w} \in K$. Inac povedane, ziskali sme

$$w_{i_1}\mathtt{h}_{i_1} + \cdots + w_{i_d}\mathtt{h}_{i_d} = 0^T,$$

kde $\mathtt{h}_{i_j}$ je i_j-ty stlpec matice $\mathtt{H}$. Nasli sme d stlpcov matice $\mathtt{H}$, ktore su LZ. Odtial vidime, ze $d_0 < d$, z coho vyplyva $d_0 + 1 \leq d$.

Kvoli jednoduchsiemu zapisu polozme

$$\partial = d_0 + 1.$$

Z definicie cisla d_0 vidime, ze existuju LZ stlpce $\mathtt{h}_{i_1}, \cdots, \mathtt{h}_{i_\partial}$ matice $\mathtt{H}$ s indexami

$$1 \leq i_1 < \cdots < i_\partial \leq n.$$

Inac povedane, existuju prvky $c_{i_1}, \cdots, c_{i_\partial} \in F$, pre ktore plati

$$c_{i_1}\mathtt{h}_{i_1} + \cdots + c_{i_\partial}\mathtt{h}_{i_\partial} = 0^T,$$

pricom aspon jedno $c_{i_j} \neq 0$. Teda, existuje take slovo $\mathbf{c} = c_1 \cdots c_n \in \mathbf{F}^n$, ze $c_i = 0$ pre $i \notin \{i_1, \cdots, i_\partial\}$ a sucasne

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

Dostali sme $\mathbf{c} \in K$. Pretoze $\mathbf{c} \neq \mathbf{0}$, tak $d_\varphi \leq \|\mathbf{c}\|$. Na druhej strane priamo vidime, ze $\|\mathbf{c}\| \leq \partial = d_0 + 1$, co dava $d_\varphi \leq d_0 + 1$. Teda $d_\varphi = d_0 + 1$, co je koniec dokazu.

Dosledok 1. Linearny kod objavuje jednoduche (dvojnasobne) chyby prave vtedy, ked kazdy

stlpec kontrolnej matice je nenulovy (ked ziaden stlpec nie je nasobkom ineho stlpca kontrolnej matice).

Dosledok 2. Nech φ je linearny kod a nech $\mathtt{H}$ je jeho kontrolna matica. Nech d_0 je najvacsie prirodzene cislo t s vlastnostou: kazdych t stlpcov matice $\mathtt{H}$ je LN. Potom

$$d_\varphi = d_0 + 1.$$

V dalsom sa budeme venovat otazkam dekodovania linearneho kodu. Stale budeme predpokladat, ze $\varphi(A) = K \subseteq \mathbf{F}^n$ je linearny (n, k)-kod nad konecnym polom F. Oznacenie

$$\mathbf{e} + K = \{\mathbf{e} + \mathbf{v} : \mathbf{v} \in K\}$$

pozname z teorie grup.

Definicia 15. Ked sa vyslalo kodove slovo $\mathbf{v} = v_1 \cdots v_n$ a prijali sme $\mathbf{w} = w_1 \cdots w_n \in \mathbf{F}^n$, tak slovo

$$\mathbf{e} = e_1 \cdots e_n = \mathbf{w} - \mathbf{v}$$

volame *chybovym* slovom (vzniklo sumom). Ekvivaletne,

$$\mathbf{w} = \mathbf{v} + \mathbf{e},$$

t.j. prijate slovo je suctom vyslaneho (kodoveho) slova a chyboveho slova.

Veta 16. Nech K je linearny (n, k)-kod nad konecnym polom F. Potom mnoziny

$$\{\mathbf{x} + K : \mathbf{x} \in \mathbf{F}^n\}$$

tvoria rozklad $\mathbf{F}^n$. Dalej mame

(i) $\mathbf{e} + K = \mathbf{e}' + K \;\Leftrightarrow\; \mathbf{e} - \mathbf{e}' \in K$;

(ii) $\mathbf{e} - \mathbf{e}' \notin K \;\Leftrightarrow\; (\mathbf{e} + K) \cap (\mathbf{e}' + K) = \emptyset$;

(iii) $|\, \mathbf{e} + K \,| = |\, K \,|$;

(iv) $\mid K \mid = q^k$, ak $\mid F \mid = q = p^r$ a

$$[\mathbf{F}^n : K] = q^{n-k}.$$

Dokaz vyplyva priamo zo znamej Lagrangeovej vety pre grupy.

Definicia 16. Nech $\varphi(A) = K \subseteq \mathbf{F}^n$ je linearny $(n, k)-$kod nad konecnym polom F. Vyberme s kazdej triedy rozkladu $\mathbf{u}+K$ slovo $\mathbf{u}'$ s najmensou vahou. Budeme ho volat *reprezentantom* triedy $\mathbf{u} + K$. (Oznacenie: $\mathbf{u}' = \text{repr}(\mathbf{u} + K)$.)

Veta 17. (Standardne dekodovanie). Nech $\varphi : A \to \mathbf{F}^n$ je linearny (n, k)-kod nad konecnym polom F. Potom zobrazenie

$$\delta : \mathbf{F}^n \to K$$

definovane predpisom

$$\delta(\mathbf{w}) = \mathbf{w} - \text{repr}(\mathbf{w} + K)$$

je dekodovanie kodu φ (volame ho standardnym).

Dokaz. Oznacme $\mathbf{w}' = \text{repr}(\mathbf{w} + K)$. Potom z vety 16(i) vidime, ze $\mathbf{w} - \mathbf{w}' \in K$, t.j. $\delta(\mathbf{w}) \in K$. Dalej, $\delta(\mathbf{w}) = \mathbf{w}$ pre $\mathbf{w} \in K$, pretoze $\mathbf{0}$ je reprezentantom tejto triedy.

Priklad 10. Zoberme (4,3)-kod celkovej kontroly parity (vid priklad 7), pricom $A = (Z_2)^3$. Potom kodovymi slovami su

$$K =$$

$$\{0000, 1001, 0101, 0011, 1111, 1100, 0110, 1010\}.$$

Zvolime nekodove slovo, napr. 1000. Potom

$$1000 + K =$$

$$\{1000, 0001, 1101, 1011, 0111, 0100, 1110, 0010\}.$$

Mame len dve triedy: K a $1000 + K$. V druhej triede sme si mohli vybrat aj ineho reprezentanta: 0001. Celu situaciu si mozeme zapisat trochu inac do tzv. Slepnianovej tabulky

0000 1001 0101 0011 1111 1100 0110 1010

0001 1000 0100 0010 1110 1101 0111 1011

V prvom riadku su uvedene kodove slova a v druhom su slova tvaru 0001+**v** z triedy 0001+K, pricom 0001+**v** lezi pod **v**. Pri tomto zapise je $\delta(\mathbf{w})$ slovo z prveho riadku v tom istom stlpci, kde sa nachadza **w**. Algoritmus prehladava 2^4 slov.

Syndromy (7. lekcia)

Zacneme este jednym poucnym prikladom.

Priklad 11. Nech K je binarny kod dlzky 6. Potom $\mathbf{u} = u_1 \cdots u_6 \in K$ prave vtedy, ked

$u_4 = u_1 + u_2$, $u_5 = u_1 + u_3$ a $u_6 = u_2 + u_3$. Inac povedane, kodove slovo **u** ma 3 informacne znaky, t.j. u_1, u_2 a u_3. Zvysne 3 znaky su kontrolne. Rychle sa presvedcime, ze K je linearny kod nad Z_2 definovany nasledovnym homogennym systemom rovnic

$$x_1 + x_2 + x_4 = 0$$

$$x_1 + x_3 + x_5 = 0$$

$$x_2 + x_3 + x_6 = 0.$$

Kontrolna matica $\mathtt{H}$ sa teraz da jednoducho urcit

$$\mathtt{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Pretoze $h(\mathtt{H}) = 3$, tak mame $2^3 = 8$ kodovych slov: 000000, 001011, 010101, 011110, 100110, 101101, 110011, 111000.

