COMPUTER SYSTEMS ARCHITECTURE SEMINAR 0x01

SEMINAR SUPPORT NOTES

Cristian Rusu

DECK, EX 1

- what is the amount of information in the deck when we draw the cards for the first time?

DECK, EX 1

- **what is the amount of information in the deck when we draw the cards for the first time?**

- **it is 0, the cards are sorted in ascending order**

- **we randomly shuffle the cards; how much information do we have now?**
  - in how many ways can we arrange the 52 cards?

DECK, EX 1

- **what is the amount of information in the deck when we draw the cards for the first time?**

- **it is 0, the cards are sorted in ascending order**

- **we randomly shuffle the cards; how much information do we have now?**
  - in how many ways can we arrange the 52 cards?
  - 52!
  - therefore, the information is $\log_2(52!)$
    - how do we compute this value?

DECK, EX 1

- **what is the amount of information in the deck when we draw the cards for the first time?**

- **it is 0, the cards are sorted in ascending order**

- **we randomly shuffle the cards; how much information do we have now?**
  - in how many ways can we arrange the 52 cards?
  - 52!
  - therefore, the information is $\log_2(52!)$
    - how do we compute this value?
    - $\log_2(a \times b) = \log_2(a) + \log_2(b)$
    - $\log_2(52!) = 225.6$ bits
    - using Stirling's approximation:

DECK, EX 1

- **what is the amount of information in the deck when we draw the cards for the first time?**

- **it is 0, the cards are sorted in ascending order**

- **we randomly shuffle the cards; how much information do we have now?**
  - in how many ways can we arrange the 52 cards?
  - 52!
  - therefore, the information is $\log_2(52!)$
    - how do we compute this value?
    - $\log_2(a \times b) = \log_2(a) + \log_2(b)$
    - $\log_2(52!) = 225.6$ bits
    - using Stirling's approximation: $\log_2(52!) \approx 52\log_2(52) - 52\log_2 e = 221.4$ bits
  - algorithmically, how do we shuffle the cards efficiently?
    - you have a function that returns a random value in the interval [0,1]

DECK, EX 1

- **what is the amount of information in the deck when we draw the cards for the first time?**

- **it is 0, the cards are sorted in ascending order**

- **we randomly shuffle the cards; how much information do we have now?**
  - algorithmically, how do we shuffle the cards efficiently?
    - you have a function that returns a random value in the interval [0,1]
      - assume the cards are sorted in ascending order
      - compute i = round(52*rand())
      - select card i from the deck
      - swap card i with card 52
      - compute i = round(51*rand())
      - select card i from the deck
      - swap card i with card 51
      - ...
  - see the “Fisher–Yates shuffle” algorithm

URN, EX 2

$$H(X) = E(I(X)) = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} = \sum_{i=1}^{N} -p_i \log_2 p_i$$

- in the urn: 5 red disks, 3 blue disks
- we draw a blue disk; how much information did we receive?

URN, EX 2

$$H(X) = E(I(X)) = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} = \sum_{i=1}^{N} -p_i \log_2 p_i$$

- **in the urn: 5 red disks, 3 blue disks**
- **we draw a blue disk; how much information did we receive?**

$$I(\text{bila albastra}) = \log_2 \left(\frac{1}{\frac{3}{8}} \right) = \log_2 \left(\frac{8}{3} \right) = 1.42 \text{ biti}$$

- **what was the entropy of the urn before the draw?**

URN, EX 2

$$H(X) = E(I(X)) = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} = \sum_{i=1}^{N} -p_i \log_2 p_i$$

- **in the urn: 5 red disks, 3 blue disks**
- **we draw a blue disk; how much information did we receive?**

$$I(\text{bila albastra}) = \log_2 \left(\frac{1}{\frac{3}{8}} \right) = \log_2 \left(\frac{8}{3} \right) = 1.42 \text{ biti}$$

- **what was the entropy of the urn before the draw?**

$$H(\text{urna}) = \frac{5}{8} \log_2 \left(\frac{8}{5} \right) + \frac{3}{8} \log_2 \left(\frac{8}{3} \right) = 0.95 \text{ biti}$$

