Solutions to Problem Set 6

1. (Sipser problem 6.20.) Show how to compute the Kolmogorov complexity $K_U(x)$ of a string x with an oracle for A_{TM}.

   The definition of an oracle is given in Sipser definition 6.20 on page 233. An oracle is essentially a subroutine. You could interprete this problem as asking for an algorithm that on input x, computes the descriptive complexity of x, that is, $K_U(x)$, using a subroutine for A_{TM}. On input $\langle M, w\rangle$ the subroutine will return 1 if M accepts w, and 0 otherwise. Whenever you invoke the subroutine on some input $\langle M, w\rangle$, use the terminology "query the A_{TM} oracle on input $\langle M, w\rangle$".

   For instance, the machine S in the proof that $HALT_{\mathsf{TM}}$ is undecidable in Sipser theorem 5.1 (page 188-189) is an example of a algorithm for $HALT_{\mathsf{TM}}$ using an oracle for A_{TM}. In that example, S only uses the A_{TM} subroutine once. In general (and for this problem), you are allowed to invoke the subroutine any number of times, and the oracle queries may be adaptive (that is, the next query may depend on the answers to the previous ones).
   [20 points]

   SOLUTION:Here's an algorithm for computing $K(x)$. On input x,

   1. Go through all binary strings s in lexicographic order, and for each such s, parse s as $\langle M, w\rangle$ for some TM M and input w. If s fails to parse, move to the next such s.
   2. Modify the machine M so that all transitions to the reject state go to the accept state. Call the modified machine M'. Now, M halts on w if and only if M' accepts w.
   3. Next, query A_{TM} on input $\langle M', w\rangle$. If A_{TM} accepts $\langle M', w\rangle$, simulate M on w (this will halt), and check whether M on input w halts with x on its tape. Output $|s|$.

   Since we are going through the strings in lexicographic order, we will output the length of the shortest description (and the lexicographically first one if there is a tie).

2. Prove that if R is the set of Kolmogorov random strings $\{x \in \Sigma^* | K(x) \geq |x|\}$ and A is a decidable subset of R, then A is finite.
   [15 points]

   SOLUTION: Supposing A is not finite, for all $k \geq 0$, $\exists x \in A$ such that $|x| \geq k$. Also, since $A \subset R$, we know that $K(x) = |x|$. Hence, on input x (taking $\lceil \log k \rceil$ bits), we enumerate all x strings of length at least k and check if $x \in A$. On finding some $x^* \in A$, we output x^*. Since $K(x^*) = |x^*| \geq k$, and our algorithm produces it on an input of length $\lceil \log k \rceil$, this is a contradiction for large enough k.

3. If $f : \Sigma^* \to \mathbb{Z}_+$ is a computable function such that $f(x) \leq K(x)\ \forall x \in \Sigma^*$, then show that f must be bounded.
   [15 points]

   SOLUTION: Suppose f is not bounded. Then for every $k \geq 0$, there is a string x such that $f(x) \geq k$. Then the following algorithm outputs a string of Kolmogorov complexity at least k given k as an input.

1. Enumerate all strings x of length at least k in lexicographic order and compute $f(x)$ for each till finding an x^* such that $f(x^*) \geq k$.
2. Output x^*.

The algorithm always terminates because a suitable x^* always exists for any $k \geq 0$. Since the length of the input is $\lceil \log k \rceil$ bits and the Kolmogorov complexity of the output is at least k, this is a contradiction for large enough k.

4. (Sipser problem 7.17.) Prove that if $\mathbf{P} = \mathbf{NP}$ then every language in $\mathbf{P}$, except $\emptyset$ and Σ^*, is $\mathbf{NP}$-complete. Why can't $\emptyset$ and Σ^* be $\mathbf{NP}$-complete?
[10 points]

Solution: Let A be any language in NP and let B be another language not equal to $\emptyset$ or Σ^*. Then there exist strings $x \in B$ and $y \notin B$. To reduce an instance w of A to that of B, we just check in polynomial time if $w \in A$. If yes, we output x and y when $w \notin A$.

The languages $\emptyset$ and Σ^* cannot be NP-complete, because to reduce a language A to a language B, we need to map instances in A to instances in B and those outside A to outside B. However, for $B = \emptyset$, there are no instances in B (and none outside B for $B = \Sigma^*$) which means there cannot be such a reduction from any language $A \neq \emptyset, \Sigma^*$.