Problem 1.

(a) We have \( H(f(U)) \leq H(f(U), U) = H(U) + H(f(U) | U) = H(U) + 0 = H(U). \)

(b) Notice that \( U \not\rightarrow V \not\rightarrow f(V) \) is a Markov chain. The data processing inequality implies that \( H(U) - H(U | f(V)) = I(U; f(V)) \leq I(U; V) = H(U) - H(U | V). \) Therefore, \( H(U | V) \leq H(U | f(V)). \)

Problem 2.

(a) We have:

\[
H(U | \hat{U}) \leq H(U, W | \hat{U}) = H(W | \hat{U}) + H(U | \hat{U}, W) \leq H(W) + H(U | \hat{U}, W)
\]

\[
= H(W) + H(U | \hat{U}, W = 0) \cdot \mathbb{P}[W = 0] + H(U | \hat{U}, W = 1) \cdot \mathbb{P}[W = 1]
\]

\[
\leq h_2(p_e) + 0 \cdot (1 - p_e) + \log(|U| - 1) \cdot p_e = h_2(p_e) + p_e \log(|U| - 1),
\]

where (*) follows from the following facts:

- \( H(W) = h_2(p_e). \)
- \( H(U | \hat{U}, W = 0) = 0: \) conditioned on \( W = 0, \) we know that \( U = \hat{U} \) and so the conditional entropy \( H(U | \hat{U}, W = 0) \) is equal to 0.
- \( H(U | \hat{U}, W = 1) \leq \log(|U| - 1): \) conditioned on \( W = 1, \) we know that \( U \neq \hat{U} \) and so there are at most \( |U| - 1 \) values for \( U. \) Therefore, the conditional entropy \( H(U | \hat{U}, W = 0) \) is at most \( \log(|U| - 1). \)

(b) Let \( \hat{U} = f(V). \) We have \( H(U | \hat{U}) \leq h_2(p_e) + p_e \log(|U| - 1) \) from (a). On the other hand, from Problem 1(b) we have \( H(U | V) \leq H(U | f(V)) = H(U | \hat{U}). \) We conclude that \( H(U | V) \leq h_2(p_e) + p_e \log(|U| - 1). \)

Problem 3.

(a) Since

\[
P(U = u, Z = z) = \begin{cases} p(u) & \text{if } z = 1, \\ q(u) & \text{if } z = 2,
\end{cases}
\]

one can immediately see that the distribution of \( U \) is \( r(u) = \theta p(u) + (1 - \theta)q(u). \)

(b) \( H(U) = h(r), \) and

\[
H(U | Z) = \sum_z P(Z = z)H(U | Z = z) = \theta h(p) + (1 - \theta)h(q).
\]

The last equality follows since given \( z = 1 \) (resp. \( z = 2 \)) \( U \) has distribution \( p \) (resp. \( q \)).

Since \( H(U) \geq H(U | Z), \) we have proved that \( h(r) \geq \theta h(p) + (1 - \theta)h(q). \)
Problem 4.

(a) We have:

\[ S = \sum_{u \in \mathcal{U}} \max\{P_1(u), P_2(u)\} \leq \sum_{u \in \mathcal{U}} (P_1(u) + P_2(u)) \]

\[ = \sum_{u \in \mathcal{U}} P_1(u) + \sum_{u \in \mathcal{U}} P_2(u) = 1 + 1 = 2, \]

It is easy to see from (*) that \( S = 2 \) if and only if \( \max\{P_1(u), P_2(u)\} = P_1(u) + P_2(u) \)
for all \( u \in \mathcal{U} \), which is equivalent to say that there is no \( u \in \mathcal{U} \) for which we have
\( P_1(u) > 0 \) and \( P_2(u) > 0 \). In other words, \( S = 2 \) if and only if
\[ \{u \in \mathcal{U} : P_1(u) > 0\} \cap \{u \in \mathcal{U} : P_2(u) > 0\} = \emptyset. \]

(b) Let \( l_i = \lceil \log_2 \frac{S}{\max\{P_1(a_i), P_2(a_i)\}} \rceil \), and let us compute the Kraft sum:

\[ \sum_{i=1}^{M} 2^{-l_i} \leq \sum_{i=1}^{M} 2^{-\log_2 \frac{S}{\max\{P_1(a_i), P_2(a_i)\}}} = \sum_{i=1}^{M} \frac{\max\{P_1(a_i), P_2(a_i)\}}{S} = 1. \]

Since the Kraft sum is at most 1, there exists a prefix-free code where the length of
the codeword associated to \( a_i \) is \( l_i \).