- **what is the entropy of the urn after the draw?**

URN, EX 2

$$H(X) = E(I(X)) = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} = \sum_{i=1}^{N} -p_i \log_2 p_i$$

- **in the urn: 5 red disks, 3 blue disks**
- **we draw a blue disk; how much information did we receive?**

$$I(\text{bila albastra}) = \log_2 \left(\frac{1}{\frac{3}{8}}\right) = \log_2 \left(\frac{8}{3}\right) = 1.42 \text{ biti}$$

- **what was the entropy of the urn before the draw?**

$$H(\text{urna}) = \frac{5}{8} \log_2 \left(\frac{8}{5}\right) + \frac{3}{8} \log_2 \left(\frac{8}{3}\right) = 0.95 \text{ biti}$$

- **what is the entropy of the urn after the draw?**

$$H(\text{urna dupa extragere}) = \frac{5}{7} \log_2 \left(\frac{7}{5}\right) + \frac{2}{7} \log_2 \left(\frac{7}{2}\right) = 0.86 \text{ biti}$$

- **additional question: continue the entropy calculation assuming that we keep drawing all blue disks, one by one**
- **additional question: repeat the entropy calculation assuming that we draw all red disks from the original urn, one by one**

BALANCE SCALE, EX 3



- **12 disks, one of them has a different weight from the others**
- **find the odd disk using the minimum number of weighings**

- **initial questions:**
  - how many weighings do we need with the “simple” solution?
    - we weigh the disks two at a time (6 experiments): one experiment will reduce the options to two disks, then we compare each disk with one that is certainly standard (2 experiments)
    - therefore, with 8 weighings we can certainly identify the disk
  - how much information do we need to discover?

BALANCE SCALE, EX 3



- **12 disks, one of them has a different weight from the others**
- **find the odd disk using the minimum number of weighings**

- **initial questions:**
  - how many weighings do we need with the “simple” solution?
    - we weigh the disks two at a time (6 experiments): one experiment will reduce the options to two disks, then we compare each disk with one that is certainly standard (2 experiments)
    - therefore, with 8 weighings we can certainly identify the disk
  - how much information do we need to discover?
    - which disk is different? log2(12) = 3.58
    - whether it is lighter or heavier: 1 bit, if we do not know which case applies

BALANCE SCALE, EX 3

consider the simpler case, in which we know whether the odd disk is heavier/lighter

- **let us analyze this systematically**
  - we have 3 values that must be specified when performing a weighing
    - how many disks we place on the left pan
    - how many disks we place on the right pan
    - how many disks remain on the table
  - all relevant possibilities:

BALANCE SCALE, EX 3

consider the simpler case, in which we know whether the odd disk is heavier/lighter

- **let us analyze this systematically**
  - we have 3 values that must be specified when performing a weighing
    - how many disks we place on the left pan
    - how many disks we place on the right pan
    - how many disks remain on the table
  - all relevant possibilities:
    - 6 vs 6, 0 on the table
    - 5 vs 5, 2 on the table
    - 4 vs 4, 4 on the table
    - 3 vs 3, 6 on the table
    - 2 vs 2, 8 on the table
    - 1 vs 1, 10 on the table

BALANCE SCALE, EX 3

consider the simpler case, in which we know whether the odd disk is heavier/lighter

- **6 vs 6, 0 on the table**
  - what are the possible outcomes?
    - the scale tips to the left
    - the scale tips to the right
    - the scale is balanced
  - what are the probabilities of each outcome?

BALANCE SCALE, EX 3

consider the simpler case, in which we know whether the odd disk is heavier/lighter

- **6 vs 6, 0 on the table**
  - what are the possible outcomes?
    - the scale tips to the left (1/2)
    - the scale tips to the right (1/2)
    - the scale is balanced (0)
  - what is the entropy of this alphabet?

$$H(6 \text{ cu } 6) = \frac{1}{2}\log_2(2) + \frac{1}{2}\log_2(2) = 1 \text{ bit}$$

BALANCE SCALE, EX 3

consider the simpler case, in which we know whether the odd disk is heavier/lighter

- **5 vs 5, 2 on the table**
  - what are the possible outcomes?
    - the scale tips to the left (5/12)
    - the scale tips to the right (5/12)
    - the scale is balanced (2/12 = 1/6)
  - what is the entropy of this alphabet?

