CS 4510: Automata and Complexity

Spring 2015

Home work 6 // Due: Friday, April 17, 2015

1. (15 points) Show that any non-trivial property $\mathcal{P}$ of recognizable sets that includes the empty-set $\emptyset$ is not recognizable.

   (Hint: Give an alternative proof of Rice's theorem that uses a reduction from $\overline{A_{\text{TM}}}$ to any non-trivial property $\mathcal{P}$ that includes TMs whose language is the $\emptyset$.)

   Solution: Let $T_\emptyset$ be a TM such that $L(T_\emptyset) = \emptyset$. Suppose that $T_\emptyset$ is in $\mathcal{P}$. Then, all TMs M such that $L(M) = \emptyset$ is in $\mathcal{P}$. Also, there is some TM T *not* in $\mathcal{P}$ (since $\mathcal{P}$ is non-trivial) and $L(T) \neq \emptyset$.

   The reduction machine on input M, w constructs a DTM M':

   - On input x:
     - Simulate M on w.
     - M halts and rejects w: reject x.
     - M halts and accepts w: run T on x. Accept x if and only if T accepts x.

   Then, $L(M') = L(T)$ if M accepts w and $\emptyset$ if M does not accept w. That is, $L(M') = \emptyset$ if and only if M does not accept w. That is, M' is in $\mathcal{P}$ if and only if $M, w \in \overline{A_{\text{TM}}}$.

2. (15 points) The *Constrained* PCP problem is the PCP problem in which each tile can be used in a match at most once.

   That is, the Constrained PCP problem is the following:

   $$\{\left[\frac{t_1}{b_1}\right], \left[\frac{t_2}{b_2}\right], \ldots, \left[\frac{t_k}{b_k}\right] \mid \exists \text{ DISTINCT indices } (i_1, i_2, \ldots, i_m) \text{ such that } t_{i_1} t_{i_2} \ldots t_{i_m} = b_{i_1} b_{i_2} \ldots b_{i_m}\}.$$

   where $t_1, t_2, \ldots, t_k, b_1, b_2, \ldots, b_k$ are strings over some alphabet Σ.

   Is the Constrained PCP problem decidable? Prove your answer.

   Solution: Let m denote the number of tiles used to get a match. Clearly $1 \leq m \leq k$, since you cannot use a tile more than once.

   For $1 \leq m \leq k$, for all m subsets S of $\{1, 2, \ldots, k\}$, for all permutations of the elements of S, check to see if the permutations correspond to a match. There are $C(n, k)$ (here, $C(n, k)$ is the binomial coefficient 'n choose k') subsets of size k, and for each such subset there are $k!$ permutations, hence the total number of possibilities is $\sum_{i=0}^{n} k! \cdot C(n, k)$. Clearly, all these possibilities can be enumerated and checked by a Turing machine that halts.

3. (10 points) Problem 7.41, page 327 of the text.

   In the proof of Cook-Levin theorem, a window is a 2×3 rectangle of cells. Show why th eproof would hav efailed if we had used a 2×2

   Solution: To see why 2×2 rectangles do not suffice, consider the following sequence of consecutive Turing machine configurations:

   $$\#aqbbd\#$$

   $$\#abbqd\#$$

   These two configuations cannot be valid consecutive configurations as the state is displaced by 2 tape cells rather than just 1. However if we just examine 2×2 block, these two will pass as valid consecutive configuations. This is because the blocks

   $$qb \quad bb \quad bd$$

   $$bb \quad bq \quad qd$$

   is valid if the machine moved to the left, and similarly the block

$$aq$$

$$ab$$

is valid for a machine that moves one cell to its right. On the other hand 2×3 rectagles get around this problem, as they contain full information about the head movement. In the above example, the following blocks will be found to be invalid:

$$aqb\ bbd$$

$$abb\ bqd$$

4. (15 points) Problem 7.43, page 327 from Sipser text.

