

**Kódování zpráv.
Bezpečnostní kódy, detekce a oprava chyb.
Lineární kódy, cyklické kódy.**

Matematické algoritmy (K611MA)

Lucie Kárná, Jan Přikryl

6. přednáška K611MA
čtvrtek 6. listopadu 2008

verze: 2008-11-06 00:44

1 Kódování zpráv

Kódování Co to je

Přenos a uchování informace $\rightarrow$ potřeba nahradit znaky původní zprávy jinými.

Důvody

- nemožnost přenést původní zprávu zvolenými technickými prostředky
- snaha o snížení redundance
- snaha o zabezpečení informace proti chybám

Matematický algoritmus, který tento proces realizuje, se nazýváme algoritmem **kódování**.

V současné době je hlavním důvodem pro volbu jiných znaků (zpravidla jedniček a nul) efektivnost a použití výpočetní techniky.

Jedním z důvodů bývá nemožnost přenést původní zprávu zvolenými technickými prostředky – například Morseův telegraf neumožňoval přenos mluvené řeči, na papír nelze přímo zaznamenat tóny (noty už jsou určitý druh kódu) a podobně.

Kódování Zdrojová a kódová abeceda

Uvažujeme dvě konečné množiny:

- $\mathcal{A}$ - zdrojová abeceda
- $\mathcal{B}$ - kódová abeceda

Prvky množiny $\mathcal{A}$, resp. $\mathcal{B}$, označujeme jako *zdrojové*, resp. *kódové znaky*.

Příklad 1. Kódová abeceda se nejčastěji skládá z číslic, písmen nebo čísel. Jejimi prvky však mohou být například i slova, barvy (dopravní semaforey), geometrické útvary (námořní znaky), tóny nebo gesta (policista řídící dopravu na křižovatce).

Kód Taxonomie dle počtu prvků abecedy

Množina všech *kódových slov* $\mathbf{K}(\mathbf{a})$, kde $\mathbf{a} \in \mathcal{A}$ jsou zdrojové znaky, se nazývá **kód**.

Dělení dle počtu prvků kódové abecedy:

- *binární kód* – množina $\{0, 1\}$
- *ternární kód* – tříprvková abeceda
- *kvaternární kód* – čtyřprvková abeceda

Slovo v dané abecedě je konečná a neprázdná posloupnost prvků z této množiny. Množinu všech slov množiny $\mathcal{A}$ značíme $\mathcal{A}^*$. Slova z $\mathcal{A}^*$, resp. $\mathcal{B}^*$, nazýváme *zdrojové*, resp. *kódové zprávy*.

1.1 Kódování

Kódování Definice

Kódování je zobrazení $\mathbf{K} : \mathcal{A} \rightarrow \mathcal{B}^*$.

Kódování zdrojových zpráv je zobrazení $\mathbf{K}^* : \mathcal{A}^* \rightarrow \mathcal{B}^*$, které je rozšířením kódování $\mathbf{K}$.

Příklad 2 (Morseova abeceda). Morseova abeceda přiřazuje jednotlivým znakům anglické abecedy sekvence teček a čárek oddělené mezerami. Můžeme na ni tedy nahlížet jako na ternární kód s kódovou abecedou $\{., -, /\}$. Zdrojovou zprávu „konec“ zakódujeme jako $- \cdot - / - - - / - \cdot / \cdot / - \cdot - \cdot //$.

Praktický význam mají pouze kódování *prostá* – různým zdrojovým znakům odpovídají vždy různá kódová slova.

Kódování Příklad

Počasí

Pro sledování vývoje počasí máme zakódovat údaje o oblačnosti pomocí posloupnosti nul a jedniček. Můžeme zvolit například toto kódování:

jasno	0
skoro jasno	01
polojasno	100
skoro zataženo	111
oblačno	1000

Zdrojovou abecedou $\mathcal{A}$ jsou pojmy jasno, skoro jasno, polojasno, skoro zataženo, oblačno, kódovou abecedou $\mathcal{B}$ je množina $\{0, 1\}$.

