IITM-CS6014 : Advanced Theory of Computation   Author : Karthik Abinav S (CS10B057)

Problem Set #3

1. (10 points) Recall

$$K = \{x \mid M_x \text{ halts on } x\}$$

Prove the following.

(a) K is SD-complete.

Answer :

To show this we have to show :

- $K \in SD$
- $\forall L \in SD$, $L \leq_m K$

To show the first part, we can re write K as follows:
$K = \{ \ x : \exists t \ M_x \text{ halts on } x \text{ in } t \text{ steps} \ \}$
Since "M_x halts on x in t steps" is decidable, this shows that $K \in \Sigma_1$.

To show the second part, it suffices to show that $HP \leq_m K$.(Because we know that HP is $\Sigma_1 - complete$)
To show this consider the following machine M' .
M' copies its input y on a seperate track.
It simulates M on x . If M halts on x , then simulate M_y on y. Accept , if M_y halts on y.

So the Language of M' can be described as :

$$M' = \begin{cases} K & \text{if } M \text{ halts on } x \\ \varnothing & \text{if } M \text{ does not halt on } x \end{cases}$$

This is the required reduction for $HP \leq_m K$.
Therefore , K is Σ_1 complete.

(b) Let $K^X = \{x \mid M_x^X \text{ is an oracle TM with oracle } X \text{ and } M_x^X \text{ halts on } x\}$. Show that K^K is Σ_2-complete.

Answer:

To show this we have to show :

- $K^K \in \Sigma_2$
- $\forall L \in \Sigma_2$, $L \leq_m K^K$

Simulate input x on M_x^K . Whenever a query is needed, use the oracle K provided to machine M_x^K . The language accepted by this machine is K^K . And since , it is semi decidable with respect to a language in Σ_1 , therefore it is in Σ_2.

To show that every language $L \in \Sigma_2$, $L \leq_m K^K$, it suffices to show that $MP_2 \leq_m K^K$(Because MP_2 is Σ_2 - complete). Consider a machine M which accepts MP_2 with an oracle access to K(Note from previous question that K is Σ_1 - complete.

Construct a machine N, which does the following :

- Runs M on input x with query to the oracle K. If it accpets, then run M_y^K on input y. Whenever a query is needed , use the oracle from the machine M. So the Language of N can be described as :

$$N = \begin{cases} K^K & \text{if } M^K \text{ accepts } x \\ \varnothing & \text{if } M^K \text{ does not accept } x \end{cases}$$

  This is the required reduction for $MP_2 \leq_m K^K$.

  Therefore, K^K is Σ_2 - complete.

2. (10 points) A set P is *partially productive* if there is a partial recursive function(i.e. computed by a Turing machine N - which need not be total) ψ called the *productive function* such that:

$$\forall x \quad (\mathcal{W}_x \subseteq P \implies N \text{ halts on } x \;\&\; \psi(x) \in P \setminus \mathcal{W}_x)$$

   Show that any productive set P has an injective recursive productive function.
   (Hint : First prove that it can be made recursive and then attempt on making it injective).

   Answer:

3. (5 points) Prove that there must exist a recursive function such that $\{\mathcal{W}_{f(n)}\}_{n\in\mathbb{N}}$ consists of precisely decidable sets.

   Answer:
   Let n_0 be the encoding of the trivial turing machine which accepts all string in Σ^*.
   Consider the function $f(n) = n_0, \forall n \in N$.
   $\{W_{f(n)}\} = \Sigma^*$ for $\forall n \in N$. Hence, this set has only decidable sets, for this particular recursive function.

4. (5 points) A simple set is *effectively simple* if there is a recursive function f such that:

$$\forall n \in \mathbb{N} : \mathcal{W}_n \subseteq \overline{A} \implies |\mathcal{W}_n| \leq f(n)$$

   Show that Post's simple set is effectively simple.

   (Extra Credit) If a set A is effectively simple, argue that $K \leq_T A$. This justifies why Friedberg-Muchnik had to do a different construction. (Hint: Normal homework rules does not apply to this question. You can look up anywhere, but cite your sources. !).

Answer:

5. (10 points) Consider the following computational problems :

   (a) Given a Turing machine M, test if $L(M)$ is Productive.

   (b) Given a Turing machine M, test if $L(M)$ is Simple.

   Are they decidable? semi-decidable?

   (Extra Credit) Place them in the arithmetic hierarchy.

Answer(a)
This language is decidable.
Productive set P is defined as for a effectively computable σ,
$\forall x,\, W_x \subseteq P \Longrightarrow \sigma(x) \in P \setminus W_x$

But , since here P is semidecidable , there fore for some particular value of $x = t$, W_t $= P$. And hence , the condition above will not satisy for this value of x. i.e.
$\forall x,\, P \subseteq P \Longrightarrow \sigma(x) \in \varnothing$.
And since, no element can belong to the set $\varnothing$, hence the property is always false. This can be easily decided by a total TM.

Answer(b)
This language is not semidecidable. This can be shown using Rice Theorem 2.

The property is a non monotone property.
Consider the language Σ^*. This is not a simple set (because $\overline{L}$ is not infinite). But in class we proved that there exists a simple set. Since, simple sets are semi decidable sets and it is a subset of Σ^* , therefore, it is not monotone. Hence from rice theorem 2 , its not semidecidable.