Solutions of homework 7

   Solution Outline: (25 points)
   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).

   Solution Outline: (25 points)
   Suppose on the contrary that the set $C$ of incompressible strings contains an infinite Turing recognizable subset. Let $E$ be an enumerator for that subset. Now, let $M$ denote the Turing machine that on input $n$ a positive integer in binary representation, outputs the first string printed by $E$ that has length at least $n$. Then, for every positive integer $n$, $\langle M, n \rangle$ is a description of an incompressible string $s_n$ of length at least $n$, and the description has length $O(1) + \log n$, so we have $n \leq K(s_n) \leq O(1) + \log n$, a contradiction for sufficiently large $n$.

   Solution Outline: (25 points)
   Fix $A \in P$. To see that $A^* \in P$, we simply use a dynamic programming approach. On input $y = y_1 \ldots y_n$, we build a table indicating for each $i = 1, 2, \ldots, n$, whether $y_1 \ldots y_i \in A^*$. For $i = 1$, it is straight-forward. For $i > 1$, $y_1 \ldots y_i \in A^*$ if $y_1 \ldots y_i \in A^*$, or for some $j, 1 \leq j \leq i - 1, y_1 \ldots y_j \in A^*$ and $y_{j+1} \ldots y_i \in A$; we can check this by enumerating over all possible $j$’s. When we reach $i = n$, we are done.

   Solution Outline: (25 points)
   Fix $L \in NP$. To see that $L^* \in NP$, simply non-deterministically select a partition of the input $x$ into substrings (that requires just $|x| - 1$ guesses), and non-deterministically guess a witness (for the relation defined by $L$, which has sized bounded by $poly(|x|)$) for each of the substrings. Alternatively, consider the verifier $V^*$ for $L^*$ that on input $\langle x, \langle x_1, \ldots, x_t, w \rangle \rangle$, checks that $x = x_1x_2 \ldots x_t$ and that the verifier for $L$ accepts $(x, w)$. If both tests accept, $V^*$ accepts; otherwise, reject.