(c) Since the code constructed in (b) is prefix free, it must be the case that \( l \geq H(U) \).
In order to prove the upper bounds, let \( P^* \) be the true distribution (which is either
\( P_1 \) or \( P_2 \)). It is easy to see that \( P^*(a_i) \leq \max\{P_1(a_i), P_2(a_i)\} \) for all \( 1 \leq i \leq M \). We have:

\[ l = \sum_{i=1}^{M} P^*(a_i).l_i = \sum_{i=1}^{M} P^*(a_i).\lceil \log_2 \frac{S}{\max\{P_1(a_i), P_2(a_i)\}} \rceil \]

\[ < \sum_{i=1}^{M} P^*(a_i).\left(1 + \log_2 \frac{S}{\max\{P_1(a_i), P_2(a_i)\}}\right) \]

\[ = \sum_{i=1}^{M} P^*(a_i).\left(1 + \log S + \log_2 \frac{1}{\max\{P_1(a_i), P_2(a_i)\}}\right) \]

\[ = 1 + \log S + \sum_{i=1}^{M} P^*(a_i).\log_2 \frac{1}{\max\{P_1(a_i), P_2(a_i)\}} \]

\[ \leq 1 + \log S + \sum_{i=1}^{M} P^*(a_i).\log_2 \frac{1}{P^*(a_i)} = H(U) + \log S + 1 \leq H(U) + 2, \]

where the inequality (*) uses the fact that \( P^*(a_i) \leq \max\{P_1(a_i), P_2(a_i)\} \) for all \( 1 \leq i \leq M \).

(d) Now let \( l_i = \lceil \log_2 \frac{S}{\max\{P_1(a_i), \ldots, P_k(a_i)\}} \rceil \), and let us compute the Kraft sum:

\[ \sum_{i=1}^{M} 2^{-l_i} \leq \sum_{i=1}^{M} 2^{-\log_2 \frac{S}{\max\{P_1(a_i), \ldots, P_k(a_i)\}}} = \sum_{i=1}^{M} \frac{\max\{P_1(a_i), \ldots, P_k(a_i)\}}{S} = 1. \]
Since the Kraft sum is at most 1, there exists a prefix-free code where the length of the codeword associated to \( a_i \) is \( l_i \). Since the code is prefix free, it must be the case that \( \bar{l} \geq H(U) \). In order to prove the upper bounds, let \( P^* \) be the true distribution (which is either \( P_1 \) or \ldots or \( P_k \)). It is easy to see that \( P^*(a_i) \leq \max\{P_1(a_i), \ldots, P_k(a_i)\} \) for all \( 1 \leq i \leq M \). We have:

\[
\bar{l} = \sum_{i=1}^{M} P^*(a_i) l_i = \sum_{i=1}^{M} P^*(a_i) \left[ \log_2 \frac{S}{\max\{P_1(a_i), \ldots, P_k(a_i)\}} \right] \\
< \sum_{i=1}^{M} P^*(a_i) \left( 1 + \log_2 \frac{S}{\max\{P_1(a_i), \ldots, P_k(a_i)\}} \right) \\
= \sum_{i=1}^{M} P^*(a_i) \left( 1 + \log_2 S + \log_2 \frac{1}{\max\{P_1(a_i), \ldots, P_k(a_i)\}} \right) \\
= 1 + \log_2 S + \sum_{i=1}^{M} P^*(a_i) \log_2 \frac{1}{P^*(a_i)} = H(U) + \log_2 S + 1,
\]

where the inequality (*) uses the fact that \( P^*(a_i) \leq \max\{P_1(a_i), \ldots, P_k(a_i)\} \) for all \( 1 \leq i \leq M \). Now notice that \( \max\{P_1(a_i), \ldots, P_k(a_i)\} \leq \sum_{j=1}^{k} P_j(a_i) \) for all \( 1 \leq i \leq M \). Therefore, we have

\[
S = \sum_{i=1}^{M} \max\{P_1(a_i), \ldots, P_k(a_i)\} \leq \sum_{i=1}^{M} \sum_{j=1}^{k} P_j(a_i) = \sum_{j=1}^{k} \sum_{i=1}^{M} P_j(a_i) = k \sum_{j=1}^{k} 1 = k.
\]

We conclude that \( H(U) \leq \bar{l} \leq H(U) + \log S + 1 \leq H(U) + \log k + 1 \).