Zdrojovou zprávu „oblačno, polojasno, skoro jasno, polojasno, polojasno“ zakódujeme jako 100010001100100.

Příklad 3 (Baudotův kód). Pro dálnopisy byl vyvinut Baudotův kód a jeho různé varianty. Baudotův kód kodoval jednotlivá písmena jako pětiznakovou sekvenci nul a jedniček (které byly reprezentovány jako otvor, resp. nevyděrovaná pozice na papírové děrné pásce). Tento kód měl $2^5 = 32$ kódových slov, což stačilo pro všechna písmena abecedy, ale již ne pro číslice. Proto musela být vyhrazena dvě speciální kódová slova jako přepínač pro psaní čísel (a speciálních znaků) a přepínač pro psaní písmen.

Jednotlivá písmena abecedy za sebou nenásledovala podle abecedy, například písmeno A mělo kód 01111, písmeno B bylo 11001, C bylo 01001, D bylo 00001 a tak dále. Volba kódových slov pro jednotlivá písmena nebyla náhodná, ale vycházela ze způsobu jejich zapisování pomocí pěti kláves tak, aby minimalizovala únavu operátora.

Kompletní popis Baudotova kódu můžete najít na webové stránce s adresou <http://www.nadcomm.com/fiveunit/fiveunits.htm>.

Zájemcům o dálnopisy a podobná historická zařízení doporučuji navštívit stránky <http://www.nadcomm.com/> nebo <http://www.vauxelectronics.com/gil/tty/>. Ty, které zajímá historie výpočetní techniky, by mohly zaujmout i weby počítačových muzeí <http://www.computer-museum.org/> a <http://www.old-computers.com/>.

Příklad 4 (Kód ASCII). ASCII kód používá pro reprezentaci každého znaku sekvenci sedmi binárních číslic. Tím má možnost vyjádřit $2^7 = 128$ znaků a to stačí pro všechna velká i malá písmena, číslice, interpunkční znaménka a 32 speciálních (řídících) znaků.

Jednotlivá písmena jsou abecedně uspořádána a tvoří souvislou posloupnost.

Příklad 5 (Kód EBCDIC). Kód vyvinula firma IBM ve stejné době, kdy vznikalo ASCII, má písmena také seřazena abecedně, netvoří však souvislou posloupnost – mezi úseky abecedy jsou vsunuty speciální znaky a interpunkční znaménka.

1.2 Prefixové kódy

Prefixový kód Poněkud překvapivá definice

Prefixem (tj. předponou) slova $v_1 v_2 \dots v_n$ je každé ze slov $v_1, v_1 v_2, \dots, v_1 v_2 \dots v_n$ – to znamená každý „začátek“ slova $v_1 v_2 \dots v_n$.

Definice 6. Kódování se nazývá *prefixové*, jestliže je prosté a žádné kódové slovo není prefixem jiného kódového slova.

Prefixové kódování je *vždy jednoznačně dekódovatelné*. Prefixové kódy jsou *jediné kódy, které lze dekódovat znak po znaku*.

Příklad 7. ASCII kód je prefixový kód. Všechna jeho kódová slova jsou stejně dlouhá a proto žádné z nich nemůže být prefixem jiného.

Ano, mnohem logičtější název by byl *neprefixové* kódování.

Možnost dekódovat znak po znaku znamená, že nemusíme čekat na přijetí celé zprávy, ale již v okamžiku, kdy přijmeme začátek zprávy, tvořený jedním kódovým slovem, můžeme určit první zdrojový znak. Tím je dán velký praktický význam prefixových kódů.

Prefixový kód Co není prefixový kód

Příklad 8. Kód z příkladu o počasí není prefixový kód: kódové slovo 0 pro jasno je prefixem kódového slova 01 pro skoro jasno.