$$H(5 \text{ cu } 5) = 2 \times \frac{5}{12} \log_2 \left(\frac{12}{5} \right) + \frac{1}{6} \log_2(6) = 1.48 \text{ biti}$$

BALANCE SCALE, EX 3

consider the simpler case, in which we know whether the odd disk is heavier/lighter

- **4 vs 4, 4 on the table**
  - what are the possible outcomes?
    - the scale tips to the left (1/3)
    - the scale tips to the right (1/3)
    - the scale is balanced (1/3 = 4/12)
  - what is the entropy of this alphabet?

$$H(4 \text{ cu } 4) = 3 \times \frac{1}{3} \log_2(3) = 1.58 \text{ biti}$$

BALANCE SCALE, EX 3

consider the simpler case, in which we know whether the odd disk is heavier/lighter

- **3 vs 3, 6 on the table**
  - what are the possible outcomes?
    - the scale tips to the left (1/4)
    - the scale tips to the right (1/4)
    - the scale is balanced (1/2 = 6/12)
  - what is the entropy of this alphabet?

$$H(3 \text{ cu } 3) = 2 \times \frac{1}{4}\log_2(4) + \frac{1}{2}\log_2(2) = 1.5 \text{ biti}$$

BALANCE SCALE, EX 3

consider the simpler case, in which we know whether the odd disk is heavier/lighter

- **Summary:**
  - 6 vs 6, 0 on the table H = 1 bit
  - 5 vs 5, 2 on the table H = 1.48 bits
  - 4 vs 4, 4 on the table H = 1.58 bits
  - 3 vs 3, 6 on the table H = 1.5 bits

  - what do we choose?

BALANCE SCALE, EX 3

consider the simpler case, in which we know whether the odd disk is heavier/lighter

- **Summary:**
  - 6 vs 6, 0 on the table H = 1 bit
  - 5 vs 5, 2 on the table H = 1.48 bits
  - 4 vs 4, 4 on the table H = 1.58 bits
  - 3 vs 3, 6 on the table H = 1.5 bits

  - what do we choose?

BALANCE SCALE, EX 3

consider the simpler case, in which we know whether the odd disk is heavier/lighter

- **4 vs 4, 4 on the table H = 1.58 bits**
  - we perform the experiment; what are the possible outcomes?
    - (case 1) the scale tips to the left
      - the disks on the left side are heavier
      - the disks on the right side are lighter
      - the disks on the table have the same weight
    - (case 2) the scale tips to the right
      - the disks on the right side are heavier
      - the disks on the left side are lighter
      - the disks on the table have the same weight
    - (case 3) the scale is balanced
      - the disks on the table include the odd disk
  - if case 3 occurs, we have reduced the number of disks to check to 4
  - if case 1 or 2 occurs, we do not know whether the disk is on the left or right, but we know for certain that it is not on the table

BALANCE SCALE, EX 3

consider the simpler case, in which we know whether the odd disk is heavier/lighter

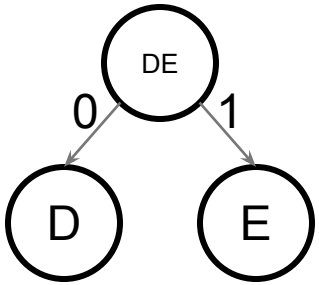
- **4 vs 4, 4 on the table H = 1.58 bits**
  - (case 3) the scale is balanced
    - the disks on the table include the odd disk
  - let us number the disks:
    - on the right side: 1 2 3 4
    - on the left side: 5 6 7 8
    - on the table: 9 10 11 12
  - what weighing comes next?
    - 1 2 3 9 vs 4 5 10 11
      - if the scale is balanced, disk 12 is the odd disk (heavier or lighter)
      - if the scale tips to the left
        - either 9 is the odd disk and is heavier
        - or 10 or 11 is the odd disk and one of them is lighter
        - we weigh 10 vs 11
          - if they are equal, 9 is the odd disk and is certainly heavier
          - if they are not equal, the lighter one is the odd disk
      - if the scale tips to the right ... the procedure is similar
  - continue on your own ... with the remaining cases

