CSC320 Assignment Four Report

2018-07-22
Name: Zhaocheng Li
ID: V00832770

Question 1:

Proof idea: Proving from two directions, applying the proof of definition on the Turing-recognizability and enumerator shown on slide 11th.

Assumption: Suppose we hypothetically have a TM M that decides L, and an enumerator E for L, such that E respects $\propto$ ordering.

PROOF:

(>>>). Given the TM M that decides L, we can try to construct such E for L.

1. *E ignores any input*
2. *For I = 1, 2, 3, …:*
   - *For each possible string s in $\{0,1\}^*$ of length less than or equal to I:*
     i. *Run M on s for I steps.*
     ii. *If M accepts, print s.*

Then such E exists with all outputs is printed in $\propto$ order, and they belong to L.

(<<<). Given the enumerator E for L and it respect the $\propto$ ordering. We can try to construct a TM M that decides L.

Here there are two cases: L is finite, or L is infinite.

Case 1: suppose L is finite. Then it is decidable because all finite languages are decidable, by theorem. Such TM M that decides L obviously exists.

Case 2: If A is infinite, a decider for L works as follow:

1. *Take w as the input of M, then the decider enumerates all strings in L in order until some string in order after w appears. This is because L is infinite.*
2. *If w has shown in the enumeration already, accept; if it has not appeared yet, it never will, reject.*

Then the TM M for L is made.

Proof done.

Question 2:

Proof idea: following the definition of undecidability of A_{TM} and the hint.

PROOF:

We know that

L = {<M> | M when started on the blank tape, eventually writes a $ somewhere on the tape}

And by the hint (i.e., give a computable reduction f, such that f (<M, w>) = <M_1>.

For any M, string w we can let M_1 be the TM which takes as input string x:

1. If $w = x$, M_1 runs M on input w and accepts if M accepts;
2. If $w \neq x$, M_1 rejects;

Now we construct TM S to decide A_{TM}. Let R be a hypothetical TM which decides L:

S has input <M, w>

1. Use <M, w> to construct <M_1> as described above;
2. Run R on <M_1> and accept if and only if R accepts.

If R decides L, then S decides A_{TM}. Therefore, R cannot exist and L is undecidable.

Question 3:

Proof Idea: Hint on the computable reduction f

PROOF:

Given the $HALT_{TM}$ and L, construct f that $HALT_{TM} \leq_m L$

For any M, string w we can let M_1 be the TM:

1. If w is less than j 1's, M_1 runs M on input w and accepts if M accepts;
2. If w is more than j 1's, M_1 rejects;

Now we construct TM S to decide $HALT_{TM}$. Let R be a hypothetical TM which decides L:

S has input <M, w>

1. Use <M, w> to construct <M_1> as described above;
2. Run R on <M_1> and accept if and only if R accepts.

If R decides L, then S decides $HALT_{TM}$. Therefore, R cannot exist and L is undecidable.

Question 4:

PROOF:

Since $E_{TM} = \{\langle M\rangle \mid M \text{ is a TM and } L(M) = \phi\}$,

then we E_{TM}^* (meaning complement) = $\{\langle M\rangle \mid M \text{ is a TM and } L(M) \neq \phi\}$.

For a nondeterministic TM H for E_{TM}^*,

For i = 1, 2, 3, …:

1. Run M for i steps, non-deterministically choose an input $x_1 \in E_{TM}^*$,
2. H accepts if M accepts on any input.

Question 5:

Proof Idea: applying reduction with E_{TM}

PROOF:

Suppose $H = \{\langle M_1, M_2\rangle \mid L(M_1) \cap L(M_2) = \phi\}$

We show $E_{TM} \leq_m H$.

Define f as follow: $f(\langle M, w\rangle) = \langle M_1, M_2\rangle$, where M_1, M_2 are machines such that:

1. M_1 accepts all inputs;
2. For any input x, M_2 runs M on w and rejects if M accepts.

If $\langle M, w\rangle \in E_{TM}$ then M accepts w. But then $L(M_1) = \phi$ and $L(M_2) = \Sigma^*$, so $\langle M_1, M_2\rangle \in H$.

If $\langle M, w\rangle \notin E_{TM}$ then $L(M_1) \cap L(M_2) \neq \phi$, so $\langle M_1, M_2\rangle \notin H$.

So, f is a mapping reduction from E_{TM} to H, and H is not recognizable since E_{TM} isn't.

Question 6:

Proof Idea: Using the hint.

PROOF:

Now we know that

$T = \{\langle M\rangle \mid M \text{ is a reversible TM}\}$

And by the hint (i.e., give a computable reduction f, such that f (<M, w>) = <M_1>.

For any M, string w we can let M_1 be the TM which takes as input string x:

1. If $w = x = x^R$, M_1 runs M on input w and accepts if M accepts;
2. If $x \neq x^R$, M_1 runs M on input w and accepts x if M accepts x^R;
3. If $w \neq x$, M_1 rejects;

Now we construct TM S to decide A_{TM}. Let R be a hypothetical TM which decides T:

S has input <M, w>

1. Use <M, w> to construct <M_1> as described above;
2. Run R on <M_1> and accept if and only if R accepts.

If R decides T, then S decides A_{TM}. Therefore, R cannot exist, and T is undecidable.