Tento kód není jednoznačně dekódovatelný.

Například kódová zpráva 001000111100 mohla vzniknout zakódováním zdrojové zprávy[2pt] „jasno, skoro jasno, jasno, jasno, jasno, skoro zataženo, polojasno“[2pt] nebo jiné zdrojové zprávy[2pt] „jasno, jasno, polojasno, skoro jasno, skoro zataženo, jasno, jasno“.

Existují ovšem i kódování, která jsou jednoznačně dekódovatelná a přitom nejsou prefixová.

Příklad 9. Kódování

A	0
B	01
C	011
D	0111
E	01111

není prefixové, protože kódové slovo pro **A** je prefixem všech ostatních kódových slov, kódové slovo pro **B** je prefixem kódových slov pro **C**, **D** a **E** atd. Každou kódovou zprávu je však možné jednoznačně dekódovat – stačí postupovat od konce zprávy.

1.3 Blokové kódy

Blokový kód Stejná vs. rozdílná délka slov

Délka kódových slov může být různá pro různé zdrojové znaky (příklad o počasí, Morseova abeceda).

Kódy s nestejnou délkou slov se uplatňují v oblastech, kde je potřeba zajistit kompresi dat.

Definice 10. Blokové kódování délky n je kódování, při kterém všechna kódová slova mají stejnou délku n .

Blokový kód je *vždy kódem prefixovým*: Žádné kódové slovo nemůže být prefixem jiného (všechna kódová slova jsou stejně dlouhá).

Ve starší literatuře můžeme najít i termín *pravidelný kód*.

Příklad 11 (Kód celkové kontroly parity). Binární kód celkové kontroly parity délky 4 sestává ze slov délky 4, která obsahují sudý počet jedniček:

0000	0011	0101	0110
1111	1100	1010	1001

Příklad 12 (Kód „dva z pěti“). Kód „dva z pěti“ je blokový kód délky 5, který obsahuje všechna slova délky 5, ve kterých jsou právě dvě jedničky:

00011	00110	01100	11000	10001
10010	10100	01010	01001	00101

Protože těchto slov je celkem 10, můžeme tento kód použít například ke kódování čísl 0–9.

Kód „dva z pěti“ byl k tomuto účelu skutečně používán, poprvé v roce 1943, a později i v prvním československém elektronickém elektronkovém počítači EPOS, uvedeném do provozu v roce 1963.

U některých blokových kódů lze rozdělit znaky na *informační*, do nichž se kóduje informace, a *kontrolní*, které jsou jednoznačně určeny volbou informačních znaků.

Definice 13. Jestliže existuje prosté zobrazení množiny všech slov délky k na množinu všech kódových slov, řekneme, že kód $\mathbf{K}$ má k informačních znaků a $n - k$ kontrolních znaků.

Takový kód označujeme jako (n, k) -kód.

Příklad 14 (Kód celkové kontroly parity). Binární kód celkové kontroly parity délky 4 má 3 informační znaky a jeden (například poslední) znak kontrolní, který je těmito třemi znaky jednoznačně určen.

Stejně tak bychom mohli za kontrolní označit například znak druhý, z praktických důvodů však dáváme přednost výše uvedené volbě kontrolního znaku.

Příklad 15 (Kód „dva z pěti“). Kód „dva z pěti“ nemá oddělené informační a kontrolní znaky. Množina kódových slov má 10 prvků, což není mocnina dvou. Proto nemůže pro žádné k existovat prosté zobrazení množiny $\{0, 1\}^k$ na množinu kódových slov.

Ad „dva z pěti“: K prostému zobrazení musí existovat inverzní zobrazení. Musí to být zobrazení „na“, nikoliv „do“.

Definice 16. Jestliže kód $\mathbf{K}$ má k informačních znaků a $n - k$ kontrolních znaků, pak podíl $\frac{k}{n}$ nazveme *informační poměr kódu $\mathbf{K}$* .