SHORT QUESTIONS, EX 5

- a) $p = C_2^N / 2^N$
- b) $p = C_{N/2}^N / 2^N$, assuming N is even
- c) $p = (¾N + 1) / 2^N$, assuming N is divisible by 4
- d) $p = 2^{N-1} / 2^N = 1 / 2$
- e) $p = 2^{N-1} / 2^N = 1 / 2$
- f) $p = N / 2^N$
- g) $p = 2^{N/2} / 2^N = 1 / 2^{N/2}$, assuming N is even
- h) $p \approx [(2^N - 1) / \ln(2^N - 1)] / 2^N \approx [2^N / \ln(2^N)] / 2^N = 1 / \ln(2^N)$
- i) $p = (1 + \sum_{i=1}^{N/2} C_{2i}^N) / 2^N = 1 / 2$, assuming N is even
- j) $p = 1 / 2^N$

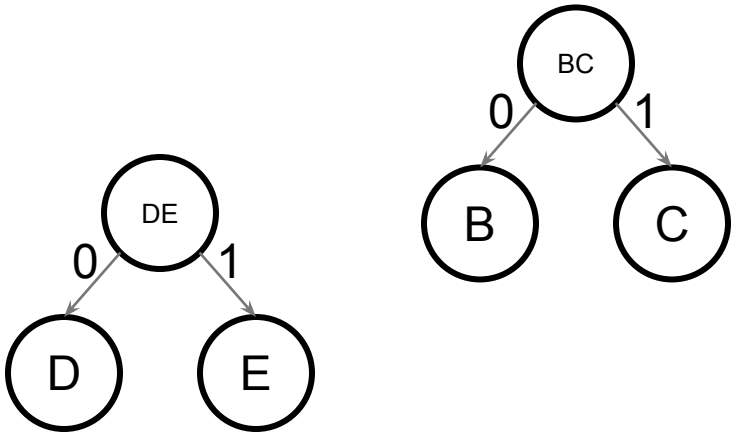
https://en.wikipedia.org/wiki/Prime-counting_function

HUFFMAN, EX 6



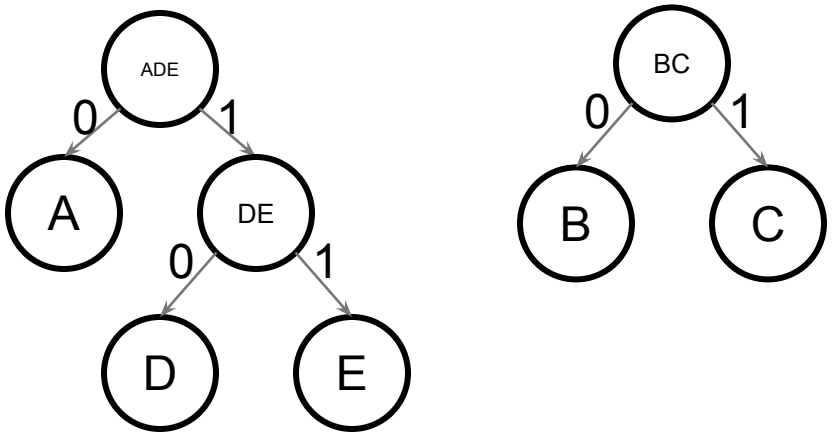
take the two least probable elements (D and E; C and E would also be fine)
the probability of DE is 1/6 + 1/12 = 3/12 = 1/4