Podla vety 16 mame $2^3 = 8$ tried rozkladu aditivnej grupy $(Z_2)^3$. Ak K znamena podgrupu

kodovych slov, tak jednotlive triedy su: K, $000001+K, 000010+K, 000100+K, 001000+K$, $010000+K$, $100000+K$ a $001100+K$. Mozno vytvorit Slepnianove tabulky a dostaneme jednoduchy algoritmus na dekodovanie.

Definicia 17. Nech φ je linearny $(n,k)-$kod nad konecnym polom F s kontrolnou maticou $\mathtt{H}$. Nech dalej $\mathbf{v} \in \mathbf{F}^n$. Potom slovo $\mathtt{s} \in \mathbf{F}^{n-k}$ sa nazyva *syndromom* slova $\mathbf{v}$, ak plati

$$\mathtt{s}^T = \mathtt{H}\mathbf{v}^T.$$

Veta 18. Nech φ je linearny $(n,k)-$kod s kontrolnou maticou $\mathtt{H}$ a podpriestorom kodovych slov K.

(i) Nech dalej $\mathbf{u}, \mathbf{v} \in \mathbf{F}^n$, pricom $\mathtt{s}_1, \mathtt{s}_2 \in \mathbf{F}^{n-k}$ su odpovedajuce syndromy. Potom $\mathtt{s}_1 = \mathtt{s}_2$ prave vtedy, ked

$$\mathbf{u} + K = \mathbf{v} + K.$$

(ii) Nech $\mathbf{w} \in \mathbf{F}^n$ je prijate slovo vyslaneho kodoveho slova $\mathbf{u}$. Zrejme $\mathbf{w} = \mathbf{u} + \mathbf{e}$. Potom $\mathbf{w}$ a $\mathbf{e}$ maju rovnake syndromy.

Dokaz. (i) Zrejme, $\mathtt{H}\mathbf{u}^T = 0^T$ prave vtedy, ked $\mathbf{u} \in K$. Druha vec, ktoru potrebujeme, je

$$\mathtt{H}(\mathbf{u} + \mathbf{v})^T = \mathtt{H}\mathbf{u}^T + \mathtt{H}\mathbf{v}^T.$$

(ii) Pretoze $\mathbf{w} \in \mathbf{e} + K$, tak z (i) dostavame tvrdenie.

Dosledok. Prijate slovo ma ten isty syndrom ako chybove slovo.

Definicia 18. Hovorime, ze linearny kod φ pri dekodovani δ *opravuje chybove slovo* $\mathbf{e}$, ak plati

$$\delta(\mathbf{e} + \mathbf{v}) = \mathbf{v} \quad \text{pre kazde kodove slovo } \mathbf{v}.$$

Veta 19. Nech φ je linearny kod s minimalnou vzdialenostou $d_\varphi = d$. Potom standardne dekodovanie opravi $t-$nasobne chyby pre vsetky $t < d/2$.

Dokaz. Predpokladame, ze sme vyslali kodove slovo $\mathbf{u}$ a prijali sme slovo $\mathbf{w}$. Dalej predpokladame, ze plati $\rho(\mathbf{u}, \mathbf{w}) = t < d/2$. Potrebujeme dokazat, ze $\rho(\mathbf{u}, \mathbf{w}) < \rho(\mathbf{x}, \mathbf{w})$ pre vsetky kodove slova $\mathbf{x} \neq \mathbf{u}$. Pretoze φ je linearny kod, tak vieme, ze $\mathbf{w} = \mathbf{u} + \mathbf{e}$, kde $\mathbf{e}$ je chybove slovo. Potom

$$\rho(\mathbf{u}, \mathbf{w}) = \|\mathbf{w} - \mathbf{u}\| = \|\mathbf{e}\| = t.$$

Zoberme kodove slovo $\mathbf{x} \neq \mathbf{u}$. Teraz

$$\rho(\mathbf{x}, \mathbf{w}) = \|\mathbf{w} - \mathbf{x}\| = \|(\mathbf{u} - \mathbf{x}) + \mathbf{e}\|.$$

Lenze $\|\mathbf{u} - \mathbf{x}\| \geq d$, lebo su to rozne kodove slova. Z toho teraz vyplyva, ze

$$\rho(\mathbf{x}, \mathbf{w}) = \|(\mathbf{u} - \mathbf{x}) + \mathbf{e}\| \geq d/2$$

a dokaz je hotovy.

Veta 20. Standardne dekodovanie opravuje prave tie chybove slova, ktore sme zvolili za reprezentantov tried. Navyse, standardne dekodovanie δ je optimalne v tom zmysle, ze ziadne ine dekodovanie neopravuje vacsiu mnozinu chybovych slov ako δ.

Dokaz. Nech $\mathbf{e}$ je reprezentantom svojej triedy $\mathbf{e} + K$. Nech $\mathbf{v}$ je kodove slovo a nech δ je standardne dekodovanie. Potom

$$\delta(\mathbf{e} + \mathbf{v}) = \mathbf{e} + \mathbf{v} - \mathbf{e} = \mathbf{v}.$$

Dalej, nech $\mathbf{e}'$ nie je reprezentantom triedy $\mathbf{e}' + K$. Je nim $\mathbf{e}$. Potom vsak

$$\delta(\mathbf{e}' - \mathbf{e}) = \mathbf{e}' - \mathbf{e} \neq \mathbf{0},$$

lebo $\mathbf{e}' - \mathbf{e}$ je kodove slovo (veta 16). Predpokladajme, ze by δ opravilo aj chybove slovo $\mathbf{e}'$. Z toho dostaneme

$$\delta(\mathbf{e}') = \mathbf{e}' - \mathbf{e} \neq \mathbf{0}.$$

Pretoze $\mathbf{0}$ je kodove slovo, tak

$$\delta(\mathbf{e}' + \mathbf{0}) = \mathbf{0} = \delta(\mathbf{e}') = \mathbf{e}' - \mathbf{e} \neq \mathbf{0},$$

co je spor. Dokazali sme prvu cast vety.

Predpokladajme, ze δ^* je dalsie dekodovanie a nech δ^* opravuje vsetky tie chybove slova, co robi δ, t.j. δ^* je vsade tam definovane, kde aj δ, pricom na spolocnom definicnom obore nadobudaju rovnake hodnoty. Nech $\mathbf{e}' \in \mathbf{e} + K$ a predpokladajme, ze $\mathbf{e}$ je reprezentantom svojej triedy, pricom $\mathbf{e} \neq \mathbf{e}'$. Zrejme $\mathbf{e}' - \mathbf{e} \in K$, lebo su z jednej triedy. Dostaneme

$$\delta^*(\mathbf{e}') = \delta(\mathbf{e}') = \mathbf{e}' - \mathbf{e} \neq \mathbf{0}.$$

Dalej postupujme nepriamo. Predpokladajme, ze by δ^* bolo lepsie ako δ, t.j. ze δ^* opravi aj chybove slovo $\mathbf{e}'$. Z toho vyplyva

$$\delta^*(\mathbf{e}') = \delta^*(\mathbf{e}' + \mathbf{0}) = \mathbf{0},$$

co je spor, lebo vyssie nam vyslo $\delta^*(\mathbf{e}') \neq \mathbf{0}$.

