Megoldás

Digitális technika I. (vimia102)
1. gyakorlat: Kódolástechnikai alapok

Elméleti anyag:

- Analóg és digitális kód, analóg és digitális feldolgozás korlátjai
- Az információátvitel általános modellje: forrás, kódoló, csatorna, dekódoló, megfejtés
- A digitális kódok felépítése, osztályozásuk: kód, kódszó, kódkészlet, bináris, nem-bináris, fix és változó hossz, osztályozás a kód célja szerint
- Forráskódolás (veszteségmentes tömörítés, zajmentes kódolás), cél a kis átlagos kódszóhossz
- Megfejthetőség, prefix kódok
- Shannon kód, Huffmann kód, forráskiterjesztés
- A tömörítés alsó határa:, forrás-entrópia
- Digitális csatornák hibái: folthiba, véletlen hiba, eltörlődéses, átállítódásos
- Hibavédelmi stratégiák: hibajelzés, -javítás, vegyes
- Hamming távolság, kapcsolata a hibajelzéssel, hibajavítással
- Pritáskód, ismétléses kód,
- **A 2. gyakorlatra áthúzódó anyag:**
- Hamming kód
- Számábrázolás: fixpontos és lebegőpontos
- Előjeles számok ábrázolása: abszolutértékes, egyes-, kettes komplemens, off-set
- Pozíciókódok (egy Hamming-távolságú kódok): Gray kód, Johnson kód
- Egyéb kódok: n-ből az m, NBCD, EXCESS-3

Irodalom:
Benesóczky Zoltán: Kódoláselméleti alapfogalmak (2005), elektronikus jegyzet
http://home.mit.bme.hu/%7Ebenes/oktatas/dig-jegyz_052/kodolas.pdf

Az első gyakorlatokhoz az előadáson a paritáskódig jutunk el (meg a felette lévő elméleti anyag) ezért a Hamming kódokról és a pozíciókódokról a 2. gyakorlaton lesz rövid megemlékezés. Természetesen az ellenőrző kérdéseket is két részre bontottuk: az első ill. második gyakorlaton sorrakerülő kérdések.
A számábrázolás CSAK a gyakorlat anyaga.
A „segédletek” könyvtárban van egy .xls program változó hosszúságú kódok ellenőrzéséhez.

Gyakorló példák:

1.1. A decimális számjegyeket az alábbi táblázat szerint kódoljuk:

0	**0000**
1	**0001**
2	**0010**
3	**0011**
4	**010**
5	**011**
6	**100**
7	**101**
8	**110**

9	111

Rajzolja fel a kódolási fát!
Kódolja születésnapját: ééhhnn (éé: az évszám utolsó két jegye, hh: a hónap, nn: a nap)!

Megoldás: pl. 561023 esetén:

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

Itt bejelöltük az egyes kódszavakat, természetesen a színeket a dekódoló nem látja!

Cseréljenek szomszédjukkal és fejtsék meg egymás szülinapját!

1.2. Egy cinkelt dobókockával való 1-6 dobások gyakorisága rendre:
0.2, 0.1, 0.1, 0.05, 0.05, 0.5.
Kódolja tömören a kockadobások eredményét és számítsa ki az átlagos kódszóhosszt!
Hány - különböző hosszúság-eloszlású - optimális kódot talál?

Megoldás:
Kísérletezzünk a Huffmannn kódolással, mert az optimális kódot ad!
Oldja meg a mindenki önállóan és számítsa ki az átlagos kódhosszat. Aki jól számolt 2.1 bitet kap.
Ezzel a feladat első részét megoldottuk.

Nehezebb a második rész, ha szerencsénk van, akkor a sok jó megoldásból kiadódnak a különböző kódhossz-eloszlások.
Mindez „tudatosan”:
Mivel 6 esemény van, összesen öt összevonással a kódolás végére érünk. Jelen példában az első és az utolsó két összevonás egyértelmű: először biztosan a 4-es és 5-ös dobásokat kell összevonni (ezek a legkisebb valószínűségűek), az utolsó előttiben a már összevont négy picihez az ötödiket csatlakoztatjuk és legutoljára vonjuk össze a többivel a 6-os dobást (ennek a valószínűsége 0.5, tehát csak akkor lehet a „legkisebb”, ha az összes többi már egy csoportban van). A közbenső összevonásoknál azonban példánkban többféleképpen választhatunk! (a másodiknál 3, a harmadiknál 2 választási lehetőségünk van) Ezeket a választási lehetőségeket foglalja össze az alábbi táblázat:

	1	variáció	2	variáció	3	variáció
	lépések	kódhosszak	lépések	kódhosszak	lépések	kódhosszak
1	45	3	45	2	45	3
2	23	3	23	4	245	3
3	145	3	2345	4	13	3
4	12345	4	12345	4	12345	4
5	123456	4	123456	4	123456	4
6		1		1		1

Átlagos kódszóhossz:

