Kódolástechnika vizsga | 2016.12.19.

1) Adott egy nem rövidített, ciklikus RS kód a következő generátorpolinommal a GF(8) felett $g(x) = x^2+y^4x+y^3$

a) Mik a kód paraméterei? (5p)

GF(8) => q=8 és n=q–1=8–1= 7

deg(g(x)) = n - k = 2 ⇔ 7-k=2 ⇔ k=5 => C(7, 5)

b) Hány hibát tud javítani a kód? (5p)

Általánosan: $t = alsóegész[(d_{min}-1)/2]$ és $d_{min} \leq n-k+1$

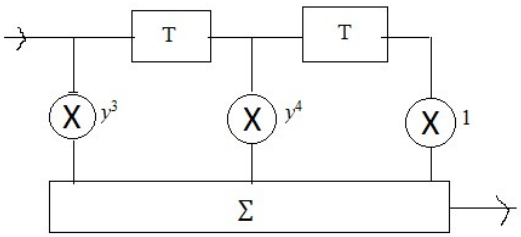
RS => MDS => $d_{min} = n-k+1$ => $t=alsóegész[(n-k)/2]= alsóegész[2/2] = 1$

c) Hányad fokú a paritásellenőrző polinom? (5p)

deg(h(x)) = k = 5

d) Milyen HW architektúrával valósítható meg a kódolás? (5p)

Előrecsatolt shiftregiszter



2) Karikázza be a helyes állításokat az alábbi listán (csak akkor adható rá 20 pont, ha minden állításról helyesen döntött, különben 0 pont).

a) Egy *C(n, k)* paraméterű ciklikus kód generátorpolinomja ~~nem~~ osztja az x^n-1 polinomot

b) Két azonos szindrómavektorhoz tartozó hibavektorból arra érdemes detektálni, amelynek kisebb a súlya

c) Egy *C(7, 4)* paraméterű kód lehet Hamming kód

Teljesíti a $2^{n-k} = n+1$ egyenlőtlenséget.

d) A szisztematikus kódok paritásellenőrző mátrixának az utolsó *k x k*-s szegmense egységmátrix. (*[n-k] x [n-k]*)

e) Két polinom szorzatát előrecsatolt shiftregisztereken lehet implementálni.

3) Adott egy emlékezetnélküli forrás a következő eloszlással: $p_1=0.7$; $p_2=0.2$; $p_3=0.1$

a) Adja meg a tömöríthetőség elvi alsó határát! (4p)

$H(x) \leq L$ => $H(x)=SZUM_{minden\ x\text{-}re}[p(x)*ld(1/p(x))] = 0.7*0.515+0.2*2.322+0.1*3.322=1.16$

b) Tömörítse a forrást Huffman kóddal: adja meg a kódot és az átlagos kódszóhosszat! (4p)

$p_1=0.7$ – 0.7 – 1 . – $c_1=(1)$
$p_2=0.2$ – 0.3 / \ – $c_2=(01)$
$p_3=0.1$ / \ $c_3=(00)$

L = *$SZUM_{minden\ x\text{-}re}[p(x)*l(x)] = 0.7*1+0.2*2+0.1*2 = 1.3$*

c) Mi lenne az átlagos kódszóhossz, ha Shannon Fano kóddal akarnánk tömöríteni? (4p)

L^{SF}= *$SZUM_{minden\ x\text{-}re}[p(x)*felsőegész[ld(1/p(x))]] = 0.7*1+0.2*3+0.1*4=1.7$*

d) Milyen hosszú blokkokban kell a forrást kódolni, ha az elvi alsó határt Ɛ=0.02-al akarnánk megközelíteni (4p)

H(X)+ Ɛ = H(X) + 1/K ⇔ Ɛ=1/K ⇔ K=1/ Ɛ=1/0.02=50

e) Mekkora lesz így a kódtábla mérete (hány sort fog tartalmazni)? (4p)

Compl: $O(3^{50})$

4) Csak egy számmal adja meg a válaszokat (minden helyes válasz 4p)!

a) Milyen min. és max. érték közé esik egy 4 darab szimbólummal rendelkező forrás entrópiája?

0 ≤ *H(x) ≤ ld(N) = ld(4) =* **2**

b) Egy 8 szimbólumot tartalmazó egyenletes eloszlású forrás esetén hány bitesek lesznek az optimális kódhosszak?

Egyenletes eloszlású => entrópia max., továbbá H(x) ≤ L és L min. (mert optimális) => L = ld(N) = **3**

c) Egy 5 szimbólummal és az $l_1=2;\ l_2=2;\ l_3=2;\ l_4=3$ kódhosszakkal rendelkező forrás esetén legalább milyen hosszú legyen az ötödik szimbólumhoz tartozó kódszó hossza, ha az egyértelmű dekódolhatóság a cél?

Egyértelmű dekódolhatóság: prefixmentes kód kell! A kód prefixmentes, ha teljesíti a $\sum_{i=1}^{n} r^{-\ell_i} \leqslant 1.$ *, r=2, n=5 (Kraft–McMillan) egyenlőtlenséget. l_5=2-re nem teljesül, de l_5=**3**-ra már igen.*

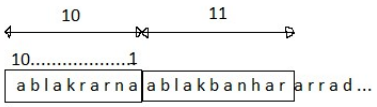
d) Adott két független egyenletes eloszlású bináris forrás. Mekkora az együttes entrópiájuk?

Bináris => N=2 => H(x), H(y) = ld(N) = 1
Flen. => H(x, y) = H(x) + H(y) = 1+1 = **2**

e) Hány darab prím szám kell az RSA algoritmus kulcsának generálásához egy adott felhasználó esetén?

p_1, p_2 => **2** *darab*

5) Adott egy LZ77 kódoló a következő architektúrával:



a) Adja meg a fenti ábrán látható szövegálláshoz tartozó kimeneti adathármast, ha az első buffer hossza 10 és (ahogy az ábra is mutatja) az első buffernél, balról a legelső betű felett helyezkedik el a 10-es koordináta, valamint a második buffer hossza 11.
A betűk kódolására az f() jelet használjuk (pl. a „b” betű kódját „f(b)” jelöli). (15p)

<10, 5, f(b)>

b) Adja meg a szöveg jelenlegi állásához tartozó kimeneti adathármas teljes bináris hosszát, ha 32 db karakterből álló szöveget tömörítünk. (5p)

$\langle p,l,n \rangle \longrightarrow \lceil \log h_s \rceil + \lceil \log h_l \rceil + \lceil \log \chi \rceil$ *=Output, ahol h_s: első buffer, h_l: második buffer, X: kódszó mérete. => Output = 4+4+5 = 13*