Informační poměr udává množství nadbytečné informace přidané do kódu:

- malý informační poměr (hodně kontrolních znaků) – „neúsporný“ kód, zabírá v paměti více místa, jeho přenos a zpracování trvá déle
- velký informační poměr – nemůže mít dobré detekční a samoopravné vlastnosti

Systematické kódování Rozšíření zdrojového slova

V *systematickém kódování* tvoří zdrojové slovo prvních k znaků kódového slova. Kód, který má systematické kódování, se nazývá *systematický kód*.

Jednou z výhod systematického kódu je jeho snadné kódování i dekódování.

Příklad 17. Binární kód celkové kontroly parity délky 4 je systematický $(4, 3)$ -kód. Ke zdrojovému slovu přidáme jeden kontrolní znak, který je součtem všech znaků zdrojového slova. Například $100 \rightarrow 1001, 101 \rightarrow 1010, 011 \rightarrow 0110$ a podobně.

Příklad 18 (Kód „dva z pěti“). Kód „dva z pěti“ není systematický kód.

Volba kódové abecedy Příklad

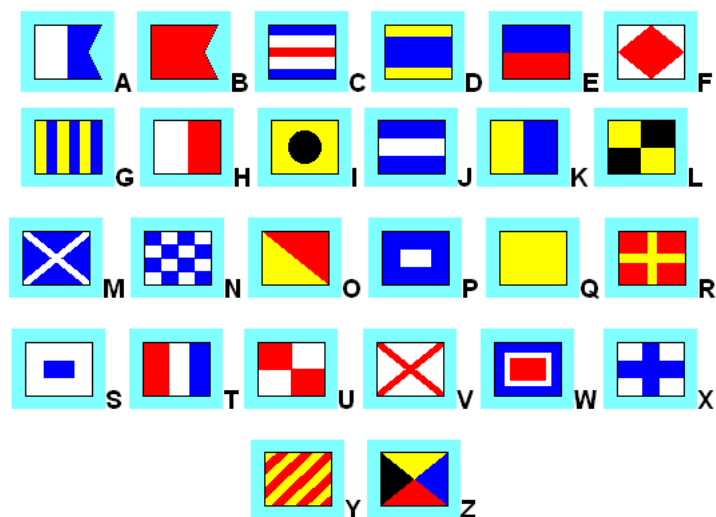
Námořní vlajkový kód

V námořní signalizaci se používá systém jedno- až třípísmenných zkratk. Jednotlivá písmena a číslice jsou reprezentovány vlajkami. Vyvěsí-li například loď vlajku V, znamená to „potřebuji pomoc“, IW říká „požár je pod kontrolou“. Signály týkající se lékařské pomoci se skládají ze tří písmen, z nichž první je M.

Jednotlivá písmena a číslice reprezentována vlajkami:

- písmena (kromě A a B) mají vlajky obdélníkové
- číslice mají tvar podlouhlých rovnoramenných trojúhelníků
- barvy: N je modrobílá šachovnice, Y má šikmé žluté a červené pruhy, I je černý kruh ve žlutém poli

To, že vlajky nejsou jen „obrázky“, ale mají význam písmen, nejen usnadňuje zápis kódu, ale umožňuje i hláskovat jména nebo informace, pro které neexistuje smluvené kódové slovo.



Volba kódové abecedy Pozorování námořního vlajkového kódu

Zřetelná snaha omezit nebezpečí nesprávného rozpoznání vlajky:

- není použita zelená barva, která by mohla být na dálku snadno zaměněna za barvu modrou,
- co nejrozmanitější tvary vzorů – vodorovné, svislé i šikmé pruhy, latinské i ondrějské kříže a další tvary.

Příklad 19. Rozdělíme-li například obdélníkovou vlajku svisle na dvě poloviny, můžeme při použití pěti barev (černá, červená, modrá, žlutá, bílá) vytvořit 10 různých barevných kombinací. Z nich jsou skutečně použité pouze dvě – žluto-modrá vlajka K a bílo-červená vlajka H.

