Problem 1. Suppose \( L : \mathbb{R}^K \to \mathbb{R}^N \) is a linear function and \( g : \mathbb{R}^N \to \mathbb{R} \) is a concave function. Show that \( f : \mathbb{R}^K \to \mathbb{R} \) defined as \( f(x) = g(L(x)) \) is concave.

Problem 2. From the notes on Lempel-Ziv algorithm, we know that the total length \( n \) of \( c \) distinct binary strings satisfies
\[
 n > c \log_2(c/8)
\]
where \( c \) is the maximum possible number of distinct parsings. The same technique, when applied to the non-binary strings yields
\[
 n > c \log_K(c/K^3)
\]
where \( K \) is the size of the alphabet the letters of the string belong to. This inequality lower bounds \( n \) in terms of \( c \). We will now show that \( n \) can also be upper bounded in terms of \( c \).

(a) Show that, if \( n \geq \frac{1}{2}m(m - 1) \), then \( c \geq m \).

(b) Find a sequence for which the bound in (a) is met with equality.

(c) Show now that \( n < \frac{1}{2}c(c + 1) \).

Problem 3. Let the alphabet be \( \mathcal{X} = \{a, b\} \). Consider the infinite sequence \( X_1^\infty = abababababababab\ldots \)

(a) What is the compressibility of \( \rho(X_1^\infty) \) using finite-state machines (FSM) as defined in class? Justify your answer.

(b) Design a specific FSM, call it \( M \), with at most 4 states and as low a \( \rho_M(X_1^\infty) \) as possible. What compressibility do you get?

(c) Using only the result in point (a) but no specific calculations, what is the compressibility of \( X_1^\infty \) under the Lempel–Ziv algorithm, i.e., what is \( \rho_{LZ}(X_1^\infty) \)?

(d) Re-derive your result from point (c) but this time by means of an explicit computation.

Problem 4. We are given a memoryless stationary binary symmetric channel \( \text{BSC}(\epsilon) \). Namely, if \( X_1, \ldots, X_n \in \{0, 1\} \) are the input of this channel and \( Y_1, \ldots, Y_n \in \{0, 1\} \) are the output, we have:
\[
P(Y_i|X_i, X_{i-1}, Y_{i-1}) = P(Y_i|X_i) = \begin{cases} 1 - \epsilon & \text{if } Y_i = X_i, \\ \epsilon & \text{otherwise.} \end{cases}
\]

Let \( W \) be a random variable that is uniform in \( \{0, 1\} \) and consider a communication system with feedback which transmits the value of \( W \) to the receiver as follows:

- At time \( t = 1 \), the transmitter sends \( X_1 = W \) through the channel.
At time \( t = i + 1 \leq n \), the transmitter gets the value of \( Y_i \) from the feedback and sends \( X_{i+1} = Y_i \) through the channel.

(a) Give the capacity \( C \) of the channel in terms of \( \epsilon \), and show that \( C = 0 \) when \( \epsilon = \frac{1}{2} \).

(b) Show that if \( \epsilon = \frac{1}{2} \), \( I(X^n; Y^n) = n - 1 \). This means that \( I(X^n; Y^n) \leq nC \) does not hold for this system.

(c) Show that although \( I(X^n; Y^n) > nC \) when \( \epsilon = \frac{1}{2} \), we still have \( I(W; Y^n) \leq nC \).

Note that since \( W \) is the useful information that is being transmitted, it is the value of \( I(W; Y^n) \) that we are interested in when we want to compute the amount of information that is shared with the receiver.

**Problem 5.** Consider the following variation on the Lempel-Ziv algorithm to encode an infinite sequence \( u_1 u_2 \ldots \) from an alphabet \( U \).

1. Set the dictionary \( D = U \). Denote the dictionary entries as \( d(0), \ldots, d(s-1) \), with \( s = |U| \) being the size of the dictionary. Set \( i = 0 \) (the number of input letters read so far).

2. Find the largest \( l \) such that \( w = u_{i+1} \ldots u_{i+l} \) is in \( D \).

3. With \( 0 \leq j < s \) denoting the index of \( w \) in \( D \), output the \( \lceil \log_2 s \rceil \) bit binary representation of \( j \).

4. Add the word \( w u_{i+l+1} \) to \( D \), i.e., set \( d(s) = w u_{i+l+1} \), and increment \( s \) by 1. Increment \( i \) by \( l \). Goto step 2.

For example, with \( U = \{a, b\} \), the input string \( a b b b a a b \ldots \) will lead to the execution steps

<table>
<thead>
<tr>
<th>( D ) at 2</th>
<th>( w )</th>
<th>output at 3</th>
<th>added-word at 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>a b</td>
<td>a</td>
<td>0</td>
<td>ab</td>
</tr>
<tr>
<td>a b ab</td>
<td>b</td>
<td>01</td>
<td>bb</td>
</tr>
<tr>
<td>a b ab bb</td>
<td>bb</td>
<td>11</td>
<td>bbb</td>
</tr>
<tr>
<td>a b ab bb bbb</td>
<td>b</td>
<td>001</td>
<td>ba</td>
</tr>
<tr>
<td>a b ab bb bbb ba</td>
<td>a</td>
<td>000</td>
<td>aa</td>
</tr>
<tr>
<td>a b ab bb bbb ba aa</td>
<td>aa</td>
<td>110</td>
<td>aab</td>
</tr>
</tbody>
</table>

(a) (5 pts) Can the decoder reconstruct the input sequence \( u_1 u_2 \ldots \) from the output of the algorithm? If so, how? (The crucial difficulty is that the description of \( w \) in step 3 does not determine the word added to the dictionary in step 4.)

(b) (5 pts) The algorithm parses the sequence \( u_1 u_2 \ldots \) into a sequence of words \( w_1 w_2 \ldots \), (the \( w \)’s found in step 2). Show that a word \( w \) can appear at most \( |U| \) times in the parsing.

(c) (5 pts) Suppose \( u^n = u_1 \ldots u_n \) is parsed into \( m(u^n) \) words \( w_1 \ldots w_m \) by the algorithm. Show that for any \( k \geq 1 \)

\[ n \geq k[m(u^n) - F(k)], \]

where \( F(k) = |U| \sum_{i=1}^{k-1} |U|^i \).

(d) (5 pts) Show that \( \lim_{n \to \infty} m(u^n)/n = 0 \).
(e) (5 pts) Show that after reading $u^n$ the algorithm outputs fewer than $m(u^n)[\log_2|\mathcal{U}| + m(u^n)]$ bits.

Let $L(m, k)$ denote the minimum possible total length of a collection of $m$ binary strings where no string appears more than $k$ times.

(f) (5 pts) Show that if $u^n$ is fed to an information lossless finite state machine with $s$ states, then the machine outputs at least $L(m(u^n), s^2|\mathcal{U}|)$ bits.