Assignment 8

Nishan Dhoj Karki

10/24/2022

1 6.21) Show how to compute the descriptive complexity of strings $K(x)$ with an oracle for A_{TM}.

Proof:

DEFINITION 6.23
Let x be a binary string. The minimal description of x, written $d(x)$, is the shortest string $<M, w>$ where TM M on input w halts with x on its tape. If several such strings exist, select the lexicographically first among them. The descriptive complexity of x, written $K(x)$, is $K(x) = |d(x)|$.

DEFINITION 6.18
An oracle for a language B is an external device that is capable of reporting whether any string w is a member of B. An oracle Turing machine is a modified Turing machine that has the additional capability of querying an oracle. We write M^B to describe an oracle Turing machine that has an oracle for language B.

The descriptive complexity of a string is at most a fixed constant more than its length given by $K(x) \leq |x|+c$, where c is a universal constant not dependent on the string. This sets the upper bound for the string generation on the description below. The following description will compute descriptive complexity of $K(x)$ using an oracle for A_{TM}:

On input x:

1. Enumerate all binary strings s until $|s| \leq |x| + c$ in lexicographic order.
2. Parse s as $\langle M, w\rangle$, where M is a turing machine and w is a string. If parsing fails move to another string s.
3. Ask the oracle for A_{TM} if $\langle M, w\rangle \in A_{TM}$.
4. If the oracle answers NO, move to another string.
5. If the oracle answers YES,
   - Run M on w.
   - if M halts on w with x in the tape, then s is the minimal description.
6. Compute $K(x) = |s|$ and output it.

We will eventually output the shortest description as we are generating the strings in lexicographic order and we have set an upper bound. Here, the above description computes the descriptive complexity of strings K(x) with an oracle for A_{TM}.

2 2. Consider the languages $HALT_{TM}$ (from Theorem 5.1) and E_{TM} (from Theorem 5.2). Prove the following statement: $E_{TM} \leq_T HALT_{TM}$

Proof:

THEOREM 5.1,
$HALT_{TM} = \{\langle M\rangle | M$ is a TM and M halts on input $w\}$.
$HALT_{TM}$ is undecidable.

THEOREM 5.2,
$E_{TM} = \{\langle M\rangle | M$ is a TM and $L(M) = \emptyset\}$.
E_{TM} is undecidable.

Claim: $E_{TM} \leq_T HALT_{TM}$
Proof:
Let us construct an oracle Turing Machine $T^{HALT_{TM}}$ such that:
$T^{HALT_{TM}} =$ "On input $\langle M\rangle$ where M is a TM:
1. The following Turing Machine U is constructed.
   $U =$ "On any input:
      (a) Run M in parallel on all strings in $\sum^*$
      (b) if M accepts any of these strings, ACCEPT."
2. Query the oracle to determine whether $\langle U, O\rangle \in HALT_{TM}$
3. If the oracle answers NO, ACCEPT.
4. If the oracle answers YES, REJECT."

$T^{HALT_{TM}}$ will accept if the language of M is empty as U will loop and oracle will answer NO. If the language of M is not empty, U will accept every input particularly O, and the oracle will answer YES and $T^{HALT_{TM}}$ will reject. Here we have shown E_{TM} is decidable relative to $HALT_{TM}$. Hence, $E_{TM} \leq_T HALT_{TM}$.