ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer and Communication Sciences

Handout 19 Information Theory and Coding
Midterm December 17, 2002

PROBLEM 1.

(a) Solution 1:

$$\sum_x 2^{-l(x)} = \sum_x 2^{-[\lceil \log_2 M\rceil + \min_m l_m(x)]} \le \sum_x 2^{-[\log_2 M + \min_m l_m(x)]}$$
$$= \frac{1}{M}\sum_x \max_m 2^{-l_m(x)} \le \frac{1}{M}\sum_x \sum_{m=1}^M 2^{-l_m(x)} = \frac{1}{M}\sum_{m=1}^M \sum_x 2^{-l_m(x)} \le \frac{1}{M}\sum_{m=1}^M 1 = 1.$$

Solution 2: We can construct a uniquely decodable (in fact prefix free) code with length $l(x)$: Given a symbol x, let m^* be such that $l_{m^*}(x) = \min_m l_m(x)$. Assign to x the codeword whose first $\lceil \log_2 M\rceil$ bits describe m^*, and the rest of the $l_{m^*}(x)$ bits is the encoding of the x with the m^*th prefix-free code. The code is clearly uniquely decodable, and so its codewords lengths satisfy the Kraft inequality. But the code encodes x using $l(x)$ bits, so the conclusion follows.

(b) Since $\min_m l_m(x) \le l_m(x)$ for any m, the inequality follows immediately.

(c) Let the mth codeword be the Huffman code for the distribution p_m. We then know that $p_m(x)l_m(x) < H_m + 1$ where H_m denotes the entropy of the distribution p_m. (Alternatively we could have taken $l_m(x) = \lceil -\log_2 p_m(x)\rceil$.) Let p_{m^*} be the true distribution so that $H(X) = H_{m^*}$. By part (b),

$$\sum_x p_{m^*}(x)l(x) \le \lceil \log_2 M\rceil + \sum_x p_{m^*}l_{m*}(x) < \lceil \log_2 M\rceil + H(X) + 1.$$

[If one applies this coding technique to blocks of source symbols, by encoding n source letters at a time, we see that the number of bits per source letter is upper bounded by

$$\frac{1}{n}H(X_1,\ldots,X_n) + \frac{1}{n}[1 + \lceil \log_2 M\rceil].$$

For large n the second term approaches zero, and for a stationasy source the first term approaches the entropy rate. We thus see that this technique performs asymptotically as well as a technique that knows the true probability distribution in advance.]

PROBLEM 2.

(a) Since the coin is fair, $P(X = 0) = P(X = 1) = 1/2$ and thus $H(X) = 1$ bit. On the other hand $H(X|Y = 0) = H(X|Y = 1) = 1/4\log_2 4 + 3/4\log_2(4/3) = 2 - 3/4\log_2 3$ and thus $I(X;Y) = 3/4\log_2 3 - 1$.

(b) At each bet, if we guess correctly, our fortune is $2(1-q)+q = 2-q$ times our original fortune, if we guess wrong our fortune is q times our original fortune. So, at the ith bet our fortune is multiplied by

$$(2-q)^{Z_i} q^{1-Z_i},$$

and the result follows.

(c) Since Z_i are i.i.d., $E[C_n] = C_0 E\big[\prod_{i=1}^n (2-q)^{Z_i} q^{1-Z_i}\big] = C_0 \prod_{i=1}^n E\big[(2-q)^{Z_i} q^{1-Z_i}\big] = C_0\big[\frac{3}{4}(2-q) + \frac{1}{4}q\big]^n = C_0[3/2 - q/2]^n$, and thus the value of q that maximizes $E[C_n]$ is $q = 0$.

(d) Observe that

$$R_n = \frac{1}{n}\sum_{i=1}^n \log_2\big[(2-q)^{Z_i} q^{1-Z_i}\big]$$

is a sum of i.i.d. random variables, and so

$$E[R_n] = E\big[\log_2[(2-q)^{Z_1} q^{1-Z_1}]\big] = \tfrac{3}{4}\log_2(2-q) + \tfrac{1}{4}\log_2 q.$$

Letting $F(q) = \frac{3}{4}\log_2(2-q) + \frac{1}{4}\log_2 q$, the value of q that maximizes $E[R_n]$ is found by setting the derivative of F equal to zero:

$$-\frac{3}{4}\frac{1}{2-q} + \frac{1}{4}\frac{1}{q} = 0,$$

which yields $q = 1/2$. With this value of q, $E[R_n] = I(X;Y)$.