HUFFMAN, EX 6



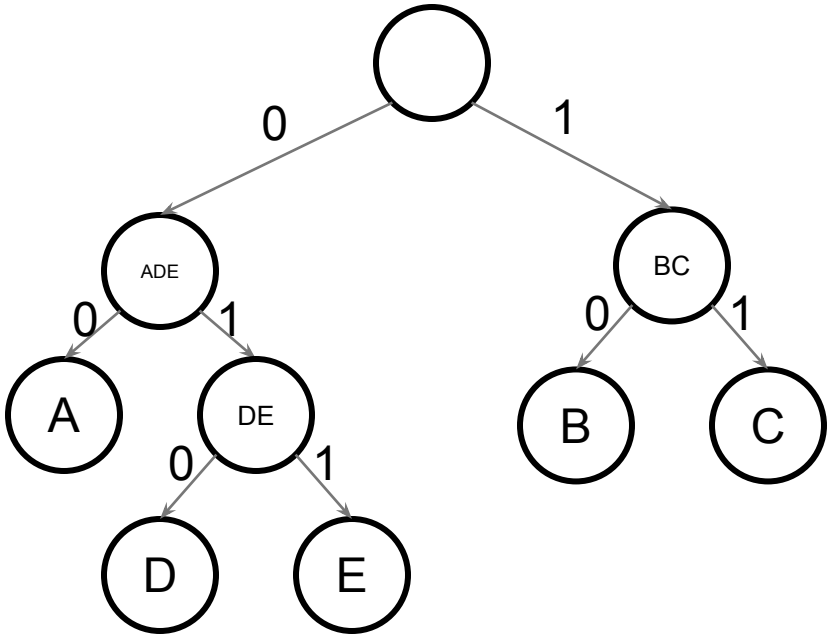
the next element to add is C, with probability 1/6
but 1/6 is smaller than the probability of DE, so C will not stand alone
the probability of BC is ¼ + 1/6 = 10/24 = 5/12

HUFFMAN, EX 6



the next element is A and it has probability 1/3
it does not matter which tree we attach it to, either together with DE or with BC
(if you place it next to BC, A gets a different code but with the same length)

HUFFMAN, EX 6



the complete tree and the codes:

A = 00

B = 10

C = 11

D = 010

E = 011

ERROR-CORRECTING CODES, EX 7

- everything marked D represents data bits
- everything marked P represents parity bits

D_{00}	D_{01}	D_{02}	$P_{0\ell}$
D_{10}	D_{11}	D_{12}	$P_{1\ell}$
D_{20}	D_{21}	D_{22}	$P_{2\ell}$
P_{c0}	P_{c1}	P_{c2}	$P_{c\ell}$

- suppose D01 changes (from 0 to 1, or vice versa)
- how many bits in the message change?

ERROR-CORRECTING CODES, EX 7

- **everything marked D represents data bits**
- **everything marked P represents parity bits**

D_{00}	D_{01}	D_{02}	$P_{0\ell}$
D_{10}	D_{11}	D_{12}	$P_{1\ell}$
D_{20}	D_{21}	D_{22}	$P_{2\ell}$
P_{c0}	P_{c1}	P_{c2}	$P_{c\ell}$

- **suppose D01 changes (from 0 to 1, or vice versa)**
- **how many bits in the message change?**
  - 3 parity bits + the data bit
- **what is the minimum Hamming distance?**

ERROR-CORRECTING CODES, EX 7

- **everything marked D represents data bits**
- **everything marked P represents parity bits**

D_{00}	D_{01}	D_{02}	$P_{0\ell}$
D_{10}	D_{11}	D_{12}	$P_{1\ell}$
D_{20}	D_{21}	D_{22}	$P_{2\ell}$
P_{c0}	P_{c1}	P_{c2}	$P_{c\ell}$

- **suppose D01 changes (from 0 to 1, or vice versa)**
- **how many bits in the message change?**
  - 3 parity bits + the data bit
- **what is the minimum Hamming distance?**
  - 4 bits change, therefore the distance is 4

ERROR-CORRECTING CODES, EX 7

- **is this message correct?**
  - if not, why?

D_{00}	D_{01}	D_{02}	$P_{0\ell}$
D_{10}	D_{11}	D_{12}	$P_{1\ell}$
D_{20}	D_{21}	D_{22}	$P_{2\ell}$
P_{c0}	P_{c1}	P_{c2}	$P_{c\ell}$

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

ERROR-CORRECTING CODES, EX 7

- **is this message correct?**
  - if not, why?

D_{00}	D_{01}	D_{02}	$P_{0\ell}$
D_{10}	D_{11}	D_{12}	$P_{1\ell}$
D_{20}	D_{21}	D_{22}	$P_{2\ell}$
P_{c0}	P_{c1}	P_{c2}	$P_{c\ell}$

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

- **it appears correct; all parity bits match what we observe**

ERROR-CORRECTING CODES, EX 7

- **is this message correct?**
  - if not, why?