Poznamka. Urobime strucny zaver o standardnom dekodovani pomocou syndromov. Pred dekodovanim si pripravime zoznam reprezentatov $\mathbf{e}_0, \cdots, \mathbf{e}_{\partial-1}$, kde $\partial = q^{n-k}$ je pocet tried rozkladu $\mathbf{F}^n$ podla K (veta 16). K tymto slovam si urcime syndromy $\mathbf{s}_0, \cdots, \mathbf{s}_{\partial-1}$. Potom, ak prijmeme slovo $\mathbf{w} \in \mathbf{F}^n$, tak vypocitame jeho syndrom, povedzme $\mathbf{s}_\triangle$. Vyhladame v nasom (pripravenom) zozname syndromov syndrom $\mathbf{s}_j$, pre ktory plati

$$\mathbf{s}_j = \mathbf{s}_\triangle.$$

(Ten je jednoznacne urceny - vid vetu 17.) Vyberieme reprezentanta $\mathbf{e}_j = \text{repr}(\mathbf{w} + K)$ a mame vysledne slovo

$$\delta(\mathbf{w}) = \mathbf{w} - \mathbf{e}_j.$$

Pomocou syndromov sa standardne dekodovanie vykona rychlejsie ako pomocou Slepnianovej tabulky. V prvom pripade sme prehladavali zoznam s q^{n-k} prvkami, v druhom pripade je to

uz q^n prvkov. Vratme sa kratko este k prikladu 11. Pouzijeme metodu syndromov. Tam sme uz nasli reprezentantov tried: $\mathbf{e}_0 = 000000$, $\mathbf{e}_1 = 100000$, $\mathbf{e}_2 = 010000$, $\mathbf{e}_3 = 001000$, $\mathbf{e}_4 = 000100$, $\mathbf{e}_5 = 000010$, $\mathbf{e}_6 = 000001$, $\mathbf{e}_7 = 001100$. Odpovedajuce syndromy rychle vypocitame: $\mathtt{s}_0 = 000$, $\mathtt{s}_1 = 110 = \mathtt{h}_1^T$, $\mathtt{s}_2 = 101 = \mathtt{h}_2^T$, $\mathtt{s}_3 = 011 = \mathtt{h}_3^T$, $\mathtt{s}_4 = 100 = \mathtt{h}_4^T$, $\mathtt{s}_5 = 010 = \mathtt{h}_5^T$, $\mathtt{s}_6 = 001 = \mathtt{h}_6^T$, $\mathtt{s}_7 = 111 = \mathtt{h}_3^T + \mathtt{h}_4^T$, kde $\mathtt{h}_i$ znamena $i-$ty stlpec kontrolnej matice $\mathtt{H}$. Teraz, ak prijmeme slovo $\mathbf{w} = 100100$, tak $\mathtt{H}\mathbf{w}^T = \mathtt{s}_{\triangle}^T$, co dava

$$\mathtt{s}_{\triangle} = 010 = \mathtt{h}_1^T + \mathtt{h}_4^T = \mathtt{s}_5.$$

Teda,

$$\delta(\mathbf{w}) = \mathbf{w} + \mathbf{e}_5 = 100100 + 000010 = 100110.$$

Hammingove kody. (8. lekcia)

Zacneme prikladom Hammingovho (7,4)-kodu. Je to binarny kod definovany kontrolnou maticou

$$\mathtt{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

kde stlpce su binarnym rozvojom cisel $1, \cdots, 7$. Napr. $1{=}0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$ alebo $7{=}\ 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$. Pretoze $h(\mathtt{H}) = 3$, tak $4{=}7\text{-}3$ je pocet informacnych znakov. Preto $\mid K \mid = 2^4$. Podla vety 16 mame $2^{7-4} = 2^3 = 8$ tried rozkladu $\mathbf{Z}_2^n$ podla K, ako aj syndromov.

Vyhladame reprezentantov. Zrejme je $\mathbf{e}_0 = 0000000$ taky. Dalej postupujeme tak, ze preskusame vsetky slova vahy 1, potom slova vahy 2 atd. Zaroven vypocitame k nim aj syndromy. Dohodneme sa, ze $\mathbf{e}_i$ pre $i \geq 1$ bude znamenat slovo dlzky n, ktore ma na $i-$tom mieste znak 1 a inac 0. Pre nas je teraz $n = 7$.

Vzhladom na tvar slov $\mathbf{e}_i$ vidime, ze patricne syndromy su $\mathtt{s}_i^T = \mathtt{H}\mathbf{e}_i^T$. Potom

$$\mathtt{s}_i = \mathtt{h}_i^T,$$

pre $i = 1, \cdots, n$, kde $\mathtt{h}_i$ znamena $i-$ty stlpec matice $\mathtt{H}$. Zrejme $\mathtt{s}_0 = 000$. Zaverom vidime, ze $\mathtt{s}_i \neq \mathtt{s}_j$ pre $i \neq j$. Mame reprezentantov a odpovedajuce syndromy. Doslo k dvom zjednoduseniam: reprezentanti su nulove slovo a jednotkove slova $\mathbf{e}_i$, pricom odpovedajuce syndromy $\mathtt{s}_i$ su binomicke rozvoje cisel $i = 0, \cdots, 7$. To znamena, ze aj dekodovanie sa zjednodusilo: ak prijate slovo $\mathbf{w} = w_1 \cdots w_7$ ma syndrom $\mathtt{s}_\triangle = \mathtt{s}_i$, tak

$$\delta(\mathbf{w}) = \mathbf{w} + \mathbf{e}_i.$$

(V prijatom slove sme zmenili $i-$ty znak, ak $i \geq 1$. Pre $i = 0$ sa nic nemeni.) Konkretnejsie, ak $\mathbf{w} = 1010111$, tak $\mathtt{H}\mathbf{w}^T = (110)^T = \mathtt{s}_6^T$. Teda opravime 6-ty znak na prijatom slove: $\delta(\mathbf{w}) = 1010101$.

Veta 21. Binarny linearny kod opravuje jednoduche chyby prave vtedy, ked stlpce jeho kontrolnej matice su nenulove a navzajom rozne.

Dokaz. (Uvedieme dokonca dva dokazy tohto tvrdenia. Obidva su zaujimave.) Vieme, ze nas kod K opravuje jednoduche chyby prave vtedy, ked jeho minimalna vzdialenost je aspon 3. Nech teda K opravuje jednoduche chyby. Ak by $i-$ty stlpec kontrolnej matice $\mathtt{H}$ bol nulovy, tak $\mathbf{e}_i \in K$, co by bol spor, lebo vychadza, ze minimalna vaha kodu K sa rovna 1. Dalej, nech $\mathtt{H}$ ma zhodne dva stlpce, povedzme $i-$ty a $j-$ty pre $i \neq j$. Uvazujme o slove

$$\mathbf{e}_{ij} = 0 \cdots 010 \cdots 010 \cdots 0$$

(znak 1 mame na $i-$tom a $j-$tom mieste). Potom $\mathbf{e}_{ij} \in K$, co dava znova spor, lebo vychadza, ze minimalna vaha kodovych slov je 2.

Obratene, nech $\mathtt{H}$ ma hore uvedene vlastnosti. Ukazeme, ze ziadne slovo vahy 1 a 2 nie je kodove. Takymi slovami su len slova tvaru $\mathbf{e}_i$ a $\mathbf{e}_{ij}$. Potom mame

$$\mathtt{H}\mathbf{e}_i^T = \mathtt{h}_i \neq 0^T$$

a

$$\mathtt{H}\mathbf{e}_{ij}^T = \mathtt{h}_i + \mathtt{h}_j \neq 0^T.$$

Vysledok: Minimalna vzdialenost slov z K je aspon 3.