   For a CNF formula with n variables and m clauses, show that you can construct in polynomial time an NFA with $O(nm)$ states that accepts all non-satisfying assignments, represented by Boolean strings of length n. Conclude that the problem of minimizing NFAs cannot be done in polynomial time unless $\mathcal{P} = \mathcal{NP}$.

   Solution: The NFA should accept iff the input is a non-satisfying assignment. The NFA guesses a clause that is not satisfied by the given assignment and verifies the guess by reading the input.

   Let the number of variables in the formula be n and the number of clauses be m. For each clause C_i, the NFA has $n+1$ states labeled $C_{i,j}$, for $1 \leq j \leq n+1$ with a transitions from $C_{i,j}$ to $C_{i,j+1}$ for $1 \leq j \leq n$. The label on the transition $C_{i,j}$ to $C_{i,j+1}$ is:

   - 0 the j-th variable appears as a positive literal,
   - 1 the j-th variable appears as a negative literal, and
   - both 0 and 1 if the j-th variable does not appear in the clause.

   All the states $C_{i,n+1}$ for all i are accept states. There is a start state s form which there is an ϵ-transition to each of the m states $C_{i,1}$.

   If F is satisfiable there is some truth assignment satisfying all clauses. The NFA will not accept the input representing this assignment.

   If F is not satisfiable the NFA will accept all its inputs.

   If this NFA could be minimized in polynomial time then we can determine if it accepts all strings of length n which is equivalent to deciding if F is not satisfiable in polynomial time.

5. Let $G = (V, E)$ be an undirected graph. Let C be a collection of pairs of edges of G. A matching M in G is said to *respect* C if at most one edge from each pair in C is in M.

   Define the *Matching with Forbidden Edges* problem as follows:

   INPUT: An undirected graph $G = (V, E)$, a collection C of pairs of edges of G, and an integer $1 \leq k \leq |E|$.

   PROPERTY: G has a matching M of size k that respects C.

   The *Independent Set* problem is defined as follows:

   INPUT: An undirected graph $G = (V, E)$, and an integer $1 \leq k \leq |V|$.

   PROPERTY: G has an independent set of size k.

   (15 points) Give a polynomial time mapping reduction from the *Independent Set* problem to the *Matching with Forbidden Edges* problem.

   Solution: Given a graph $G = (V, E)$ and an integer k, output a graph $G' = (V', E')$ and an integer k' as follows. For each vertex $u \in V$ create two vertices u_1, u_2 and an edge between them. For each edge $\{u, v\}$ in E, add the pair of edges $(\{u_1, u_2\}, \{v_1, v_2\})$ in E' to C.

   By construction, E' is itself a matching. Each edge in E' corresponds to a vertex in V. Each pair in C corresponds to an edge in E. Let M be a subset of E' that respects C. The vertex subset of V that corresponds to M then can include only one end point of any edge in E.

6. Given two $n \times n$ matrices A and B with 0/1 entries, two $n \times 1$ vectors X and Y with 0/1 entries, and a natural number k, the problem is to decide if the maximum value in the vector $A \times X + B \times Y$ is at most k. (Here $A \times X$ is the product of the matrix A with the vector X. Similarly, $B \times Y$ is the product of the matrix B with the vector Y.)

   (10 points) Show that this problem is in $\mathcal{L}$.

   (See section 8.4, pages 320-321 of the Sipser text for the definition of the class $\mathcal{L}$.)

   Solution: We cannot compute $A \times X$ first, compute $B \times Y$ next, add the results, and then find the maximum since that would take too much space. Instead, do the following:

   - Set current maximum value to be 0.
   - For $1 \leq i \leq n$ do:
     - Form the i-th component of the vector $A \times X$, form the i-th component of $B \times Y$, add the results to get the i-th component of the sum.
     - Change the current maximum value, if necessary, based the sum value just computed.
   - Finally, compare the maximum value computed with k.

   Space used: current maximum value, the result of the current component of $A \times X$, the result of the current component of $B \times Y$, and their sum. Since the maximum magnitude of these values is $O(n)$, the algorithm uses $O(\log n)$ space.