PROBLEM 1. (12 points)
Suppose X, Y, Z are random variables. Consider the following statements about (X, Y, Z):

(i) X and Y are independent.

(ii) X and Z are independent.

(iii) $Z = f(X, Y)$ for some deterministic function f.

(iv) $X = g(Z, Y)$ for some deterministic function g.

(a) (2 pts) Show that if (i) and (iv) hold, then $I(X; Z|Y) = H(X)$.

Solution: By (iv), $H(X \mid Y, Z) = 0$, so

$$I(X; YZ) = H(X) - H(X \mid YZ) = H(X).$$

By the chain rule,

$$I(X; YZ) = I(X; Y) + I(X; Z \mid Y).$$

Since $X \perp Y$ we have $I(X; Y) = 0$, hence

$$H(X) = I(X; YZ) = I(X; Z \mid Y),$$

□

(b) (2 pts) Show that if (ii) and (iii) hold, then $I(Y; Z|X) = H(Z)$.

Solution: By (iii), $H(Z \mid X, Y) = 0$, so

$$I(Z; XY) = H(Z) - H(Z \mid XY) = H(Z).$$

Chain rule gives

$$I(Z; XY) = I(Z; X) + I(Z; Y \mid X).$$

By (ii) $I(Z; X) = 0$, so $H(Z) = I(Z; Y \mid X)$, as required. □

(c) (2 pts) Show that if (ii) and (iii) hold then $H(Y) \geq H(Z)$.

Solution: By (iii), $H(Z|XY) = 0$. So,

$$H(Z) = I(Z; XY)$$

Also, by chain rule,

$$I(Z; XY) = I(Z; X) + I(Z; Y|X)$$

Then, $H(Z) = I(Z; XY) = I(Z; Y|X)$ by (ii). Also, by chain rule

$$H(Y) = H(Y|X) + I(Y; X) \geq H(Y|X) = H(Y|Z, X) + I(Z; Y|X) \geq I(Z; Y|X)$$

Then, we obtain $H(Y) \geq H(Z)$. □

(d) (2 pts) Show that if (i) and (iv) hold, then $H(Z) \geq H(X)$.

Solution: By chain rule

$$H(Z) = H(Z|Y) + I(Y;Z) \geq H(Z|Y) = H(Z|Y,X) + I(Z;X|Y) \geq I(Z;X|Y)$$

Also, we have $I(Z;X|Y) = H(X)$ when (i) and (iv) hold by part (a). Then, we obtain $H(Z) \geq H(X)$.

□

(e) (4 pts) In shared-key cryptograpy, a secret key K is used to encrypt plaintext $T \in \mathcal{T}$, to cyphertext $C = enc(T, K)$. The plaintext T can be recovered from the knowledge of cyphertext C and the secret key K. The key K is chosen independently of T. In a *perfectly secret* system the distribution of K and the encyption function *enc* are chosen to ensure that the cyphertext C reveals no information about T for any distribution of T. Show that, in such a system, $H(K) \geq \log |\mathcal{T}|$.
Hint: Identify the triple (C, K, T) here with some permutation of the triple (X, Y, Z) above.

Solution: Model our plaintext T with the random variable X, secret key K with the random variable Y, and cyphertext C with the random variable Z. Then,

- That "$C = enc(T, K)$" is equivalent to (iii) with *enc* representing the deterministic function f.
- That "the plaintext T can be recovered from the knowledge of cyphertext C and the secret key K" is equivalent to (iv).
- That "the key K is chosen independently of T" is equivalent to (i).
- That "the cyphertext C reveals no information about T" is equivalent to (ii).

Then, choosing the distribution of T uniform and using part (c) and part (d),

$$H(K) = H(Y) \geq H(Z) \geq H(X) = H(T) = \log(T).$$

□

PROBLEM 2. (14 points)

Enumerate $\{0,1\}^*$ as $\{t_1, t_2, t_3, \dots\}$ where $t_1 = \varnothing$, $t_2 = 0$, $t_3 = 1$, $t_4 = 00$, $t_5 = 01$, $t_6 = 10$, $t_7 = 11$, $t_8 = 000, \dots$ Note that $\text{length}(t_j) = \lfloor \log j \rfloor$.
Let U take values in $\mathcal{U} = \{1, \dots, k\}$ with $\Pr(U = j) = p_j$.