Druhy dokaz vyplyva z dosledku 1 vety 15: ziaden stlpec kontrolnej matice nie je nasobkom ineho stlpca. V pripade binarne kodu to znamena: 0-nasobok alebo 1-nasobok. Teda nulovy stlpec alebo dva rovnake stlpce. To je vsak vylucene.

Definicia 19. Binarny kod sa vola *Hammingovym*, ak ma kontrolnu maticu, ktorej stlpce

su vsetky nenulove slova danej dlzky a ziadne z nich sa neopakuje.

Poznamka. Mame nekonecne vela Hammingovych matic. Ich typy su $m \times (2^m - 1)$ pre $m = 2, 3, \cdots$. Tak dostavame binarne (3,1)-kody, (7,4)-kody, ci (15,11)-kody atd.

Priklad 12. Pre $m = 1$ dostaneme trivialnu maticu $\mathtt{H} = (1)$. Zoberme preto $m = 2$ a Hammingovou maticou bude

$$\mathtt{H} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

a hociktora ina matica, ktora vznikne z $\mathtt{H}$ permutaciou stlpcov. V kazdom pripade sa jedna len o jeden (3,1)-kod, ktory sa da opisat rovnicami (alebo ekvivalentnou sustavou rovnic)

$$x_2 + x_3 = 0$$

$$x_1 + x_3 = 0.$$

Dava to opakovaci kod dlzky 3, ktory opravuje jednoduche chyby.

Priklad 20. Pre $m = 3$ ziskame napr. nasu znamu maticu

$$\mathtt{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Ak na $\mathtt{H}$ urobime permutacie stlpcov, tak dostaneme (vsetky) Hammingove matice pre $m = 3$. Davaju navzajom ekvivalentne kody. Dalej hladame ku $\mathtt{H}$ generujucu maticu. Pouzijeme vetu 14. Najprv urobime na $\mathtt{H}$ taku permutaciu stlpcov, aby sme dostali maticu tvaru

$$\mathtt{H}' = (\mathtt{B}^T \mid \mathtt{E}).$$

Potrebujeme urobit na stlpcoch nasledovne transpozicie: $1 \leftrightarrow 7$, $2 \leftrightarrow 6$ a $4 \leftrightarrow 5$. Dostaneme

$$\mathtt{H}' = \left(\begin{array}{cccc|ccc} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right).$$

Tomu odpovedajuca generujuca matica bude

$$\mathtt{G}' = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right) = (\mathtt{E}_1 \mid \mathtt{B}).$$

Spatne prehodenie stlpcov dava generujucu maticu $\mathtt{G}$ ku povodnej kontrolnej matici $\mathtt{H}$. Dostaneme

$$\mathtt{G} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Priklad 21. Nech $m = 4$. Obdrzime (jeden) Hammingov (15,11)-kod s kontrolnou maticou

$$\mathtt{H} = \begin{pmatrix} 000000011111111 \\ 000111100001111 \\ 011001100110011 \\ 101010101010101 \end{pmatrix}.$$

Dalsie kontrolne matice ziskame permutaciou stlpcov, co davaju navzajom ekvivalentne kody.

Standardne dekodovanie Hammingovho kodu je jednoduche. Predpokladajme pripad, ze $i-$ty stlpec $\mathtt{h}_i$ kontrolnej matice $\mathtt{H}$ dava binarny rozklad cisla $i \in \{1, \cdots, 2^m - 1\}$. Nech $\mathbf{e}_i$ su jednotkove slova dlzky $2^m - 1$, t.j. $\mathbf{e}_0 = 0 \cdots 0$ a $\mathbf{e}_i$ ma jeden znak 1 na $i-$tom mieste a inac 0 pre $i = 1, \cdots, 2^m - 1$. Potom plati:

$$\mathtt{s}_i^T = \mathtt{H}\mathbf{e}_i^T = \mathtt{h}_i \quad \text{pre} \quad i = 1, \cdots, 2^m - 1.$$

Z vlastnosti matice $\mathtt{H}$ vidime, ze slova

$$\mathbf{e}_0, \quad \mathbf{e}_1, \quad \cdots, \quad \mathbf{e}_{2^m-1}$$

tvoria mnozinu reprezentantov nasho kodu. Jednotlive syndromy su potom

$$\mathtt{s}_0 = 0, \quad \mathtt{s}_1 = \mathtt{h}_1^T, \quad \cdots, \quad \mathtt{s}_{2^m-1} = \mathtt{h}_{2^m-1}^T.$$

Nech $\mathtt{s}_\triangle^T = \mathtt{H}\mathbf{w}^T$. Dalej, nech pre syndromy plati

$$\mathtt{s}_\triangle = \mathtt{s}_i.$$

Potom pre dekodovanie mame

$$\delta(\mathbf{w}) = \mathbf{w} + \mathbf{e}_i.$$

Veta 22. Standardne dekodovanie Hammingovho kodu je spravne v pripade jednoduchej chyby, t.j. ked prijate slovo **w** sa lisi od vyslaneho kodoveho slova **u** nanajvys v jednom znaku.

Dokaz vyplyva z vety 19.

Perfektne kody. (9. lekcia)

Definicia 20. Linearny kod je *perfektny* pre $t-$nasobne opravy, ak mnozina vsetkych slov vahy $\leq t$ tvori system reprezentantov jeho tried.

Lema 5. Nech $\mathtt{H}$ je kontrolna matice Hammingovho kodu typu $m \times 2^m - 1$. Potom $h(\mathtt{H}) = m$.

Dokaz. Pre $m = 1$ je to pravda. V dalsom postupujme nepriamo. Pretoze $m < 2^m - 1$ pre $m \geq 2$, tak vidime, ze $\ell = h(\mathtt{H}) < m$. teda existuje ℓ riadkov matice $\mathtt{H}$, ktore su LN a kazdy

dalsi riadok je ich linearnou kombinaciou. Bez ujmy na vseobecnosti mozeme predpokladat, ze sa jedna o prvych ℓ riadkov nasej matice. Kazdy dalsi riadok je uz linearnou kombinaciou prvych ℓ riadkov. Vzhladom na to, ze

$$2^\ell < 2^m - 1,$$

tak v prvych ℓ riadkoch mame bud nulovy stlpec alebo dva rovnake stlpce. Tvrdime, ze nulovy stlpec v prvych ℓ riadkoch je nulovym stlpcom v celej matici. Totiz linearnou kombinaciou nul dostaneme zase len nuly. Podobne je to aj s rovnakymi stlpcami v prvych ℓ riadkoch. Tie budu rovnake v celej matici vdaka linearnej kombinacii. To je vsak spor s konstrukciou kontrolnej matice Hammingovho kodu. Teda $h(\mathtt{H}) = m$.

Lema 6. Minimalna vzdialenost Hammingovho kodu pre $m \geq 2$ je 3.

Dokaz. Z vety 21 vieme, ze lubovolny Hammingov kod opravuje len jednoduche chyby. Preto minimalna vzdialenost Hammingovho kodu je ≥ 3. Nech K je podpriestor kodovych slov nejakeho Hammingovho kodu pre $m \geq 2$. Uvazujme o $\mathbf{e}_{ij} \in \mathbf{Z}_2^n$. Podla vety 16 plati

$$\mathbf{e}_{ij} \in \mathbf{e}_k + K$$

pre nejake $\mathbf{e}_k$ (pozri tiez uvahy tesne pred vetou 22). Preto

$$\mathbf{e}_{ij} = \mathbf{e}_k + \mathbf{v}$$

pre vhodne kodove slovo $\mathbf{v}$. Pretoze $\| \mathbf{e}_{ij} \| = 2$, tak nemoze platit $\| \mathbf{v} \| \geq 4$, lebo $\| \mathbf{e}_k \| = 1$. Ostava $\| \mathbf{v} \| = 3$, co dava nase tvrdenie.