(e) The law of large numbers applies to R_n, so that with probability 1,

$$\lim_{n\to\infty} R_n = \tfrac{3}{4}\log_2(2-q) + \tfrac{1}{4}\log_2 q.$$

Thus for large n, our fortune is close to $2^{nF(q)}$ with high probability, and so we should be choosing the value of q which maximizes $F(q)$, namely $1/2$. [In fact if we had chosen $q = 0$, we would have lost all our money as soon as we guess wrong, which is sure to happen eventually.]

PROBLEM 3.

(a) Since there are only 2^k distinct binary sequence of length k, if the code assigned more than 2^k of the symbols to binary sequences of length k, it cannot be non-singular, so (1) is necessary for the code to be non-singular. On the other hand, if we are given a length function that satisfies (1), we can assign to each symbol x a different binary sequence of length $l(x)$: since for every k there are enough binary sequences of length k to make sure that if $l(x) = l(y) = k$ then $C(x) \neq C(y)$. (If $l(x) \neq l(y)$ then $C(x) \neq C(y)$ is automatically true.)

(b) Assume to the contrary, that C is a non-singular code with least average length L and there is x and y for which $l(x) > l(y)$ and $p(x) > p(y)$. Consider a new code C' obtained from C by exchanging the codewords for the symbols i and j and let L' be its average length. Then

$$L' - L = p(x)l(y) + p(y)l(x) - p(x)l(x) - p(y)l(y) = [p(x) - p(y)][l(y) - l(x)] < 0$$

contradicting the premise that C has least average length.

(c) Since the source alphabet is of size K, it is clear that the non-singular code of least average length will only use the K shortest distinct binary sequences as the set of possible codes, namely the first K elements of the sequence λ, 0, 1, 00, 01, ... From the previous part we know that more probable letters should get shorter codes, and so we see that a code with shortest average length will assign to the ith source letter the ith element of the above sequence. By the hint, this element has length $\lfloor \log_2 i \rfloor$, and the conclusion follows.

(d) By the part above, the least average length is $\sum_{i=1}^{K} p(i)\lfloor \log_2 i \rfloor$, so for any non-singular code

$$\begin{aligned} L &\geq \sum_{i=1}^{K} p(i)\lfloor \log_2 i \rfloor \\ &\geq \sum_{i=1}^{K} p(i)[(\log_2 i) - 1] \\ &= -1 + \sum_{i=1}^{K} p(i) \log_2 i \\ &= -1 + \sum_{i=1}^{K} p(i) \log_2 \frac{ip(i)}{p(i)} \\ &= -1 + \sum_{i=1}^{K} p(i) \log_2 \frac{1}{p(i)} + \sum_{i=1}^{K} p(i) \log_2 [ip(i)] \\ &= H(X) - 1 - \sum_{i=1}^{K} p(i) \log_2 \frac{1}{ip(i)}. \end{aligned}$$

(e) Since the both sides scale by a constant when we change the base of the logarithm it suffices to prove the result for the natural logarithm:

$$\begin{aligned} \sum_{i=1}^{K} p(i) \ln \frac{1}{ip(i)} - \ln S_K &= \sum_{i=1}^{K} p(i) \ln \frac{1}{ip(i)S_K} \\ &\leq \sum_{i=1}^{K} p(i) \left[\frac{1}{ip(i)S_K} - 1 \right] = \frac{1}{S_K} \sum_{i=1}^{K} \frac{1}{i} - \sum_{i=1}^{K} p(i) = 1 - 1 = 0. \end{aligned}$$

(f) Putting (d) and (e) together we obtain the desired result.

[Observe that if one applies the bound to a non-singular code for the alphabet $\mathcal{X}^n$, then we find that the number of bits per source letter such a code emits is lower bounded by

$$\frac{1}{n} H(X_1, \ldots, X_n) - \frac{1}{n}[1 + \log_2(1 + n \ln K)].$$

As n gets large, the second term approaches zero, and for a stationary source the first term approaches the entropy rate. So, we see that for large block lenghts the non-singular codes cannot beat uniquely decodable codes.]