(a) (2 pts) Suppose $p_1 \geq p_2 \geq \cdots \geq p_k$. Find an *injective* code $c : \mathcal{U} \to \{0,1\}^*$ that minimizes $\mathbb{E}[\text{length}(c(U))]$.

Solution: This problem explores how much better *injective* codes can be compared to *prefix-free* codes.

Since shorter codewords reduce the expected length, we should assign shorter codewords to more probable symbols. Because $p_1 \geq p_2 \geq \cdots \geq p_k$, the optimal injective code simply maps

$$c(j) = t_j, \qquad j = 1, \dots, k,$$

that is, the j-th most probable symbol is assigned the j-th binary string in the enumeration. Hence

$$\text{length}(c(j)) = \lfloor \log j \rfloor, \qquad \mathbb{E}[\text{length}(c(U))] = \sum_{j=1}^{k} p_j \lfloor \log j \rfloor.$$

□

(b) (2 pts) Under the assumption in (a), show that for all j we have $jp_j \leq 1$.

Solution: Consider the sum of the first j probabilities: $p_1 + \cdots + p_j \leq 1$. Since $p_1 \geq \cdots \geq p_j$, we have $jp_j \leq p_1 + \cdots + p_j \leq 1$.

(c) (2 pts) Show that for the code you constructed in (a),

$$\mathbb{E}[\text{length}(c(U))] \leq H(U).$$

Solution: Using part (b) and the fact that $\text{length}(t_j) = \lfloor \log j \rfloor$, we have

$$\mathbb{E}[\text{length}(c(U))] = \sum_{j=1}^{k} p_j \lfloor \log j \rfloor \leq \sum_{j=1}^{k} p_j \log \frac{1}{p_j} = H(U).$$

Thus, the injective code from part (a) has expected length at most the entropy.

(d) (2 pts) Show that for any $s > 1$ there exists a binary prefix-free code e for alphabet $\mathcal{L} = \{0, 1, \dots\}$ such that for every $j \in \mathcal{L}$,

$$\text{length}(e(j)) \leq \left\lceil \log\left(\tfrac{s}{s-1}\right) + s \log(1 + j) \right\rceil.$$

Hint: Use that for $s > 1$, $\sum_{j=0}^{\infty} (1+j)^{-s} < s/(s-1)$. Also, remember Kraft's Inequality from the class.

Solution:

We want a binary prefix-free code $e : \mathcal{L} \to \{0,1\}^*$ satisfying the Kraft inequality:

$$\sum_{j=0}^{\infty} 2^{-\text{length}(e(j))} \leq 1.$$

Consider choosing

$$\text{length}(e(j)) = \left\lceil \log \frac{1}{q_j} \right\rceil$$

for some distribution $(q_j)_{j\geq 0}$ with $\sum_j q_j \leq 1$. Let

$$q_j = \frac{s-1}{s}\frac{1}{(1+j)^s}, \quad s > 1.$$

Using the hint, we have

$$\sum_{j=0}^{\infty} q_j = \frac{s-1}{s}\sum_{j=0}^{\infty}\frac{1}{(1+j)^s} < \frac{s-1}{s}\cdot\frac{s}{s-1} = 1,$$

so the Kraft inequality is satisfied.

Thus a prefix-free code exists with

$$\text{length}(e(j)) = \left\lceil \log \frac{1}{q_j} \right\rceil = \left\lceil \log \frac{s}{s-1} + s\log(1+j) \right\rceil.$$

□

(e) (2 pts) Let c be the injective code from (a) and e the prefix-free code from (d). Define $c' : \mathcal{U} \to \{0,1\}^*$ by

$$c'(u) = e\big(\text{length}(c(u))\big)\, c(u).$$

Argue that c' is uniquely decodable (indeed prefix-free).