Poznamka. Pre $m = 1$ je minimalna vzdialenost $= 1$.

Veta 23. Binarny kod je perfektny pre jednoduche opravy prave vtedy, ked je Hammingov.

Dokaz. Nech K je Hammingov kod dlzky $n = 2^m - 1$. Podla lemy 5 je vsak $h(\mathtt{H}) = m$ a preto

$$k = n - h(\mathtt{H}) = 2^m - m - 1$$

je pocet informacnych znakov kodu K a m je jeho pocet kontrolnych znakov. Teda K je linearny $(n, k)-$kod, kde $n = 2^m - 1$ a $k = 2^m - m - 1$. Potom $\mid K \mid = 2^k$ a pocet tried rozkladu $\mathbf{Z}_2^n$ podla K je $2^{n-k} = 2^m$ (veta 16). Tvrdime, ze reprezentantmi tychto tried su jednotkove vektory (slova)

$$\mathbf{e}_0, \quad \mathbf{e}_1, \quad \cdots, \quad \mathbf{e}_{2^m-1}.$$

Kazde take slovo lezi v jednej triede rozkladu $\mathbf{u} + K$. Najprv ukazeme, ze ziadne dve rozne slova z nasho zoznamu nelezia v jednej a tej istej triede. Povedzme, ze by to nebola pravda. Potom by sme mali pre syndromy

$$\mathtt{H}\mathbf{e}_i^T = \mathtt{H}\mathbf{e}_j^T$$

pre nejake $i \neq j$. To je ekvivalentne s tvrdenim $\mathtt{H}(\mathbf{e}_i - \mathbf{e}_j)^T = 0^T$. To by znamenalo, ze $\mathbf{e}_{ij} =$

$\mathbf{e}_i - \mathbf{e}_j$ by bolo kodovym slovom, co vsak nie je pravda, pretoze $\| \mathbf{e}_{ij} \| = 2$ a minimalna vaha K je 3 (vid lemu 6). Teda horeuvedeny zoznam slov predstavuje jedinych reprezentantov vahy ≤ 1 a dlzky slov n.

Obratene, nech K je perfektny binarny linearny kod dlzky n na opravovanie jednoduchych chyb. Nech K ma m kontrolnych znakov. Potom K ma kontrolnu maticu H typu $m \times n$. Podla vety 21 su stlpce matice H navzajom rozne a nenulove. Preto $n \leq 2^m - 1$. Na druhej strane je index

$$[\mathbf{Z}_2^n : K] = n + 1,$$

co je pocet reprezentantov tried, t.j. slov vahy 0 a 1. Z vety 16 vieme este, ze

$$n + 1 = 2^{n-k} = 2^m.$$

Teda $n = 2^m - 1$. Zaverom vidime, ze H obsahuje stlpce, co su vsetky nenulove a navzajom rozne binarne slova dlzky m. Teda H definuje kontrolnu maticu Hammingovho kodu.

Poznamka 1. Zvykne sa pouzivat aj tzv. rozsireny Hammingov kod. Urobi sa to podobne, ako sme to uz robili v priklade 1. Vystartujeme z nejakeho Hammingovho kodu, t.j. binarneho linearneho $(2^m - 1, 2^m - m - 1)-$kodu

$$\varphi : A \to \mathbf{Z}_2^n$$

pre $n = 2^m - 1$. Novy kod bude mat tvar

$$\psi : A \to \mathbf{Z}_2^{n+1}$$

a kodove slova definujeme nasledovne: Ak $\varphi(a) = v_1 \cdots v_n \in \varphi(A)$, tak

$$\psi(a) = v_1 \cdots v_n v_{n+1},$$

pricom plati

$$v_1 + \cdots + v_n + v_{n+1} = 0.$$

Rychle sa vidi, ze tento novy kod je tiez binarny, linearny $(2^m, 2^m - m - 1)-$kod, pretoze Hammingov kod je tiez taky. Jedna sa vlastne o akysi kod celkovej kontroly parity (vid priklad 1). Nakoniec, tvrdime, ze minimalna vzdia-

lenost (vaha) rozsireneho Hammingovho kodu je $= 4$. Naozaj, nech

$$\varphi(a) = \mathbf{v} = v_1 \cdots v_n$$

je take kodove slovo, ze plati $\| \mathbf{v} \| = 3$ (vid lema 6). Potom pre slovo $\psi(a) = v_1 \cdots v_n v_{n+1}$ dostaneme $\| \psi(a) \| = 4$, z coho vyplyva nase tvrdenie. Zaverom vidime, ze rozsireny Hammingov kod opravuje jednoduche chyby a objavuje 3-nasobne chyby.

Poznamka 2. Vsimnime si binarny opakovaci kod dlzky $2n+1$. Vieme, ze je to linearny $(2n+1, 1)-$kod s generujucou maticou

$$\mathtt{G} = (11 \cdots 1)$$

typu $1 \times 2n+1$. Jeho kontrolna matica ma tvar

$$\mathtt{H} = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 1 \\ & & \cdots & & \\ 0 & 0 & \cdots & 1 & 1 \end{pmatrix}$$

typu $2n \times 2n+1$. Zrejme $K = \{00\cdots0, 11\cdots1\}$. Teda $\mid K \mid = 2$. Nech $\mathbf{e} \in \mathbf{Z}_2^{2n+1}$. Potom

$$\mathbf{e} + K = \{\mathbf{e}, \mathbf{e}'\},$$

kde $\mathbf{e} + \mathbf{e}' = 11\cdots1$. Hned vidime, ze tried je tolko, kolko je prvkov vahy $\leq n$, lebo bud $\| \mathbf{e} \| \leq n$ alebo $\| \mathbf{e}' \| \leq n$.

Veta 24. (Tietavainenon, Van Lint) Jedine netrivialne perfektne binarne kody su tieto:

- Hammingove kody pre jednoduche chyby,
- Golayov kod pre trojnasobne chyby a kody s nim ekvivalentne,
- Opakovacie kody dlzky $2n + 1$, kde $n = 1, \cdots,$ pre $n-$nasobne chyby.

Dokaz. Je hodne kombinatoricky. Neurobime ho.

Opiseme jedine Golayov linearny kod. Uvedieme jeho generujucu maticu $\mathtt{G}_{23}$ typu 12×23. Je to

$$\mathtt{G}_{23} = (\mathtt{E}_{12} \mid \mathtt{C}),$$

kde $\mathtt{C}$ je typu 12×11 a plati

$$\mathtt{C} = \left(\begin{array}{c} \mathtt{B} \\ \hline 1 \quad 1 \quad \cdots \quad 1 \quad 1 \end{array} \right).$$

Podmatica $\mathtt{B}$ je stupna 11 a vznikne cyklickymi posuvmi prveho riadku: 11011100010, t.j.

$$\mathtt{B} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ & & & & \cdots & & & & & & \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Prvy riadok $\mathtt{B}$ ma znak 1 na mieste $i = 0, 1, \cdots, 10$, prave vtedy, ked i je stvorcom mod 11. Su to

miesta: $0^2 = 0$, $1^2 = 1$, $2^2 = 4$, $3^2 = 9$, $4^2 = 5$ a $5^2 = 3$ pocitane v Z_{11}.