**Problem 5.**

(a) We prove the identity by induction on \( n \geq 1 \). For \( n = 1 \), the identity is trivial. Let \( n > 1 \) and suppose that the identity is true up to \( n - 1 \). We have:

\[
I(Y_1^n; X_n) = I(Y_1^{n-2}, Y_{n-1}; X_n) \overset{(**)}{=} I(Y_1^{n-2}; X_n) + I(X_n; Y_{n-1}|Y_1^{n-2}) \\
\overset{(**)}{=} \sum_{i=1}^{n-2} I(X_n; Y_i|Y_1^{i-1}) + I(X_n; Y_{n-1}|Y_1^{n-2}) = \sum_{i=1}^{n-1} I(X_n; Y_i|Y_1^{i-1}).
\]

The identity (**) is by the chain rule for mutual information, and the identity (***) is by the induction hypothesis.

(b) For every \( 0 \leq i \leq n \), define \( a_i = I(X_i^n; Y_i^n) \), and for every \( 1 \leq i \leq n \), define \( b_i = I(X_{i+1}^n; Y_{i+1}^n) \). It is easy to see that \( a_0 = a_n = 0 \). We have:

\[
\sum_{i=1}^{n} I(X_{i+1}^n; Y_i^n|Y_i^{i-1}) \overset{(***)}{=} \sum_{i=1}^{n} \left( I(X_{i+1}^n; Y_i^n) - I(X_{i+1}^n; Y_1^{i-1}) \right) = \left( \sum_{i=1}^{n} a_i \right) - \left( \sum_{i=1}^{n} b_i \right) \\
\overset{(***)}{=} \left( \sum_{i=0}^{n-1} a_i \right) - \left( \sum_{i=1}^{n} b_i \right) = \left( \sum_{i=1}^{n} a_{i-1} \right) - \left( \sum_{i=1}^{n} b_i \right) = \sum_{i=1}^{n} (a_{i-1} - b_i) \\
= \sum_{i=1}^{n} \left( I(X_i^n; Y_i^{i-1}) - I(X_{i+1}^n; Y_1^{i-1}) \right) \overset{(***)}{=} \sum_{i=1}^{n} I(Y_{i-1}^n; X_i^n|X_{i+1}^n).
\]

3
The identities (∗) and (∗∗) are by the chain rule for mutual information. The identity (∗∗) follows from the fact that \(a_0 = a_n = 0\), which implies that \(\sum_{i=1}^{n} a_i = \sum_{i=0}^{n-1} a_i\).

**Problem 6.**

(a) The number of binary sequences of length \(n\) that have a given substring of length \(m \leq n\) is \(2^{n-m}\): for each of the \(n-m\) positions outside the substring we have 2 choices. Consequently the number of words in \(A_j\) that have \(C(i)\) as an initial substring (prefix) is \(2^{l_j-i_j}\) and similarly for the number of words that have \(C(i)\) as a suffix.

(b) The words removed in (∗) and (∗∗) are precisely those discussed in (a). As some of those may have been removed in a prior step, and since the words in (∗) and (∗∗) may overlap, the number of words removed is at most \(2 \cdot 2^{l_j-i_j} = 2^{l_j-i_j+1}\).

(c) The number of words removed from \(A_i\) at the time we test \(A_i \neq \emptyset\) is at most

\[
\sum_{m=1}^{i-1} 2^{l_m-l_{m+1}} = 2^{l_i} 2 \sum_{m=1}^{i-1} 2^{-l_m} < 2^{l_i}
\]

since \(\sum_{m=1}^{i-1} 2^{-l_m} < \sum_{m=1}^{k} 2^{-l_m} \leq \frac{1}{2}\). As the initial size of \(A_i\) was \(2^{l_i}\) we see that \(A_i\) is not empty at the time of the test, and thus the algorithm will not fail.