Solution:

To show that c' is uniquely decodable (indeed prefix-free), observe:

- The prefix $e(\text{length}(c(u)))$ is itself prefix-free, so reading from left to right we can always determine the length of $c(u)$ unambiguously.
- Once the length of $c(u)$ is known, the next $\text{length}(c(u))$ bits correspond exactly to $c(u)$, which is injective. Therefore, we can uniquely recover u.

Hence c' is uniquely decodable. In particular, it is prefix-free because no codeword $c'(u)$ can be a prefix of another: any codeword is split into the prefix-free $e(\text{length}(c(u)))$ followed by the injective $c(u)$, so no overlaps occur.

□

(f) (2 pts) With $H = H(U)$ show that for any $s > 1$,

$$H \leq 1 + \log\left(\tfrac{s}{s-1}\right) + s\log(1+H) + \mathbb{E}[\text{length}(c(U))].$$

Hint: Use that c' is uniquely decodable so $H \leq \mathbb{E}[\text{length}(c'(U))]$, then bound $\text{length}(c'(U))$ using (d) and take expectations.

Solution:

From part (e) the code c' is prefix-free, so its expected codeword length satisfies:

$$\mathbb{E}[\text{length}(c'(U))] \geq H.$$

Using the length bound from part (d) (applied to the index $\text{length}(c(U))$) we have for every u,

$$\text{length}(c'(u)) = \text{length}(e(\text{length}(c(u)))) + \text{length}(c(u))$$

$$\leq 1 + \log\frac{s}{s-1} \;+\; s\log\big(1+\text{length}(c(u))\big) \;+\; \text{length}(c(u)).$$

Taking expectations and using linearity of expectation,

$$\mathbb{E}[\text{length}(c'(U))] \leq 1 + \log\frac{s}{s-1} \;+\; s\,\mathbb{E}\big[\log(1+\text{length}(c(U)))\big] \;+\; \mathbb{E}[\text{length}(c(U))].$$

Since $x \mapsto \log(1+x)$ is concave, Jensen's inequality gives

$$\mathbb{E}\big[\log(1+\text{length}(c(U)))\big] \leq \log\big(1+\mathbb{E}[\text{length}(c(U))]\big).$$

By part (c) we have $\mathbb{E}[\text{length}(c(U))] \leq H$, hence

$$\mathbb{E}[\text{length}(c'(U))] \leq 1 + \log\frac{s}{s-1} + s\log(1+H) + \mathbb{E}[\text{length}(c(U))].$$

Combining this with $\mathbb{E}[\text{length}(c'(U))] \geq H$ yields for any $s > 1$

$$H \leq 1 + \log\big(\tfrac{s}{s-1}\big) + s\log(1+H) + \mathbb{E}[\text{length}(c(U))].$$

□

(g) (2 pts) Conclude that for any injective code c,

$$\mathbb{E}[\text{length}(c(U))] \geq H - \log(1+H) - \log(1+\log(1+H)) - 2.$$

Hint: Choose $s = 1 + 1/\log(1+H)$ in (f).

Solution: Choose $s = 1 + 1/\log(1+H)$ in (f). Then:

$$H \leq 1 + \log(1+\log(1+H)) + \log(1+H) + 1 + \mathbb{E}[\text{length}(c(U))].$$

$$\implies \mathbb{E}[\text{length}(c(U))] \geq H - \log(1+H) - \log(1+\log(1+H)) - 2.$$

□

PROBLEM 3. (16 points)
A (non-deterministic) finite state machine (FSM) source is specified by

- a finite state set $\mathcal{S}$ and an initial state $s_0 \in \mathcal{S}$;
- for each state $s \in \mathcal{S}$ a probability distribution $p(\cdot \mid s)$ on alphabet $\mathcal{U}$;
- a next-state function $g : \mathcal{S} \times \mathcal{U} \to \mathcal{S}$.

Such a machine generates $U_1, U_2, \ldots$ by setting $S_0 = s_0$, and for each $t = 0, 1, \ldots$, sampling $U_{t+1} \sim p(\cdot \mid S_t)$ and setting $S_{t+1} = g(S_t, U_{t+1})$.
A *finite-state predictor* is an FSM which, after seeing u^t, outputs a distribution q_{t+1} on $\mathcal{U}$ (its belief about (U_{t+1})).