D_{00}	D_{01}	D_{02}	$P_{0\ell}$
D_{10}	D_{11}	D_{12}	$P_{1\ell}$
D_{20}	D_{21}	D_{22}	$P_{2\ell}$
P_{c0}	P_{c1}	P_{c2}	$P_{c\ell}$

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

ERROR-CORRECTING CODES, EX 7

- **is this message correct?**
  - if not, why?

D_{00}	D_{01}	D_{02}	$P_{0\ell}$
D_{10}	D_{11}	D_{12}	$P_{1\ell}$
D_{20}	D_{21}	D_{22}	$P_{2\ell}$
P_{c0}	P_{c1}	P_{c2}	$P_{c\ell}$

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

- **it is not correct: parity bit Pc1 signals an error, but the row parity bits signal nothing; therefore Pc1 itself was corrupted**

ERROR-CORRECTING CODES, EX 7

- **is this message correct?**
  - if not, why?

D_{00}	D_{01}	D_{02}	$P_{0\ell}$
D_{10}	D_{11}	D_{12}	$P_{1\ell}$
D_{20}	D_{21}	D_{22}	$P_{2\ell}$
P_{c0}	P_{c1}	P_{c2}	$P_{c\ell}$

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

ERROR-CORRECTING CODES, EX 7

- **is this message correct?**
  - if not, why?

D_{00}	D_{01}	D_{02}	$P_{0\ell}$
D_{10}	D_{11}	D_{12}	$P_{1\ell}$
D_{20}	D_{21}	D_{22}	$P_{2\ell}$
P_{c0}	P_{c1}	P_{c2}	$P_{c\ell}$

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

- **it is not correct: parity bit P0ℓ is wrong and parity bit Pc0 is wrong => data bit D00 is wrong, so it must be changed from 0 to 1; check: total parity bit Pcℓ is 0, so the message must contain an odd number of 1 bits (therefore, after the change it is correct)**

ERROR-CORRECTING CODES, EX 7

- **is this message correct?**
  - if not, why?

D_{00}	D_{01}	D_{02}	$P_{0\ell}$
D_{10}	D_{11}	D_{12}	$P_{1\ell}$
D_{20}	D_{21}	D_{22}	$P_{2\ell}$
P_{c0}	P_{c1}	P_{c2}	$P_{c\ell}$

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

ERROR-CORRECTING CODES, EX 7

- **is this message correct?**
  - if not, why?

D_{00}	D_{01}	D_{02}	$P_{0\ell}$
D_{10}	D_{11}	D_{12}	$P_{1\ell}$
D_{20}	D_{21}	D_{22}	$P_{2\ell}$
P_{c0}	P_{c1}	P_{c2}	$P_{c\ell}$

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

- **it is not correct: all row and column parity bits are OK, but total parity bit Pcℓ is not; therefore the erroneous bit is Pcℓ itself**

ERROR-CORRECTING CODES, EX 7

- **is this message correct?**
  - if not, why?

D_{00}	D_{01}	D_{02}	$P_{0\ell}$
D_{10}	D_{11}	D_{12}	$P_{1\ell}$
D_{20}	D_{21}	D_{22}	$P_{2\ell}$
P_{c0}	P_{c1}	P_{c2}	$P_{c\ell}$

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

ERROR-CORRECTING CODES, EX 7

- **is this message correct?**
  - if not, why?

D_{00}	D_{01}	D_{02}	$P_{0\ell}$
D_{10}	D_{11}	D_{12}	$P_{1\ell}$
D_{20}	D_{21}	D_{22}	$P_{2\ell}$
P_{c0}	P_{c1}	P_{c2}	$P_{c\ell}$

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

- **it is not correct: parity bits P1ℓ, P2ℓ, Pc1, and Pc2 are all wrong, while total parity bit Pcℓ is correct => the error is somewhere on rows/columns 1 and 2; therefore we can detect 2-bit errors but cannot correct them**

