Homework 6

ALECK ZHAO

April 19, 2018

1. Show that a language can be decided by a TM (i.e., can be recognized by some TM that halts on all inputs), if and only if some machine can enumerate the language (i.e., enumerate all strings in this language) in lexicographic order. You may assume the alphabet is $\{0, 1\}$.

    Proof. ($\implies$): Let $w_1, w_2, \cdots$ be a lexicographic ordering of the strings in a language L that can be decided by a TM R. Then we can define such an enumerator:

    (1) For $i = 1, 2, \cdots$:

        (a) Simulate R on w_i.

        (b) If R accepts w_i, print w_i. Otherwise, continue.

    Since R decides L, it is guaranteed to halt on all inputs w_i, so this machine prints all strings in L in lexicographic order.

    ($\impliedby$): Suppose L can be enumerated in lexicographic order by a machine E. If L is finite, then it is decidable. If L is infinite, then construct the TM R as follows:

    $R =$ on input w:

    (a) Let E print the next string x in L.

        (a) If $x = w$, accept.

        (b) Else if x is lexicographically after w, reject.

        (c) Else (if x is lexicographically before w), continue.

    Since L is infinite, but w has a finite lexicographic position, this machine must halt at some point, and since w was arbitrary, E halts on all inputs in L. $\qquad\square$

2. Let $T = \{\langle M \rangle \mid M \text{ is a TM that accepts } \alpha^{\mathcal{R}} \text{ whenever it accepts } \alpha\}$. Show that T is undecidable.

Proof. Suppose T is decidable, and that R is a decider. Construct a TM S to decide A_{TM} as follows: $S = $ on input $\langle M, w \rangle$ where M is a TM and w is a string:

(1) Construct a TM N as follows:

$N = $ on input x :

    (a) Simulate M on w.

    (b) If M accepts w and $x = 01$, then N accepts. Otherwise, N rejects.

    (c) If M rejects w, then N rejects.

(2) Simulate R on $\langle N \rangle$.

(3) If R rejects, then S accepts, otherwise S rejects.

Now, if M accepts w, then N only accepts $\{01\}$, so R would reject $\langle N \rangle$, and thus S would accept $\langle M, w \rangle$. On the other hand, if M rejects w, then N rejects every string, so R would accept $\langle N \rangle$, and thus S would reject $\langle M, w \rangle$. Thus, S would be TM that decides A_{TM}, which is a contradiction since E_{TM} is undecidable. $\qquad\square$

3. Define the language

$$C_{TM} = \{\langle M_1, M_2 \rangle \mid M_1, M_2 \text{ are two Turing machines such that } L(M_1) \subseteq L(M_2)\}$$

Show that C_{TM} is undecidable.

Proof. Suppose that C_{TM} was decidable by R. Then construct the TM S to decide E_{TM} as follows:
$S =$ on input $\langle M \rangle$ where M is a TM:

(1) Run R on input $\langle M, M_1 \rangle$ where M_1 is a TM that rejects all inputs.
(2) If R accepts, S accepts, otherwise S rejects.

Now, if R accepts $\langle M, M_1 \rangle$, then $L(M) \subseteq L(M_1) = \varnothing \implies L(M) = \varnothing$, so S would correctly accept M, and likewise if R rejects $\langle M, M_1 \rangle$, then $L(M) \neq \varnothing$, so S would correctly reject. Thus, S would be a TM that decides E_{TM}, which is a contradiction since E_{TM} is undecidable. $\qquad\square$

4. Prove that the following language is in P.

$$2COL = \{G : \text{graph } G \text{ has a coloring with 2 colors}\}$$

Here a coloring of G with c colors is an assignment of a number in $\{1, \cdots, c\}$ to each vertex such that no adjacent vertices get the same number.

Proof. Consider the following algorithm, where v is an arbitrary starting vertex:

(1) Begin by marking every vertex with 0.

(2) Mark v with color 1.

(3) For all vertices w with an edge to v :

    (a) If w has color 0, then give it the opposite color of v.

        (i) Recursively call this coloring subroutine on w and its neighbors.

    (b) If w has the same color as v, reject.

(4) Accept

This is a modification of the DFS graph search algorithm, which runs in polynomial time in the size of the graph. If we reach the end of the algorithm and accept, then every vertex will have been reached (assuming a connected graph), and we will have a 2-coloring since the only time this algorithm rejects is when adjacent vertices have the same color, and vice versa. Thus, $2COL$ is in P. $\square$