(a) (4 pts) Show that if $U_1, U_2, \ldots$ is generated by an FSM source as above, then there is an FSM predictor which, for every t, outputs the exact conditional distribution

$$q_{t+1}(u) = \Pr(U_{t+1} = u \mid U^t = u^t).$$

(Describe its states and q_{t+1}.)

Solution: Let the FSM source be given by state set $\mathcal{S}$, start state s_0, emission distributions $p(\cdot \mid s)$ for $s \in \mathcal{S}$, and next-state function $g : \mathcal{S} \times \mathcal{U} \to \mathcal{S}$. Define the predictor FSM as follows:

- The predictor's state set is $\mathcal{S}$ (the same as the source).
- The predictor's start state is s_0.
- Upon seeing symbol u_t (reading the sequence u^t sequentially), the predictor updates its state deterministically by

$$s_t \leftarrow g(s_{t-1}, u_t),$$

  i.e. exactly the same update rule as the source.
- When in state s_t (after seeing u^t), the predictor outputs the distribution

$$q_{t+1}(u) := p(u \mid s_t), \qquad u \in \mathcal{U}.$$

If the source started in $S_0 = s_0$ then after observing the realized symbols u^t the source state is exactly

$$S_t = g(\cdots g(g(s_0, u_1), u_2) \cdots, u_t),$$

so S_t is a deterministic function of u^t. Therefore

$$\Pr(U_{t+1} = u \mid U^t = u^t) = \Pr(U_{t+1} = u \mid S_t = s_t) = p(u \mid s_t),$$

where s_t is the state computed above. Hence the predictor's output $q_{t+1}(u) = p(u \mid s_t)$ equals the desired conditional distribution $\Pr(U_{t+1} = u \mid U^t = u^t)$ for every t. □

(b) (4 pts) Extend (a) to predicting the next k letters: show that for each $k \geq 1$ there is a FSM predictor which outputs the exact distribution of $(U_{t+1}, \ldots, U_{t+k})$ given U^t. (Part (a) above is the case $k = 1$.)

Solution: We now extend the construction from part (a) to predict block of k future symbols.

– The predictor's state set remains $\mathcal{S}$, with start state s_0 and update rule

$$s_t \leftarrow g(s_{t-1}, u_t),$$

identical to the source FSM.

– When in state s_t, the predictor outputs the distribution

$$q_{t+1}(u_{t+1}, \ldots, u_{t+k}) := \Pr(U_{t+1}^{t+k} = u_{t+1}^{t+k} \mid U^t = u^t).$$

Given $S_t = s_t$, the FSM source generates U_{t+1}^{t+k} recursively:

$$\Pr(U_{t+1}^{t+k} = u_{t+1}^{t+k} \mid S_t = s_t) = \prod_{i=1}^{k} p(u_{t+i} \mid s_{t+i-1}),$$

where

$$s_{t+i} = g(s_{t+i-1}, u_{t+i}), \qquad s_t \text{ known from } u^t.$$

Therefore the FSM predictor, knowing s_t, can compute this exact product and output the full joint distribution on $\mathcal{U}^k$.

Since S_t is a deterministic function of U^t, conditioning on $U^t = u^t$ is equivalent to conditioning on $S_t = s_t$. Hence

$$\Pr(U_{t+1}^{t+k} = u_{t+1}^{t+k} \mid U^t = u^t) = \Pr(U_{t+1}^{t+k} = u_{t+1}^{t+k} \mid S_t = s_t) = q_{t+1}(u_{t+1}^{t+k}),$$

so the constructed FSM predictor indeed outputs the exact conditional block distribution for every t and k. □

(c) (4 pts) Suppose $u_1, u_2, \ldots$ is the realization of $U_1, U_2, \ldots$ generated by a FSM source. For each $k \geq 1$, construct an information-lossless FSM encoder which, after seeing u^n, outputs at most

$$-\log \Pr(U^n = u^n) + \lfloor n/k \rfloor$$

bits.