		2.1		2.1		2.1

	4	variáció	5	variáció	6	variáció
	lépések	kódhosszak	lépések	kódhosszak	lépések	kódhosszak
1	45	2	45	3	45	2
2	245	4	345	3	345	3
3	2345	3	12	3	2345	4
4	12345	5	12345	4	12345	5

5	123456	5	123456	4	123456	5	
6		1		1		1	

Átlagos kódszóhossz:

		2.1		2.1		2.1

Az egyes variációknál az összevonásokat a „lépések” oszlopban adjuk meg, pl. a 4. variáció összevonás sorozata:

- 45: 4-es és 5-ös összevonása
- 245 a 2-es összevonása a 45-tel
- 2345 a 3-mas összevonása a 245-tel
- 12345 az előzőhöz az 1-es csatolása
- 123456 itt a vége!

Látható, hogy a 6 variációból három különböző kódhossz-eloszlás adódik:

1-3-3-3-4-4
1-2-4-4-4-4
1-2-3-4-5-5

Természetesen ezeknél az átlagos kódhossz megegyezik, mert mindegyik optimális kód.

Akinek van kedve, az a Shannon módszerrel is megoldhatja ezt a feladatot.

1.3. Keressen tömör kódot, ha az egyes események gyakorisága:
a/ 1/2, 1/4, 1/8, 1/16, 1/32, 1/32.
b/ 1/4, 1/4, 1/8, 1/8, 1/8, 1/16, 1/16.
c/ 10 db. 0.1.
Minden esetben számítsa ki az átlagos kódszóhosszat is!

Megoldás:
Pl. optimális prefix kódok rendre:
a/ 0, 10, 110, 1110, 11110, 11111 — 31/16 = 1.9375 bit
b/ 00, 01, 100, 101, 110, 1110, 1111 — 21/8 = 2.625 bit
c/ 000, 001, 010, 011, 100, 101, 1100, 1101, 1110, 1111 — 3.4 bit

1.4. Kételemű eseményrendszere van 0.8 és 0.2 valószínűségekkel.
a. Mennyi a forrás entrópiája?
b. Mekkora a bináris kód üzenetenkénti hossza?
c. Készítse el a kétszeres forráskiterjesztéhez az optimális kódot!
d. Ugyanez a háromszoros kiterjesztéshez.
Hogyan alakulnak az üzenetenkénti átlagos hosszak?

Megoldás:
a. Az entrópia 0.722 bit
b. A két eseményt 0-val és 1-el kódoljuk, az átlagos hossz: 1 bit
c. A kétszeres kiterjesztésnél az alábbi valószínűségek, kódhosszak, entrópia és átl. Hossz adódik (egy eredeti üzenetre az entrópiának és az átl. Hosszanak a fele jut.

valószínűségek	Kódhoszak	Entrópia	1.444
0.64	1	Átlagos kódhossz	1.560
0.16	2		
0.16	3		

0.04	3

d. Háromszoros kiterjesztésnél a következő táblázat adja meg az adatokat:

valószínűségek	Kódhoszak	Entrópia	2.166
0.512	1	Átlagos kódhossz	2.184
0.128	3		
0.128	3		
0.128	3		
0.032	5		
0.032	5		
0.032	5		
0.008	5		

Összefoglalva:
Entrópia: 0.722 bit
Átl. hossz egyszeres kiterjesztésnél: 1 bit
Kétszeres kiterjesztésnél: 0.78 bit
Háromszoros kiterjesztésnél: 0.728 bit
Ez utóbbi már nagyon közel van a lehetséges minimumhoz!

1.5. Adja meg 12 biten az alábbi számokat

- előjeles abszolutértékes,
- egyes komplemens,
- kettes komplemens,
- offset kódú bináris számábrázolásban!

35, -35, 278, -278, 0, -2047, 2047, ..

Megoldás:

35:	0000.0010.0011	0000.0010.0011	0000.0010.0011	1000.0010.0011
-35:	1000.0010.0011	1111.1101.1100	1111.1101.1101	0111.1101.1101
278:	0001.0001.0110	0001.0001.0110	0001.0001.0110	1001.0001.0110
-278:	1001.0001.0110	1110.1110.1001	1110.1001.1010	0110.1001.1010
0:	0000.0000.0000	0000.0000.0000	0000.0000.0000	1000.0000.0000
	1000.0000.0000	1111.1111.1111		
2047:	0111.1111.1111	0111.1111.1111	0111.1111.1111	1111.1111.1111
-2047:	1111.1111.1111	1000.0000.0000	1000.0000.0001	0000.0000.0001

1.6. A furfangos hallgató jól meg akarja védeni nyolcbites adatait, ezért a következő redundáns kódot találja ki:
A nyolc bit után írja

- a/ egy biten a kiinduló adat paritását,
- b/ egy biten a kiinduló adat első négy bitjének paritását,
- c/ egy biten a második négy bit paritását,
- d/ egy biten a páros pozíciójú bitek paritását és
- e/ egy biten a páratlan pozíciójú bitek paritását.
- f/ Ezzel még nem elégszik meg, biztonság kedvéért az egész végére írja még a kiinduló adat 1-es komplemensét.

