ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
Communication Systems Department

Handout 16 Information Theory and Coding
Solutions to Midterm December 4, 2000

Problem 1.

(a) By the chain rule, the left hand side and the right hand side both equal $H(E, M|Y)$.

(b) Since E is a function of M and Y, we have $H(E|M, Y) = 0$.

(c) $H(M|E, Y) = \Pr(E = 0)H(M|Y, E = 0) + \Pr(E = 1)H(M|Y, E = 1)$. But $H(M|Y, E = 0)$ is zero since when $E = 0$, $g(Y) = M$ and thus Y determines M, so we have (i). On the other hand, given Y and $E = 1$, we know that M can take on all values except $g(Y)$. Thus M can take on at most $|\mathcal{M}| - 1$ values and its entropy can be at most $\log(|\mathcal{M}| - 1)$.

(d) Conditioning does not increase entropy, hence $H(E|Y) \leq H(E)$.

(e) E takes on the value 1 when and only when $\hat{M} \neq M$. This event has probability P_e, so $\Pr(E = 1) = P_e$, and $\Pr(E = 0) = 1 - P_e$. We then conclude that $H(E) = -P_e \log P_e - (1 - P_e) \log(1 - P_e) = h(P_e)$.

(f) We have

$$
\begin{aligned}
H(M|Y) + H(E|M, Y) &= H(E|Y) + H(M|E, Y) && \text{from (a)} \\
H(M|Y) &= H(E|Y) + H(M|E, Y) && \text{from (b)} \\
H(M|Y) &\leq H(E|Y) + \Pr(E = 1) \log(|\mathcal{M}| - 1) && \text{from (c)} \\
H(M|Y) &\leq H(E) + \Pr(E = 1) \log(|\mathcal{M}| - 1) && \text{from (d)} \\
H(M|Y) &\leq h(P_e) + P_e \log(|\mathcal{M}| - 1) && \text{from (e).}
\end{aligned}
$$

Problem 2.

(a) We have

(i) $H(X|Y) = H(X)$ since X and Y are independent.

(ii) $H(X|K) = H(X)$ since X and K are independent.

(iii) $H(Y|X, K) = 0$ since X and K determine Y.

(iv) $H(X|Y, K) = 0$ since Y and K determine X by the decryptability condition.

(v) $I(X; Y|K) = H(X|K) - H(X|Y, K) = H(X)$ by (iv) and (ii).

(vi) $H(Y|K) = I(X; Y|K) + H(Y|X, K) = H(X)$ by (v) and (iii).

(b) Suppose k a key common to both $\mathcal{K}(x_1)$ and $\mathcal{K}(x_2)$. Then, the pair y_0, k can be decrypted as either x_1 or x_2, contradicting the decryptability condition.

(c) Since $I(X;Y) = 0$ we know that X and Y are independent and thus, $\Pr(Y = y) = \Pr(Y = y|X = x)$ for all x and y. In particular

$$0 < \Pr(Y = y_0) = \Pr(Y = y_0|X = x).$$

Thus for each x, $\mathcal{K}(x)$ is not empty, for otherwise $\Pr(Y = y_0|X = x)$ would have been zero. If any $\mathcal{K}(x)$ had more than one element, then the total number of keys would exceed the number of source letters; thus each $\mathcal{K}(x)$ must have exactly one element.

(d) Given that $X = x$, the only way $Y = y_0$ is when $K = k(x)$. Since X and K are independent this happens with probability $\Pr(K = k(x))$.

(e) We have $\Pr(Y = y_0) = \Pr(Y = y_0|X = x) = \Pr(K = k(x))$. Since the left hand side does not depend on x, the same must be true for the right hand side. Since $k(x)$ exhausts all the keys as x ranges over the source letters, we see that $\Pr(K = k)$ does not depend on k and hence that K is uniformly distributed.

Problem 3.

(a) If for some i, $q_i < p_{i-1}$, we can exchange the subtrees rooted at q_i and p_{i-1}. This would elongate by 1 the codewords for a set of source letters with probability q_i and shorten by 1 the codewords for a set of source letters with probability p_{i-1}. Since $q_i < p_{i-1}$ this shortens the expected codeword length by $p_{i-1} - q_i$, contradicting the optimality of the Huffman code. [Alternatively, if $q_i < p_{i-1}$, the Huffman procedure would have merged q_{i-1} with q_i, not p_{i-1}.]

(b) We have $p_0 = F_0 p_0$ and $p_1 = q_0 + p_0 \geq F_1 p_0$. Using these facts as our induction base, suppose that $p_n \geq F_n$ for all $n < i$. Then,

$$\begin{aligned}
p_i &= p_{i-1} + q_{i-1} \\
&\geq p_{i-1} + p_{i-2} && \text{part (a)} \\
&\geq F_{i-1}p_0 + F_{i-2}p_0 && \text{induction hypothesis} \\
&= F_i p_0 && \text{Fibonacci recursion}
\end{aligned}$$

completing the the proof by induction.

(c) Since $1 = p_{n_0} \geq F_{n_0} p_0$, the claim follows.

(d) If $q_i = p_{i-1}$ the Huffman procedure can choose to merge q_{i-1} with either q_i or p_{i-1} without loss of optimality. For three source letters, any distribution of the form $\{\alpha, \alpha, 1 - 2\alpha\}$ for $\alpha \geq 1/3$ yields a valid example. For larger source alphabets, $\{1/8, 1/8, 2/8, 4/8\}$, $\{1/16, 1/16, 2/16, 4/16, 8/16\}$, ... are other possible examples.

(e) Since $q_0 > 0$, we have $p_1 > F_1 p_0$ which says that the bound in part (b) (and thus in part (c)) cannot be made to hold with equality. However, by letting $q_i = p_{i-1}$ for $i \geq 1$ as in part (d), one will get $p_i = F_i p_0 + F_{i-1} q_0$ for $i \geq 1$: for $i = 1, 2, 3$, the equality holds (induction base), assuming that it holds for all $n < i$,

$$\begin{aligned}
p_i &= p_{i-1} + q_{i-1} \\
&= p_{i-1} + p_{i-2} \\
&= F_{i-1}p_0 + F_{i-2}q_0 + F_{i-2}p_0 + F_{i-3}q_0 \\
&= F_i p_0 + F_{i-1}q_0
\end{aligned}$$

proving the claim. Now, choose q_0 small enough to approach equality in $p_i \geq F_i p_0$ all i (upto n_0). Same construction yields

$$p_0 = (1 - F_{n_0-1} q_0)/F_{n_0}$$

which can be made arbitrarily close to $1/F_{n_0}$ by taking small enough q_0.

The Fibonacci recursion can be solved to yield $F_n = [\phi^{n+1} - \phi^{-n-1}]/\sqrt{5}$ where $\phi = (1+\sqrt{5})/2$. The last result shows that, for small p_0 one can get

$$n_0 \approx \frac{-\log_2 p_0}{\log_2 \phi} \approx -1.45 \log_2 p_0.$$

In other words, for some source letters, the Huffman procedure can yield a codeword that is much longer than one would expect, $-\log p_0$.