(d) We know from (c) that algorithm will not fail. Since \(C(i)\) is chosen from \(A_i\) it is of length \(l_i\). Also, steps (∗) and (∗∗) ensure that \(C(i)\) is neither a prefix nor a suffix of \(C(j)\) for \(j > i\). On the other hand since \(l_1 \leq \cdots \leq l_k\), \(C(i)\) can not be a prefix or suffix of \(C(j)\) for \(j < i\) either. So the returned code is fix-free.

(e) Choosing \(l(u) = \lceil \log \frac{1}{p(u)} \rceil + 1\) yields

\[
\log \frac{1}{p(u)} + 1 \leq l_i \leq \log \frac{1}{p(u)} + 2.
\]

The right hand side inequality ensures \(E[l(U)] \leq H(U) + 2\), whereas the left hand side inequality ensures \(2^{-l(u)} \leq p(u)/2\) and thus \(\sum_u 2^{-l(u)} \leq 1/2\) and consequently the existence of a fix-free code \(C\) with these lengths.

**Problem 7.**

(a) We can write the following chain of inequalities:

\[
Q^n(x) = \prod_{i=1}^{n} Q(x_i) = \prod_{a \in \mathcal{X}} Q(a)^{N(a|x)} = \prod_{a \in \mathcal{X}} Q(a)^{nP_x(a)} = \prod_{a \in \mathcal{X}} 2^{nP_x(a) \log Q(a)} \tag{1}
\]

\[
= \prod_{a \in \mathcal{X}} 2^{nP_x(a) \log Q(a)-P_x(a) \log P_x(a)+P_x(a) \log P_x(a)} \tag{2}
\]

\[
= 2^n \sum_{a \in \mathcal{X}} (-P_x(a) \log \frac{P_x(a)}{Q(a)}+P_x(a) \log P_x(a)) = 2^n(\mathcal{D}(P_x||Q)+H(P_x)),
\]

where 1 follows because the sequence is i.i.d., grouping symbols gives 2, and 3 is the definition of type.
(b) Upper bound: We know that
\[ \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} = 1. \]

Consider one term and set \( p = k/n \). Then,
\[ 1 \geq \binom{n}{k} \left( \frac{k}{n} \right)^k \left( 1 - \frac{k}{n} \right)^{n-k} = \binom{n}{k} 2^{n \left( \frac{k}{n} \log \frac{k}{n} + \frac{n-k}{n} \log \frac{n-k}{n} \right)} = \binom{n}{k} 2^{-nh_2\left( \frac{k}{n} \right)} \]

Lower bound: Define \( S_j = \binom{n}{j} p^j (1-p)^{n-j} \). We can compute
\[ \frac{S_{j+1}}{S_j} = \frac{n-j}{j+1} \frac{p}{1-p}. \]

One can see that this ratio is a decreasing function in \( j \). It equals 1, if \( j = np + p - 1 \), so \( \frac{S_{j+1}}{S_j} < 1 \) for \( j = \lfloor np + p \rfloor \) and \( \frac{S_{j+1}}{S_j} \geq 1 \) for any smaller \( j \). Hence, \( S_j \) takes its maximum value at \( j = \lfloor np + p \rfloor \), which equals \( k \) in our case. From this we have that
\[ 1 = \sum_{j=0}^{n} \binom{n}{j} p^j (1-p)^{n-j} \leq (n+1) \max_j \binom{n}{j} p^j (1-p)^j \]
\[ \leq (n+1) \binom{n}{k} \left( \frac{k}{n} \right)^k \left( 1 - \frac{k}{n} \right)^{n-k} = (n+1) \binom{n}{k} 2^{-nh_2\left( \frac{k}{n} \right)}. \] (3)

The last equality comes from the derivation we had when proving the upper bound.

(c) Since for every \( x \in T(P) \), \( Q^n(x) = 2^{-nH(P)+D(P\|Q)} \) (by part (a)) and \( \frac{1}{n+1} 2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)} \) (by part (b)), we have
\[ \frac{1}{n+1} 2^{-nD(P\|Q)} \leq Q^n(T(P)) \leq 2^{-nD(P\|Q)} \]