Így összesen 21 bites redundáns kódkészletet kap, mekkora ennek a (minimális)

Hamming távolsága?

Megoldás:
Ha a kiinduló kódszóban 1 bitet változatunk meg, akkor ez a/-ban 1, b/ vagy c/-ben 1, d/ vagy e/-ben 1 és f-ben újabb egy bit eltérést okoz, tehát összesen 5 a Hamming távolság.

Ha a kiinduló adatban "ügyesen" rontunk el két bitet (pl. az 1. és a 3. bitet invertáljuk), akkor a/-e/ ellenőrző bitek nem változnak. Így a kiinduló kódrész és az f/ együttesen 4 bitben fognak eltérni. Ennél kevesebbet már semmilyen módon nem lehet produkálni, ezért a furfangos kód Hamming távolsága 4.

1.7. Mennyi a (minimális) Hamming távolsága annak a decimális kódnak, amelynek kilenc információs jegyéhez az ***r*** redundáns karaktert illesztjük:

r = mod10 (1.a1+2.a2+3.a3+4.a4+5.a5+6.a6+7.a7+8.a8+9.a9)

Megoldás:
Sajnos csak 1, tehát a "legrosszabb" esetet tekintve semmire sem jó!
Ugyanis ha a2-ben 5-öt, vagy a5-ben 2-t tévedünk, akkor az ellenőrző karakter változatlan marad.

1.8. Mekkora Hamming távolságú kódszókészlet kell 6 átállítódásos hiba javításához? Ez a kódszókészlet milyen további javítás/jelzés variációkban használható még fel?

Megoldás:
Hat átállítódásos hiba javításához min 13-as Hamming távolság kell.
Ez a kód alkalmas még:

5 hiba javítására és még a 6.és 7. jelzésére,
4 hiba javítására és még az 5.-8. jelzésére,
3 hiba javítására és még a 4.-9. jelzésére,
2 hiba javítására és még a 3.-10. jelzésére,
1 hiba javítására és még a 2.-11. jelzésére.

Nehéz példák az érdeklődőknek:

1.n1. A hétbites üzeneteket kell tömörebben kódolni, mert ezekben a 0-ák sokkal gyakoribbak, mint az 1-esek:

Kiinduló üzenet	és a valószínűség
0000000	0.9
0000001	0.01
0000010	0.01
0000100	0.01
0001000	0.01
…….	
1000000	0.01

a többi 120 kiinduló üzenet egyforma valószínűségű: egyenként 0.03/120

Kódolja tömören ezt a 128 elemű üzenet-rendszert!

Megoldás:

1. A forrás entrópiája: 0.97 bit – ez tehát a tömörítés alsó határa
2. Ha nem tömörítünk, akkor az átl. kódszóhossz: 7 bit
3. Ha 127 lépésben (ügyesek ebből néhányat spórolhatnak) végigvinnénk a Huffmann kódolást, akkor 1.498 bites átl. kódszóhosszat kapnánk.
4. Ugyanerre a Shannon kódolás 1.502 bitet adna
5. Ezeknél a tömör kódoknál alig valamivel rosszabb az alábbi intuitív kódolás:

- a csupa 0 üzenet kódja legyen: 0
- a 7 db. egyetlen egyest tartalmazó üzenet kódja legyen: 1 után írjuk három biten az egyes sorszámát (1000…1110)
- a többit nem tömörítjük, hanem leírjuk az eddig szabadon maradt 1111 után

Ennél a megoldásnál az átl. kódhossz:

0.9x1 bit + 0.07x4 bit + 0.03x11 bit = 1.51 bit

6. Még ez is messze van az entrópia által megszabott alsó határtól. További javítás a „forráskiterjesztéssel” lehetséges. Ki lehet próbálni 2, 3 stb. 7 bites üzenet összevonásával jó intuitív kódolásokat.

1.n2. A Shannon kód nem”optimális”, vagyis lehet olyan eseményrendszer, amelyiknél rosszabb eredményt ad a Huffmann kódnál.
Keressen ilyen példát! A legszebb példa a lehető legkevesebb számú eseményt tartalmazza.

1.n3. Hogyan kellene gazdaságosan kódolni, ha a kód ábc elemeinek különböző lenne a költsége?
Pl. bináris kódnál a 0 karakter ára 2 Ft, az 1 karakter ára pedig 1 Ft. Ilyenkor nyilván több 1-est és kevesebb 0-t használnánk a legolcsóbb kódhoz.
Találjon ki erre a problémára „jó” kódolási módszert!

1.n4. Bizonyítsa be, hogy a hatbites, hármas Hamming távolságú kódszavak maximális száma nyolc.

1.n5. Melyik a „legolcsóbb” számrendszer?