ENTROPY RATE, EX 10

- **average message length**
- **entropy in our case**
- **entropy rate**

ENTROPY RATE, EX 10

- **average message length**
  - $1xp + 2x(1-p) = 2 - p$
- **entropy in our case**
- **entropy rate**

ENTROPY RATE, EX 10

- **average message length**
  - 1xp + 2x(1-p) = 2 – p

- **entropy in our case**

$$H = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$$

- **entropy rate**

ENTROPY RATE, EX 10

- **average message length**
  - 1xp + 2x(1-p) = 2 – p

- **entropy in our case**

$$H = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$$

- **entropy rate**

$$R = \frac{p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}}{2-p}$$

- **how do we maximize it?**

ENTROPY RATE, EX 10

- **average message length**
  - 1xp + 2x(1-p) = 2 – p
- **entropy in our case**

$$H = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$$

- **entropy rate**

$$R = \frac{p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}}{2-p}$$

- **how do we maximize it?**
  - differentiate $R' = \frac{2 \log_2 \frac{1}{p} - \log_2 \frac{1}{1-p}}{(2-p)^2}$ and set it equal to zero

ENTROPY RATE, EX 10

- **average message length**
  - 1x*p* + 2x(1-*p*) = 2 – *p*

- **entropy in our case**

$$H = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$$

- **entropy rate**

$$R = \frac{p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}}{2-p}$$

- **how do we maximize it?**
  - differentiate $R' = \frac{2 \log_2 \frac{1}{p} - \log_2 \frac{1}{1-p}}{(2-p)^2}$ and set it equal to zero $\log_2 \frac{1-p}{p^2} = 0$
  - *p is the solution of the equation*

ENTROPY RATE, EX 10

- **average message length**
  - 1x*p* + 2x(1-*p*) = 2 – *p*

- **entropy in our case**

$$H = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$$

- **entropy rate**

$$R = \frac{p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}}{2-p}$$

- **how do we maximize it?**
  - differentiate $R' = \frac{2 \log_2 \frac{1}{p} - \log_2 \frac{1}{1-p}}{(2-p)^2}$ and set it equal to zero $\log_2 \frac{1-p}{p^2} = 0$
  - *p is the solution of the equation* $p^2 + p - 1 = 0$

ENTROPY RATE, EX 10

- **average message length**
  - 1xp + 2x(1-p) = 2 – p

- **entropy in our case**

$$H = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$$

- **entropy rate**

$$R = \frac{p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}}{2-p}$$

- **how do we maximize it?**
  - differentiate $R' = \frac{2 \log_2 \frac{1}{p} - \log_2 \frac{1}{1-p}}{(2-p)^2}$ and set it equal to zero $\log_2 \frac{1-p}{p^2} = 0$
  - *p is the solution of the equation* $p^2 + p - 1 = 0 \longrightarrow p = \frac{\sqrt{5}-1}{2}$

what is the connection with the golden ratio?

MAXIMIZING ENTROPY, EX 11

- suppose we have probabilities such that p1 < p2
- we have p1 + ε < p2 - ε for sufficiently small ε
- we want to show that

$$H(\{p_1 + \epsilon, p_2 - \epsilon, p_3, \ldots, p_N\}) > H(\{p_1, p_2, p_3, \ldots, p_N\})$$

MAXIMIZING ENTROPY, EX 11

- suppose we have probabilities such that p1 < p2
- we have p1 + ε < p2 - ε for sufficiently small ε
- we want to show that

$$H(\{p_1+\epsilon, p_2-\epsilon, p_3, \ldots, p_N\}) > H(\{p_1, p_2, p_3, \ldots, p_N\})$$

- we need to show that

$$H(\{p_1+\epsilon, p_2-\epsilon, p_3, \ldots, p_N\}) - H(\{p_1, p_2, p_3, \ldots, p_N\}) > 0$$

- the difference between the two entropies can be written as

MAXIMIZING ENTROPY, EX 11

- suppose we have probabilities such that p1 < p2
- we have p1 + ε < p2 - ε for sufficiently small ε
- we want to show that

$$H(\{p_1+\epsilon, p_2-\epsilon, p_3, \ldots, p_N\}) > H(\{p_1, p_2, p_3, \ldots, p_N\})$$