Cyklicke kody. (Uvod.) (10. lekcia)

Definicia 21. Linearny kod $K \subseteq \mathbf{F}^n$ volame *cyklickym,* ak $v_0 v_1 \cdots v_{n-1} \in K$ implikuje $v_{n-1} v_0 \cdots v_{n-2} \in K$.

Doteraz sme vychadzali z toho, ze K bol podpriestorom vektoroveho priestoru $(F; \mathbf{F}^n)$. Dokonca sme zistili, ze $K \cong \mathbf{F}^k$, kde $k = \dim(K)$ (vid poznamku za vetou 10). Najprv ukazeme, ze $\mathbf{F}^n$ je bohatsou strukturou, totiz *linearnou algebrou*.

Zacneme vyhodnejsim zapisom slov linearneho kodu. Nech $F[x]$ znamena okruh polynomov v neurcitej x nad (konecnym) polom F. Odteraz bude $\mathbf{F}^n$ este znamenat

$$\mathbf{F}^n = \{f(x) \in F[x] : \mathrm{st} f(x) < n\}.$$

Potom mame zobrazenie

$$\beta : K \to \mathbf{F}^n,$$

s predpisom

$$\beta : v_0 v_1 \cdots v_{n-1} \mapsto v_0 + v_1 x + \cdots + v_{n-1} x^{n-1}.$$

Navyse, zobrazenie β je proste. (Preco?)

Veta 25. Okruh polynomov $F[x]$ je hlavny, t.j. kazdy ideal z $F[x]$ je hlavny.

Veta 26. Nech $g(x) \in F[x]$. Potom ideal $(g(x))$ generuje okruhovu kongruenciu mod$(g(x))$ na $F[x]$ tak, ze plati:

$$f \equiv h \ (\mathrm{mod}(g(x))) \quad \Leftrightarrow \quad f - h \in (g(x)).$$

Dalej, ak st $g(x) = n \geq 1$, tak pre kazde $f \in F[x]$ existuje $r(x) \in F[x]$ s vlastnostou, ze st $r(x) < n$ a plati

$$f \equiv r \ (\mathrm{mod}(g(x))).$$

Navyse, polynom $r = r(x)$ s horeuvedenou vlastnostou je ku f jednoznacne urceny.

Dokaz. Prva cast vety je znama zo vseobecnej teorie. V druhej casti po deleni so zvyskom

$$f(x) = g(x)q(x) + r(x),$$

pricom st$r$ $<$ st$g$ $= n$ a polynomy q a r su jednoznacne urcene. Zaverom dostavame

$$f \equiv r \ (\mathrm{mod}(g(x))).$$

Dosledok. Nech su splnene predpoklady vety 26. Potom existuje epimorfizmus

$$\varphi : F[x] \to F[x]/(g(x))$$

definovany predpisom

$$\varphi : f(x) \mapsto f(x) + (g(x)) = r(x) + (g(x)),$$

kde st$r(x) < n$ a $r(x)$ je jednoznacne ku $f(x)$ urceny.

Dokaz vyplyva zo vseobecnej vety o epimorfizme ku faktorovemu okruhu a z predchadzajucej vety.

Veta 27. Nech $g(x) \in F[x]$, kde st$g = n$. Nech $\mathbf{F}^n$ je mnozina vsetkych polynomov $f(x) \in F[x]$ stupna $< n$. Potom $(\mathbf{F}^n; +, *)$ je okruhom, ak $+$ definujeme ako obycajne scitovanie polynomov v $F[x]$ a $f * h$ je nasobenie mod$(g(x))$, t.j.

$$f * h = r \in \mathbf{F}^n,$$

ak v okruhu $F[x]$ plati

$$f(x)h(x) = g(x)q(x) + r(x),$$

pricom st$r < n$. Navyse, $\mathbf{F}^n \cong F[x]/(g(x))$.

Nacrt dokazu. Tvrdenie plati na zaklade vety 26. Priradenie

$$c_0 + \cdots + c_{n-1}x^{n-1} \mapsto c_0 + \cdots + c_{n-1}x^{n-1} + (g(x))$$

je bijekciou medzi $\mathbf{F}^n$ a $F[x]/(g(x))$. Rychle sa overi, ze je to dokonca izomorfizmus vzhladom

na $+$ a $*$. Preto je $(\mathbf{F}^n; +, *)$ okruhom. Urobte detailnejsi dokaz!

Dosledok. Nech su splnene predpoklady vety 27. Nech $f(x), h(x) \in \mathbf{F}^n$. Predpokladajme, ze st$f(x)$+st$h(x) < n$. Potom

$$f(x) * h(x) = f(x)h(x).$$

Definicia 22. Nech $(F; V)$ je nasledovna algebraicka struktura

1. $(F; V)$ je vektorovy priestor nad polom F,

2. $V = (V; +, \cdot)$ je (komutativny) okruh a

3. pre $c \in F$ a $\mathbf{x}, \mathbf{y} \in V$ plati

$$c(\mathbf{xy}) = (c\mathbf{x})\mathbf{y} = \mathbf{x}(c\mathbf{y}).$$

Potom $(F; V)$, resp. len V, volame *linearnou algebrou* nad polom F.

Poznamka. S pojmom linearna algebra, alebo len kratko algebra, su automaticky definovane aj pojmy (linearnej) podalgebry, homomorfizmu ci kongruencie a dalsie odvodene pojmy.

Priklad 22. Nech F je pole. Potom dvojica $(F; F)$ je linearnou algebrou, dokonca jednorozmernou.

Priklad 23. Nech C znamena pole komplexnych cisel. Potom $(R; C)$ je dvojrozmerna linearna algebra, ak R znamena pole realnych cisel.

Priklad 24. Nech $F[x]$ je okruh polynomov v neurcitej x nad polom F. Potom dvojica $(F; F[x])$ je linearnou algebrou nad polom F, ktora je uz nekonecne dimenzionalna.

Pre nas je dolezity nasledovny priklad

Priklad 25. Nech F je (konecne) pole a nech $\mathbf{F}^n$ znamena okruh z vety 27. Potom $(F; \mathbf{F}^n)$ je $n-$rozmerna linearna algebra s bazou $1, x, \cdots, x^{n-1}$.

V dalsich uvahach budeme stale pouzivat polynom $g(x) = x^n - 1$. Navyse zavedieme este

Definicia 23. Nech F je konecne pole. Potom

$$\mathbf{F}^n = (F; \mathbf{F}^n)$$

bude znamenat linearnu algebru *zvyskovych tried polynomov stupna* $< n$ z prikladu 25 a vety 27 pre $g(x) = x^n - 1$.

Lema 7. Nech $(\mathbf{F}^n; +, *) \cong F[x]/(x^n - 1)$. Potom v okruhu $\mathbf{F}^n$ plati

$$x^{*k} = \begin{cases} x^k, & \text{ak } 0 \leq k < n; \\ 1, & \text{ak } k = n, \end{cases}$$

kde x^{*k} znamena $k-$tu mocninu v $\mathbf{F}^n$, t.j. $x * \cdots * x$ ($k-$krat). Dalej,

$$x^{*(n+k)} = x^{*k}.$$

Dokaz. Potrebujeme len vybavit x^{*n}. Treba najst $r = r(x) \in \mathbf{F}^n$, pre ktory plati

$$x^n \equiv r \ (\mathrm{mod}(x^n - 1))$$

zase v okruhu $F[x]$. Vzhladom na vztah $x^n = (x^n - 1) + 1$ vidime, ze $r(x) = 1$. Teda $x^{*n} = 1$. Ostatne vztahy su jednoduche. Urobte to!

