1. Recall that a prefix-free machine W is called *universal* if for every prefix-free machine S there effectively exists a constant c such that for each input string x:

$$K_W(x) \leq K_S(x) + c.$$

(a) Prove that for every universal prefix-free machines U and V there is a constant d such that for all strings x:

$$\mid K_U(x) - K_V(x) \mid \leq d.$$

Solution: We have for c large enough $K_U(x) \leq K_V(x) + c$ and $K_V(x) \leq K_U(x) + c$. Thus $|K_U(x) - K_V(x)| \leq c$.

(b) Prove that every universal prefix-free machine machine W is onto, i.e., for every y there exists x such that $W(x) = y$.

Solution: Let S be some onto prefix free machine (such as given in class). For each x we have $K_W(x) \leq K_S(x) + c$. In particular, there is p such that $W(p) = x$.

(c) Is every universal prefix-free machine W *infinitely* onto? I.e., do there exist for every y infinitely many x such that $W(x) = y$?

Solution: NO. Let U be a universal machine. Let W be the prefix free machine defined as follows. When at stage s we see a new computation $U_s(p) = y$, declare $W(p) = y$ unless there already is a shorter program q for which we have declared $W(q) = y$.
Clearly, W is universal; in fact $K_W(y) = K_U(y)$.
However, for each string y the preimage under W is finite.

(d) Let W be a universal prefix-free machine such that $W(x) = y$. Is the restriction of W to $\text{dom}(W) \setminus \{x\}$ is still universal?

Solution: Not always. We modify W obtained above so that for some y, say ϵ, the preimage has only one element x. Taking x out of the domain means that the preimage of y becomes empty.

2. Let α be a real in $(0, 1]$. Show that the following conditions are equivalent (such reals are called c.e.): [10 marks]

(a) There is a computable, nondecreasing sequence of rationals which converges to α.

(b) The set of rationals less than or equal to α is c.e.

(c) There is an infinite prefix-free c.e. set $A \subseteq B^*$ with $\alpha = \Omega_A$.

Solution:
(a)$\rightarrow$ (c): Let the sequence be $(q_n)_{n\in\mathbb{N}}$. We may assume they are all dyadic rationals (i.e. of the form $0.x$ for some binary string x), because the dyadic rationals are dense in $\mathbb{Q}$. We may also suppose that $0 \leq q_n < q_{n+1}$ for each n.
Define the prefix-free c.e. set A as follows: say $q_n = 0.x$ and $q_{n+1} = 0.y$ where x, y have the same length k (which we can assume after putting some 0s at the end of a string if necessary). Put into A all strings z of length k such that $x \leq_L z <_L y$ ($\leq_L$ is the lex order).

Question: how would you improve this to make A even computable?

(c)$\rightarrow$ (b): Let A_s be the finite set of strings listed by stage s. List a rational q if at some s we have $q \leq \Omega_{A_s}$. Then this is a computable enumeration of the left cut $\{q \in \mathbb{Q}\colon\ q \leq \alpha\}$. (If α is rational, also list α itself.)

(b)$\rightarrow$ (a): Suppose we have a computable enumeration of the left cut $\{q \in \mathbb{Q}\colon\ q \leq \alpha\}$. Let q_s be the maximum of the rationals listed by stage s. Then (q_s) is a sequence as required.

3. A real is called computable if its binary representation is computable. Give an example of a c.e. real that is not random but also is not computable. [5 marks]

   Solution: There are many possibilities. For instance, take the binary expansion of Ω and insert 0 at each other bit position. Call this α. Clearly α is computationally equivalent to Ω, hence not computable. Also α is c.e. as a real. Finally, we have

$$K(\alpha_1 \dots \alpha_{2n+1}) \leq K(\Omega_1 \dots \Omega_n) + c,$$

   so α is not algorithically random.

4. Recall that $C(x)$ is the plain Kolmogorov complexity of string x, where any machine is allowed (warning: this is denoted $K(x)$ in Sipser).

   (a) Explain why there are constants $d, d' \in \mathbb{N}$ such that $C(x) \leq |x| + d$ and $C(x) \leq K(x) + d'$ for each x.

   Solution: (a) The "copying machine" M introduced in class computes the identity function. Thus, $C_M(x) = |x|$ for each x. Since M is simulated by the standard universal machine used to define C, we have $C(x) \leq C_M(x) + d = |x| + d$ for some d.

   (b) The standard universal prefix free machine is simulated by the (plain) standard universal machine. Thus $C(x) \leq K(x) + d'$ for some d'.

   (b) Show that for each $b \in \mathbb{N}$, for each n, there are at least $2^n - 2^{n-b} + 1$ strings x of length n such that $C(x) \geq n - b$.

   Solution: There are at most $1 + 2 + 4 + \ldots + 2^{n-b-1} = 2^{n-b} - 1$ programs of length $< n-b$. Thus at most $2^{n-b}-1$ strings x satisfy $C(x) < n-b$. Hence at least $2^n-(2^{n-b}-1)$ strings of length n satisfy $C(x) \geq n - b$

   (c) Show that there is a constant d such that $C(xy) \leq K(x) + C(y) + d$ for each strings x, y.

   Solution: Let U be universal prefix free, and V universal plain. Machine M on input p searches for a splitting $p = qr$ such that $U(q) = x$ converges and $V(r) = y$ converges. Then it prints xy.

   If we let q be a shortest prefix free program for x, and r a shortest plain program for y, then $M(p) = xy$ where $p = qr$. This shows

$$C(xy) \leq |p| + d = |q| + |r| + d = K(x) + C(y) + d.$$