- we need to show that

$$H(\{p_1+\epsilon, p_2-\epsilon, p_3, \ldots, p_N\}) - H(\{p_1, p_2, p_3, \ldots, p_N\}) > 0$$

- the difference between the two entropies can be written as

$$-p_1 \log_2\left(\frac{p_1+\epsilon}{p_1}\right) - \epsilon \log_2(p_1+\epsilon) - p_2 \log_2\left(\frac{p_2-\epsilon}{p_2}\right) + \epsilon \log_2(p_2-\epsilon) > 0$$

- now, we want to use $\log_2(1+x) \approx x + O(x^2)$

MAXIMIZING ENTROPY, EX 11

- **the difference between the two entropies can be written as**

$$-p_1 \log_2 \left(\frac{p_1 + \epsilon}{p_1} \right) - \epsilon \log_2 (p_1 + \epsilon) - p_2 \log_2 \left(\frac{p_2 - \epsilon}{p_2} \right) + \epsilon \log_2 (p_2 - \epsilon) > 0$$

- **now, we want to use** $\log_2(1 + x) \approx x + O(x^2)$
- **on the left-hand side of the inequality, we obtain:**

MAXIMIZING ENTROPY, EX 11

- **the difference between the two entropies can be written as**

$$-p_1 \log_2 \left(\frac{p_1 + \epsilon}{p_1} \right) - \epsilon \log_2 (p_1 + \epsilon) - p_2 \log_2 \left(\frac{p_2 - \epsilon}{p_2} \right) + \epsilon \log_2 (p_2 - \epsilon) > 0$$

- **now, we want to use** $\log_2(1 + x) \approx x + O(x^2)$
- **on the left-hand side of the inequality, we obtain:**

$$-\epsilon - \epsilon \log_2 p_1 + \epsilon + \epsilon p_2 + O(\epsilon^2) = \epsilon \log_2 \frac{p_2}{p_1} + O(\epsilon^2)$$

- **is this expression always positive?**

MAXIMIZING ENTROPY, EX 11

- **the difference between the two entropies can be written as**

$$-p_1 \log_2 \left(\frac{p_1 + \epsilon}{p_1} \right) - \epsilon \log_2 (p_1 + \epsilon) - p_2 \log_2 \left(\frac{p_2 - \epsilon}{p_2} \right) + \epsilon \log_2 (p_2 - \epsilon) > 0$$

- **now, we want to use** $\log_2(1 + x) \approx x + O(x^2)$
- **on the left-hand side of the inequality, we obtain:**

$$-\epsilon - \epsilon \log_2 p_1 + \epsilon + \epsilon p_2 + O(\epsilon^2) = \epsilon \log_2 \frac{p_2}{p_1} + O(\epsilon^2)$$

- **is this expression always positive?**
  - yes, because we assumed p1 < p2

PREFIX CODING, EX 12

- **examples of codes:**

PREFIX CODING, EX 12

- **examples of codes:**
  - 1
  - 01
  - 001
  - 0001
  - ...

- **regardless of pi, each symbol has one more bit than the previous one, but the most frequent symbols still receive shorter codes than the less frequent ones**
- **entropy**
- **average code length**

PREFIX CODING, EX 12

- **examples of codes:**
  - 1
  - 01
  - 001
  - 0001
  - ...

- **regardless of pi, each symbol has one more bit than the previous one, but the most frequent symbols still receive shorter codes than the less frequent ones**
- **entropy** $H = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i}$
- **average code length** $L = \sum_{i=1}^{N} i p_i$

- **when do we have H = L?**

PREFIX CODING, EX 12

- **examples of codes:**
  - 1
  - 01
  - 001
  - 0001
  - ...

- **regardless of pi, each symbol has one more bit than the previous one, but the most frequent symbols still receive shorter codes than the less frequent ones**
- **entropy** $H = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i}$
- **average code length** $L = \sum_{i=1}^{N} i p_i$

- **when do we have H = L?** $\sum_{i=1}^{N} p_i (i - \log_2 \frac{1}{p_i}) = 0 \longrightarrow i = \log_2 \frac{1}{p_i}$, deci $p_i = 2^{-i}$

$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots = 1$ pentru $N \to \infty$