Hint: 1) For any probability distribution q, there is a prefix-free code (called the 'Shannon code') with $\text{length}(\text{code}(u)) \leq -\log(q(u)) + 1$.

2) Use the predictor from (b) to Shannon-code successive blocks of k source letters.

Solution:

Use the predictor from part (b), and partition the sequence u^n into consecutive non-overlapping k-blocks

$$(u_1^k), (u_{k+1}^{2k}), \ldots, (u_{(m-1)k+1}^{mk}),$$

where $m = \lfloor n/k \rfloor$.

For each block, the encoder:

– applies a Shannon Code to the predicted distribution q_{t+1}, assigning codeword length

$$\ell(u_{t+1}^{t+k}) = \left\lceil -\log q_{t+1}(u_{t+1}^{t+k}) \right\rceil;$$

– updates its state according to the source transition rule after reading the k symbols.

The FSM encoder is information-lossless because the Shannon code for each distribution is prefix-free. The total number of bits produced is

$$L(u^n) = \sum_{t\in\{0,k,\ldots,(m-1)k\}} \left\lceil -\log q_{t+1}(u_{t+1}^{t+k}) \right\rceil \le \sum_{t\in\{0,k,\ldots,(m-1)k\}} \left(-\log q_{t+1}(u_{t+1}^{t+k}) + 1\right).$$

By the chain rule for probabilities,

$$\sum_{t\in\{0,k,\ldots,(m-1)k\}} -\log q_{t+1}(u_{t+1}^{t+k}) = -\log \Pr(U^{mk} = u^{mk}),$$

and there are m terms in the sum, so

$$L(u^n) \le -\log \Pr(U^{mk} = u^{mk}) + m \le -\log \Pr(U^n = n^n) + m.$$

□

(d) (4 pts) Using (c) and the properties of Lempel–Ziv discussed in class, show that for any $\varepsilon > 0$ and large enough n, the (expected) number of bits produced by LZ that allow recovery of U^n is at most

$$H(U^n) + n\varepsilon.$$

Hint: For any ILFSM encoder M and any $\delta > 0$, for large n, LZ will produce fewer than δ bits/symbol in excess of those produced by M. Choose k large, and consider the machine M found in (c).

Solution: Fix $\varepsilon > 0$. Choose an integer block length k so large that

$$\frac{1}{k} \le \frac{\varepsilon}{2}.$$

Use part (c) to build the ILFSM encoder tailored to this k. (The ILFSM from (c) has a finite number of states that depends only on the source FSM and on k; in particular the encoder is an ILFSM with a fixed finite number of states once k is fixed.)

By construction in part (c), for every sequence u^n the ILFSM from (c) outputs at most

$$-\log \Pr(U^n = u^n) + \frac{n}{k}$$

bits (note that $\lfloor n/k \rfloor \le n/k$), hence taking expectation under the true source distribution gives

$$\mathbb{E}[\text{length}_{\text{ILFSM}}(U^n)] \le H(U^n) + \frac{n}{k} \le H(U^n) + \frac{n\varepsilon}{2}.$$

A standard property of Lempel–Ziv (as discussed in class) is that for any fixed finite-state encoder (i.e. for any ILFSM with a fixed number of states) there exists N such that for all $n \ge N$ the expected LZ length is no larger than the expected length of that finite-state encoder plus an additive $n\varepsilon/2$ term. (Intuitively: for sufficiently long sequences LZ outperforms any fixed-state scheme by an arbitrarily small per-letter gap.)

Combining these two facts, for the chosen k and for all $n \ge N$ we obtain

$$\mathbb{E}[\text{length}_{\text{LZ}}(U^n)] \le \mathbb{E}[\text{length}_{\text{ILFSM}}(U^n)] + \frac{n\varepsilon}{2} \le \left(H(U^n) + \frac{n\varepsilon}{2}\right) + \frac{n\varepsilon}{2} = H(U^n) + n\varepsilon.$$

This proves that for any $\varepsilon > 0$ and all sufficiently large n the expected number of bits produced by LZ that allow recovery of U^n is at most $H(U^n) + n\varepsilon$. □