Vzhladom na priklad 25 mozeme sa na linearny kod $K \subseteq \mathbf{F}^n$ divat ako na podmnozinu linearnej algebry $(F; \mathbf{F}^n)$, t.j. okruhu $\mathbf{F}^n$ a zaroven vektoroveho priestoru. Preto nasledovna veta dava zmysel.

Veta 28. Nech F je konecne pole. Potom linearny kod $K \subseteq \mathbf{F}^n$ je cyklicky prave vtedy, ked K je idealom okruhu

$$(\mathbf{F}^n; +, *) \cong F[x]/(x^n - 1).$$

Dokaz. Nech $K \subseteq \mathbf{F}^n$ je cyklicky kod. Potrebujeme ukazat, ze K je idealom v okruhu $\mathbf{F}^n$. Zrejme, $f, h \in K$ implikuje $f - h \in K$, pretoze K je linearny. Ostava este dokazat, ze

$$K * h = h * K \subseteq K$$

pre lubovolne $h \in \mathbf{F}^n$. Urobime to postupne. Predpokladajme, ze

$$f(x) = v_0 + \cdots + v_{n-1}x^{n-1} \in K.$$

Polozme najprv $h(x) = x$. Potom

$$x * (v_0 + \cdots + v_{n-2}x^{n-2} + v_{n-1}x^{n-1}) =$$

$$= v_{n-1} + v_0 x + \cdots + v_{n-2}x^{n-1}$$

podla lemy 7. Dostali sme

$$x * K = K * x \subseteq K,$$

lebo operacia $*$ je komutativna a K je cyklicky. Indukciou ziskame $x^i * K \subseteq K$ pre kazde $i = 1, \cdots, n-1$. Zoberme nakoniec

$$h(x) = u_0 + \cdots + u_{n-1}x^{n-1} \in \mathbf{F}^n.$$

Potom

$$h(x) * f(x) =$$

$$\begin{aligned} &= (u_0 * f(x)) + \cdots + (u_{n-1}x^{n-1} * f(x)) \\ &= (u_0 f(x)) + \cdots + u_{n-1}(x^{n-1} * f(x)) \in K, \end{aligned}$$

lebo K je cyklicky.

Obratene, nech K je idealom okruhu $\mathbf{F}^n$. Vezmime kodove slovo

$$f(x) = v_0 + \cdots + v_{n-1}x^{n-1} \in K.$$

Potom z predpokladu a lemy 7 vyplyva

$$x * f(x) = v_{n-1} + v_0 x + \cdots + v_{n-2}x^{n-1} \in K.$$

Teda K je cyklicky kod.

Priklad 26. Nech K je binarnym kodom celkovej kontroly parity dlzky 4. V priklade 7 sme ukazali, ze jeho generujuca matica je tvaru

$$\mathtt{G} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Rychle sa presvedcime, ze K pozostava z nasledujucich slov (polynomov): 0, $1+x$, $1+x^2$, $1+x^3$, $x+x^2$, $x+x^3$, x^2+x^3 a $1+x+x^2+x^3$. Priamo z definicie alebo z vety 28 sa presvedcime, ze sa jedna o cyklicky kod.

Generujuci polynom. (11. lekcia)

Veta 29. Kazdy netrivialny cyklicky $(n,k)-$kod K obsahuje polynom $g(x)$ stupna $n-k$. Ten ma nasledujuce vlastnosti:

(i) Kod K, t.j. ideal K v okruhu $\mathbf{F}^n$, je hlavny a je generovany polynomom $g(x)$;

(ii) Polynomy $g(x),\ x*g(x),\ \cdots,\ x^{k-1}*g(x)$ tvoria bazu kodu (podpriestoru) K;

(iii) $g(x) \mid x^n-1$ v okruhu $F[x]$.

Dokaz. Vyberme $g(x) \in K$ najnizsieho mozneho stupna a nech $g(x) \neq 0$. (K predpokladame netrivialne!) Nech st $g(x) = s$. Pozrime sa najprv na (i). Chceme ukazat, ze $g(x)$ generuje K, t.j. ze pre kazde $v(x) \in K$ existuje take $q(x) \in \mathbf{F}^n$ s vlastnostou

$$v(x) = g(x) * q(x).$$

Podla vety o deleni so zvyskom mame v okruhu $F[x]$

$$v(x) = g(x)q(x) + r(x)$$

pre nejake $q(x), r(x) \in F[x]$, pricom st $r(x) < s$. Ak $q(x) = 0$, tak $v(x) = r(x) = 0$, lebo inac sme v spore s vyberom $g(x)$. Podobne, $s = 0$ implikuje $r(x) = 0$. Ovsem $r(x) = 0$ dava zase

$$v(x) = g(x)q(x) = g(x) * q(x)$$

na zaklade dosledku ku vete 27. Teda, mozeme predpokladat, ze $s \geq 1$, $q(x) \neq 0$ a $r(x) \neq 0$. Potom

$$\mathrm{st}v(x) = \mathrm{st}g(x) + \mathrm{st}q(x).$$

Teda st$q(x) < n$. Podla vety 27 a dosledku ku vete 26 mame epimorfizmus

$$\varphi : F[x] \to \mathbf{F}^n \cong F[x]/(x^n - 1)$$

s predpisom

$$v(x)\varphi = g(x)\varphi * q(x)\varphi + r(x)\varphi =$$

$$= v(x) = g(x) * q(x) + r(x).$$

(Pripominame, ze $f(x)\varphi = f(x)$, ak st$f(x) < n$.) Predpokladame vsak, ze K je cyklicky kod, teda K je idealom okruhu $\mathbf{F}^n$ (veta 28). Podla predpokladu $v(x), g(x) \in K$, z coho vyplyva, ze

$$r(x) = v(x) - g(x) * q(x) \in K.$$

Ak by $r(x) \neq 0$, tak by sme dostali spor s vyberom polynomu $g(x)$. Preto $r(x) = 0$. Potom

$$v(x) = g(x)q(x) = g(x) * q(x)$$

a dokazali sme (i).

(ii) Na zaklade (i) predpokladame, ze $\mathrm{st}g(x) = s$ a $\mathrm{st}q(x) \leq n - s - 1$. Potom mame $g(x)q(x) = g(x) * q(x)$ a polynom $g(x) * q(x)$ je linearnou kombinaciou polynomov

$$g(x), \quad x * g(x), \cdots, \quad x^{n-s-1} * g(x),$$

lebo

$$g(x) * q(x) = q_0 g(x) + q_1(x * g(x)) + \cdots +$$

$$+ q_{n-s-1}(x^{n-s-1} * g(x)),$$

ak $q(x) = q_0 + q_1 x + \cdots + q_{n-s-1} x^{n-s-1}$. Ukazeme este, ze generujuce polynomy su aj LN. Nech teda

$$0 = m_0 g(x) + m_1(x * g(x)) + \cdots +$$

$$+ m_{n-s-1}(x^{n-s-1} * g(x)) = m(x) * g(x)$$

pre $m(x) = m_0 + \cdots + m_{n-s-1}x^{n-s-1} \in F[x]$. Potom

$$m(x) * g(x) = m(x)g(x)$$

v okruhu $\mathbf{F}^n$, lebo st$m(x) \leq n - s - 1$. Teda

$$m(x)g(x) = 0$$

v okruhu $F[x]$. Odtial vyplyva $m(x) = 0$, lebo $g(x) \neq 0$ a $F[x]$ je OI. Jedna sa o bazu a plati

$$\dim(K) = k = n - s.$$

Preto $s =$ st$g(x) = n - k$ a plati (ii).