2 Bezpečnostní kódy

Bezpečnostní kódy Motivace

Vnější vlivy přenosu – rušení, případně šum:

- elektromagnetické rušení při rádiovém přenosu dat,
- poškrábaný kompaktní disk,
- přejevo lidského operátora,
- optická komunikace je narušena mlhou nebo sněžením.

Potřeba zamezit zkreslení a špatné interpretaci informace se v kódování vyskytuje velmi často:

- zpravidla nelze řešit jednoduchými prostředky (například vhodnou volbou kódové abecedy)
- je nezbytné zabezpečit správnost informace *vlastnostmi samotného kódu*.

Bezpečnostní a samoopravný kód Definice

Pro zabezpečení informace před šumem se používají dvě třídy kódů:

- **bezpečnostní kódy**, jež umožňují odhalit určitý počet chyb,
- **samoopravné kódy**, které umožňují určitý počet chyb odhalit, případně některé chyby opravit.

Předpokládáme šum, který se projevuje záměnou vyslaného znaku za jiný znak. Šum, při kterém dochází k *poruše synchronizace* (tedy k pohlcení vyslaného znaku nebo vytvoření znaku, který nebyl vyslán), nebudeme uvažovat.

2.1 Objevování chyb

Objevování chyb Definice

Blokovým kód $\mathbf{K}$ délky n v konečné abecedě $\mathcal{T} \rightarrow \mathcal{T}^n$ je množina všech slov délky n .

$\mathbf{K}$ je pouze podmnožinou $\mathcal{T}^n$:

- slova z $\mathbf{K}$ jsou kódová slova,
- slova z $\mathcal{T}^n - \mathbf{K}$ nazýváme *nekódová slova*.

Předpokládáme, že vysíláme kódová slova a přijímáme slova z množiny $\mathcal{T}^n$. Pokud jsme přijali nekódové slovo, říkáme, že jsme **objevili chybu**. Pokud je přijaté slovo kódové, pak *buď k chybě nedošlo, nebo jsme ji neobjevili*.

Objevování chyb Příklad

Příklad 20. Používáme binární kód celkové kontroly parity délky 4 (viz příklad 11). Vyšleme slovo 0110. Dojde-li při přenosu ke změně 3. znaku, přijmeme nekódové slovo 0100 a je zřejmé, že nastala chyba – objevili jsme chybu. Jestliže dojde ke změně 1. a 3. znaku zároveň, je přijaté slovo 1100 kódové a chybu jsme neobjevili.

Objevování chyb Násobné chyby

Chyba je *t-násobná*, jestliže počet chybných míst je nejvýše t .

Definice 21. Kód $\mathbf{K}$ objevuje *t-násobné chyby*, jestliže při vyslání kódového slova v a vzniku jakékoliv *t-násobné chyby* je přijaté slovo w vždy nekódové.

Příklad 22. Binární kód celkové kontroly parity (libovolné délky) objevuje jednoduché chyby (při změně jednoho znaku se změní jejich součet), ale neobjevuje dvojnásobné chyby.

Ad neobjevuje dvojnásobné chyby – k tomu nám stačí protipříklad uvedený výše.

2.2 Minimální vzdálenost

Minimální vzdálenost Definice Hammingovy vzdálenosti

Mějme dvě slova délky n : $u = u_1 u_2 \dots u_n$ a $v = v_1 v_2 \dots v_n$.

Hammingovou vzdáleností (nebo jen *vzdáleností*) slov u a v nazveme počet pozic, ve kterých se slova od sebe liší, tedy velikost množiny $\{i; i = 1, 2, \dots, n, u_i \neq v_i\}$.

Hammingovu vzdálenost slov u a v značíme $d(u, v)$.

Definice 23. Minimální vzdáleností blokového kódu $\mathbf{K}$ nazveme nejmenší Hammingovu vzdálenost dvou různých kódových slov tohoto kódu.

Minimální vzdálenost Příklady

Příklad 24. Hammingova vzdálenost slov $u = 1101001$ a $v = 0110000$ je $d(u, v) = 4$, Hammingova vzdálenost slov $w = 01101$ a $z = 01001$ je $d(w, z) = 1$. Hammingova vzdálenost libovolného slova od sebe sama je $d(u, u) = 0$.

Příklad 25 (Kód „dva z pěti“). Minimální vzdálenost kódu „dva z pěti“ je 2. Například slova 00110 a 00101 mají Hammingovu vzdálenost 2 a žádná dvě slova nemají Hammingovu vzdálenost 1.

Násobné chyby A jejich vztah k minimální vzdálenosti

Blokový kód minimální vzdálenosti d objevuje t -násobné chyby pro všechna $t < d$, ale neobjeví všechny d -násobné chyby. Může ovšem odhalovat určitou podmnožinu těchto chyb.

Příklad 26 (Kód „dva z pěti“). Kód „dva z pěti“, jenž má minimální vzdálenost 2, objevuje jednoduché chyby a neobjevuje všechny dvojnásobné chyby – neodahlí například dvojnásobnou chybu, kdy je místo slova 01100 přijato slovo 01010. Objeví však dvojnásobnou chybu, při které je místo slova 01100 přijato slovo 01111.

2.3 Opravování chyb

Dekódování Základní pravidlo

Pravěpodobnost, že při vyslání slova v bude přijato slovo w , značíme $P_e(v, w)$. Pravidlo dekodování maximální pravděpodobnosti můžeme zapsat takto

Definice 27. Dekódování maximální pravděpodobnosti: Přijaté slovo dekodujeme jako to kódové slovo, které má od něj co nejmenší Hammingovu vzdálenost.

Je-li přijato slovo w , dekodujeme jej jako to kódové slovo $v \in \mathbf{K}$, pro které je

$$P_e(v, w) = \max_{u \in \mathbf{K}} P_e(u, w).$$

Všimněte si: Přijaté kódové slovo má samo od sebe má nejmenší možnou (nulovou) Hammingovu vzdálenost.

Opravování a detekce chyb Příklady

Předpokládejme, že při přenosu binárního kódu dojde s pravděpodobností p_e k záměně znaku 0 za znak 1 nebo znaku 1 za znak 0.

Příklad 28. Nechť je 0011 kódové slovo. Pak pravděpodobnost, že při vyslání tohoto slova bude přijato slovo 0101 je $P_{p_e}(0011, 0101) = (1 - p_e)^2 p_e^2$.

Příklad 29. Buď $\mathbf{K}$ kód délky 4. Pak pro každé kódové slovo $v \in \mathbf{K}$ je pravděpodobnost, že v bylo přijato správně, $P_{p_e}(v, v) = (1 - p_e)^4$.

Korekční schopnosti blokového kódu Vztah d a t

Blokový kód minimální vzdálenosti d opravuje t -násobné chyby pro všechna $t < \frac{d}{2}$, ale neopraví všechny chyby násobnosti $\frac{d}{2}$.

Příklad 30. Binární opakovací kód délky 5 opravuje dvojnásobné chyby (jeho minimální vzdálenost je 5).

Příklad 31. Binární kód celkové kontroly parity (libovolné délky) neopravuje žádné chyby.

Chyba dekódování Kdy použít samoopravné kódy

Při opravě chyby se samozřejmě může stát, že dojde k *chybě dekódování*, pokud vyslané slovo nemá od slova přijatého nejmenší možnou Hammingovu vzdálenost.

Pravděpodobnost chyby – velmi malá, *nikdy však nemůže být nulová*.

Proto se v některých aplikacích – zejména v aplikacích vztahujících se k bezpečnosti – samoopravné vlastnosti kódů nevyužívají.