(iii) Nech $x^n - 1 = g(x)q(x) + r(x)$ v okruhu $F[x]$, pricom st$r(x) < n-k$, kde $g(x)$ je generator idealu K. Uvazujme znova o epimorfizme (dosledok za vetou 26)

$$\varphi : F[x] \to \mathbf{F}^n \cong F[x]/(x^n - 1).$$

Potom

$$(x^n - 1)\varphi = 0 = g(x)\varphi * q(x)\varphi + r(x)\varphi =$$

$$= g(x) * q(x) + r(x).$$

Odtial,

$$r(x) = g(x) * (-q(x)) \in K.$$

Ovsem $r(x) \neq 0$ vedie k sporu s vyberom $g(x)$, co znamena $r(x) = 0$. Zaverom mame $x^n - 1 = g(x)q(x)$ v okruhu $F[x]$, t.j. $g(x) \mid x^n - 1$ v okruhu $F[x]$.

Poznamka. Polynom $g(x)$ z predchadzajucej vety je az na asociovanost jednoznacne urceny. (Preco?)

Definicia 24. Polynom $g(x) \in F[x]$ z vety 29 volame *generujucim* polynomom cyklickeho $(n, k)-$kodu.

Teraz mozeme generujucu maticu `G` typu $k \times n$ cyklickeho kodu zapisat pomocou generujuceho polynomu

$$g(x) = g_0 + g_1 x + \cdots + g_{n-k} x^{n-k}$$

nasledovne

$$\begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & . & . & . & 0 \\ 0 & g_0 & \cdots & g_{n-k-1} & g_{n-k} & 0 & 0 & \cdots & 0 \\ & . & . & . & \cdots & . & . & . & \\ 0 & 0 & \cdots & g_0 & g_1 & & \cdots & . & g_{n-k} \end{pmatrix}.$$

Inac povedane, matica $\mathtt{G}$ je jednoznacne urcena svojim prvym riadkom

$$g_0\ g_1\ \cdots\ g_{n-k}\ 0\ \cdots\ 0$$

dlzky n s $k-1$ nulami na konci. Druhy riadok dostaneme z prveho cyklickym posunom doprava o jedno miesto. Podobne dostaneme treti riadok z druheho, az nakoniec, $k-$ty riadok z predposledneho.

Priklad 27. V priklade 26 sme zistili, ze kod celkovej kontroly parity pre $n = 4$ je cyklicky. Ked sa pozrieme na kodove slova, zistime, ze $1 + x$ je jeho generujucim polynomom. Potom generujuca matica vyzera nasledovne

$$\mathtt{G} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Priklad 28. Nech K je binarny kod urceny generujucou maticou

$$\mathtt{G} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Potom $K = \{0000, 1111, 0101, 1010\}$. Je to cyklicky kod. Jeho generujucim polynomom je $g(x) = 1 + x^2$. Ak vyrobime generujucu maticu $\mathtt{G}_1$ pomocou tohto polynomu, tak dostaneme

$$\mathtt{G}_1 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Poznamka. Samozrejme, nie kazdy linearny kod je cyklicky. Prominentnym prikladom takeho kodu je Hammingov kod $(7, 4)$ s kontrolnou maticou $\mathtt{H}$ (vid priklad 20) typu 3×7. Pripomeniname, ze stlpce tejto matice odpovedali binarnemu rozvoju cisel $1, \ldots, 7$. Potom slovo 1101001 je kodovym, ale uz slovo 1110100 nie je take. (Preco?) Situacia sa da napravit vhodnou permutaciou stlpcov matice $\mathtt{H}$. Dostaneme novu

kontrolnu maticu $\mathtt{H}_1$, ktora uz definuje cyklicky kod. V nasom pripade sme na ziskanie matice $\mathtt{H}_1$ pouzili nasledovne transpozicie stlpcov matice $\mathtt{H}$: 3 $\leftrightarrow$ 4, 5 $\leftrightarrow$ 7 a 5 $\leftrightarrow$ 6.

Kontrolny polynom. (12. lekcia)

Veta 30. Nech $K \subseteq \mathbf{F}^n$ je cyklicky (n, k)-kod urceny generujucim polynomom $g(x) \in F[x]$. Nech $x^n - 1 = g(x)h(x)$ v $F[x]$. Potom $v(x) \in K$ prave vtedy, ked $v(x) * h(x) = 0$ v okruhu $\mathbf{F}^n$.

Dokaz. Pretoze $g(x)h(x) = x^n - 1$ v $F[x]$, tak dostaneme $g(x) * h(x) = 0$ v okruhu $\mathbf{F}^n$. Nech $v(x) \in K$. Kedze $K = (g(x))$ v okruhu $\mathbf{F}^n$, tak $v(x) = q(x) * g(x)$ v $\mathbf{F}^n$. Odtial,

$$v(x) * h(x) = (q(x) * g(x)) * h(x) =$$

$$= q(x) * (g(x) * h(x)) = 0$$

v okruhu $\mathbf{F}^n$. Obratene, nech $v(x) * h(x) = 0$ v $\mathbf{F}^n$. Mozeme predpokladat, ze v $F[x]$

$$v(x) = g(x)q(x) + r(x),$$

pricom st$r(x) <$ st$g(x) = n - k$. Zrejme $q(x) \in \mathbf{F}^n$, lebo st$q(x) < k < n$. Pouzijeme epimorfizmus

$$\varphi : F[x] \to F[x]/(x^n - 1) \cong \mathbf{F}^n.$$

Potom

$$(v(x)h(x))\varphi = v(x)\varphi * h(x)\varphi = v(x) * h(x) = 0 =$$

$$(q(x)\varphi) * (g(x)\varphi) * (h(x)\varphi) + (r(x)\varphi) * (h(x)\varphi) =$$

$$q(x) * g(x) * h(x) + r(x) * h(x) = r(x) * h(x),$$

lebo $g(x) * h(x) = 0$. Lenze $r(x) * h(x) = 0$ v $\mathbf{F}^n$ je ekvivalentne s tym, ze

$$x^n - 1 \mid r(x)h(x)$$

v okruhu $F[x]$. Pretoze st$r(x) <$ st$g(x) = n - k$, tak st$(r(x)h(x)) < n$. Dostali sme $r(x)h(x) =$

0, co znamena, ze $r(x) = 0$, lebo $h(x) \neq 0$ a $F[x]$ je O.I. Teda plati $v(x) = g(x)q(x) = g(x) * q(x)$, z coho vyplyva $v(x) \in K$.

Definicia 25. Nech $x^n - 1 = g(x)h(x)$ v okruhu $F[x]$ a nech $K \subseteq \mathbf{F}^n$ je cyklicky kod s generujucim polynomom $g(x)$. Potom $h(x) \in F[x]$ nazyvame *kontrolnym* polynomom kodu K.

Nech $h(x) = h_0 + h_1 x + \cdots + h_k x^k \in F[x]$ znamena kontrolny polynom kodu K. Nasledujuca matica $\mathtt{H}$ bude typu $n-k \times n$ a predstavuje kontrolnu maticu urcenu kontrolnym polynomom.

$$\mathtt{H} = \begin{pmatrix} 0 & 0 & \cdots & 0 & h_k & \cdots & h_1 & h_0 \\ 0 & 0 & \cdots & h_k & \cdots & h_1 & h_0 & 0 \\ . & . & \cdots . & . & \cdots & . & . & \\ & & & & & & & \\ h_k & \cdots & h_1 & h_0 & 0 & \cdots & 0 & 0 \end{pmatrix}.$$

Poznamka. Pozor, koeficienty $h_0, \cdots, h_k$ zapisujeme v opacnom poradi, t.j. sprava dolava. Je tu rozdiel v porovnani ku generujucej matici. Dalej, prvy riadok jednoznacne urcuje celu maticu.