V mnoha jiných aplikacích (pokud komunikace není obousměrná nebo není možnost zprávu opakovat) jsou samoopravné kódy velmi užitečné.

3 Lineární kódy

Lineární kódy Úvod

Důležitou podskupinou blokových kódů jsou *kódy lineární*.

Význam:

- pro jejich kódování a dekódování existují výpočetně relativně nenáročné postupy
- teoretické stanovení a ověření jejich vlastností (včetně bezpečnostních) je snazší než u obecných blokových kódů

3.1 Základní pojmy

Lineární kódy Základní pojmy

Definice 32. Blokový kód $\mathbf{K}$ délky n nad abecedou $\mathcal{T}$ se nazývá **lineární**, jestliže je podprostorem lineárního prostoru $\mathcal{T}^n$.

Každý lineární kód obsahuje nulové slovo o (slovo ze samých nul).

Příklad 33 (Kód celkové kontroly parity). Binární kód celkové kontroly parity délky 4 je lineární kód. Součet libovolných dvou kódových slov (tedy slov se sudou paritou) má také sudou paritu a je tudíž kódovým slovem.

Být podprostorem znamená v podstatě to, že součet libovolných dvou kódových slov je kódové slovo a že skalární násobek libovolného kódového slova je opět kódové slovo.

Lineární kódy Další vlastnosti

Je-li kód $\mathbf{K}$ délky n podprostorem dimenze k , nazývá se *lineární (n, k) -kód*.

Lineární (n, k) -kód

- má k informačních a $n - k$ kontrolních znaků a jeho informační poměr je tedy $\frac{k}{n}$

- nad r -prvkovou abecedou má r^k slov.

Binární charakter výpočetní techniky $\rightarrow$ největší praktický význam mají kódy nad *tělesy charakteristiky 2*. V těchto tělesech pro každý prvek b platí: $b + b = 0$ a každý prvek je tedy opačný sám k sobě. Operace odčítání je v tělese charakteristiky 2 totožná s operací sčítání.

Příklad 34. Binární lineární (n, k) -kód má tedy 2^k slov.

3.2 Chyby a minimální vzdálenost

Chyby a minimální vzdálenost Linearita kódu usnadňuje analýzu chyb

Definice 35. Hammingovu vzdálenost slova v od nulového slova nazveme **Hammingovou váhou** slova v .

Vyslali jsme slovo $v_1 v_2 \dots v_n$ a přijali slovo $w_1 w_2 \dots w_n$. Rozdíl

$$e_1 e_2 \dots e_n = (w_1 w_2 \dots w_n) - (v_1 v_2 \dots v_n)$$

nazveme *chybovým slovem*.

Lineární kód $\mathbf{K}$ objevuje chybové slovo $e_1 e_2 \dots e_n$, jestliže součet tohoto chybového slova s libovolným kódovým slovem není kódové slovo.

Speciálně, kód objevuje t -násobné chyby, právě když objevuje všechna chybová slova Hammingovy váhy menší nebo rovné t .

Lineární kód objevuje právě ta chybová slova, která nejsou slovy kódovými.

Minimální vzdálenost lineárního kódu je rovna nejmenší Hammingově váze nenulového kódového slova:

$$d(\mathbf{K}) = \min \{ \|v\|; v \in \mathbf{K} - \{0\} \}.$$

Jednodušší výpočet minimální vzdálenosti

- buď spočítat Hammingovu váhu všech kódových slov,
- nebo postupně
 - generovat všechna slova z T^n , s Hammingovou váhou $1, 2, \dots$
 - zjišťovat, zda jsou slovy kódovými
 - při nalezení prvního kódového slova s váhou $\|v\|$ výpočet ukončíme – minimální vzdálenost kódu je pak $\|v\|$

Pro kódy s větší minimální vzdáleností nebo s malým počtem kódových slov je vhodnější první postup.

4 Závěr

Příště ...

Analýza algoritmů, složitost.