INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD
Department of Computer Science and Engineering
CS5616 Computational Complexity
January – May 2024

Problem Set – 3
Name: Neeraj Krishna N
Roll no: 112101033

Total Points – 50
Given on 29 Feb
Due on 11 Mar

Instructions

- Use of resources other than class notes and references is forbidden.
- Collaboration is not allowed. Credit will be given for attempts and partial answers.

1. (10 points) (**RE vs co-RE**) Show that the set

$$\{M \mid M \text{ halts on all inputs of length less than } 42\}$$

is recursively enumerable, but is its complement is not.

Solution:

Let us construct a turing machine M' for which

$$L(M') = \{M \mid M \text{ halts on all inputs of length less than } 42\}$$

M' = "

On input M,

(1) For each $x \in \Sigma^*, |x| < 42$, Run M on x

(2) Accept M

"

$M \in L$

$\Rightarrow M$ halts on all inputs of length less than 42

$\Rightarrow$step (1) of M terminates and reaches step (2)

$\Rightarrow M'$ accepts M

$\Rightarrow M \in L(M')$

$M \notin L$

$\Rightarrow M$ loops on some input of length less than 42

$\Rightarrow$step (1) of M' loops when it reaches such an x

$\Rightarrow M'$ loops

$\Rightarrow M'$ does not accept M

$\Rightarrow M \notin L(M')$

Hence $L(M') = L$

Suppose complement of L is also recursively enumberable, then L would be decidable since we know

$$L \text{ is decidable} \Leftrightarrow L \text{is r.e and } \overline{L} \text{ is r.e}$$

we will show a reduction $\mathsf{HP} \leq_m L$

The reduction machine σ on input $M\#x$ (an instance of HP) outputs M' an instance of L

$M' =$ "

On input y,

1. Run M on x
2. If M halts on x, accept y

"

we need to prove

$M\#x \in \mathsf{HP} \Leftrightarrow M' \in L$

$M\#x \in \mathsf{HP}$

$\Rightarrow M$ halts on x

$\Rightarrow M'$ accepts all y, hence M' halts on all inputs of length less than 42

$\Rightarrow M' \in L$

$M\#x \notin \mathsf{HP}$

$\Rightarrow M$ loops on x

$\Rightarrow M'$ loops on all y, hence M' halts on some inputs of length less than 42

$\Rightarrow M' \notin L$

therefore $M\#x \in \mathsf{HP} \Leftrightarrow M' \in L$

Since we know HP is undecidable, L is also undecidable, and thus $\overline{L}$ is not r.e

2. (10 points) (**Alternate definition for Δ_i**) Let A be any language. Define $\mathcal{D}^A$ be the class of all languages L such that L is decidable in A. Similarly, $\mathcal{SD}^A$ be the class of all L such that L is semi-decidable in A and $\mathsf{co}\mathcal{SD}^A$ be the class of all languages whose complement is in $\mathcal{SD}^A$.
   (a) (5 points) Show that $\mathcal{D}^A = \mathcal{SD}^A \cap \mathsf{co}\mathcal{SD}^A$.

(b) (5 points) For any $i \geq 1$, by definition, $\Delta_i = \Sigma_i \cap \Pi_i$. Show that

$$\Delta_i = \{L \mid \text{ there exists } A \in \Sigma_{i-1} \text{ such that } L \text{ is decidable in A}\}.$$

Solution:

(a) Suffices to prove :

(i) $\mathcal{D}^A \subseteq \mathcal{SD}^A \cap co\mathcal{SD}^A$

(ii) $\mathcal{D}^A \supseteq \mathcal{SD}^A \cap co\mathcal{SD}^A$

(a) Let $L \in \mathcal{D}^A$, by definition of D^A, L is decidable in A, let the total T.M be M^A

L is decidable in A

$\Rightarrow L$ is r.e in A

$\Rightarrow L \in \mathcal{SD}^A$

to prove $\overline{L} \in \mathcal{SD}^A$, we need to construct a T.M N^A such that $L(N^A) = \overline{L}$

$N^A =$ "

On input x,

i. Run M^A on x, (using the oracle A to answer queries of the form $x \in A$?)

ii. If M^A accepts, reject

iii. If M^A rejects, accept

"

The above machine is total since every step is total including the step (i) since M^A is total

$x \in \overline{L}$

$\Leftrightarrow x \notin L$

$\Leftrightarrow M^A$ rejects x

$\Leftrightarrow N^A$ accepts x

$\Leftrightarrow x \in L(N^A)$

Hence $L(N^A) = \overline{L}$

and thus $\overline{L}$ is decidable in A

$\Rightarrow \overline{L}$ is r.e in A

$\Rightarrow \overline{L} \in \mathcal{SD}^A$

$\Rightarrow L \in co\mathcal{SD}^A$

and hence proved that $L \in \mathcal{D}^A \Rightarrow L \in \mathcal{SD}^A \cap co\mathcal{SD}^A$

$\Rightarrow \mathcal{D}^A \subseteq \mathcal{SD}^A \cap co\mathcal{SD}^A$

(b) Let $L \in \mathcal{SD}^A \cap co\mathcal{SD}^A$

$\Rightarrow L \in \mathcal{SD}^A$ and $\overline{L} \in \mathcal{SD}^A$

$\Rightarrow$there exists an OTM N_1^A accepting L and and also N_2^A accepting $\overline{L}$ now let us construct an OTM N^A such that it accepts L and is total

$N^A =$ "

On input x,

i. Run N_1^A and N_2^A on x with time sharing, and also whenever a query of the form $x \in A$ is made, use the oracle A

ii. If N_1^A accepts, accept

iii. If N_2^A accepts, reject

"

$\forall x \in \Sigma^*, x \in L \vee x \in \overline{L}$ but not both

$\Rightarrow$either N_1^A accepts x or N_2^A accepts x

Since for each input, one of the machines is guaranteed to accept, the machine N^A halts on all inputs and hence is total

To prove $L(N^A) = L$

$x \in L$

$\Rightarrow N_1^A$ accepts x

$\Rightarrow N^A$ accepts x

$\Rightarrow x \in L(N^A)$

$x \notin L$

$\Rightarrow x \in \overline{L}$

$\Rightarrow N_2^A$ accepts x

$\Rightarrow N^A$ rejects x

$\Rightarrow x \notin L(N^A)$

Therefore $L = L(N^A)$ and hence proved that $\mathcal{SD}^A \cap co\mathcal{SD}^A \subseteq \mathcal{D}^A$

Hence $D^A = \mathcal{SD}^A \cap co\mathcal{SD}^A$

(b)

3. (10 points) (**Closure properties of Σ_n, Π_n**). Fix any $i \geq 1$. Show that Σ_i as well as Π_i are closed under intersection and union.

Solution:

Claim 1 : Σ_i closed under union $\Rightarrow \Pi_i$ closed under intersection

Claim 2 : Σ_i closed under intersection $\Rightarrow \Pi_i$ closed under union

Proof of Claim 1 :

Σ_i closed under union

$\Rightarrow \forall L_1, L_2 \in \Sigma_i, L_1 \cup L_2 \in \Sigma_i$

$\Rightarrow \forall L_1, L_2 \in \Sigma_i, \overline{(\overline{L_1} \cap \overline{L_2})} \in \Sigma_i$, by De Morgan's Leftrightarrow

$\Rightarrow \forall \overline{L_1}, \overline{L_2} \in \Pi_i, (\overline{L_1} \cap \overline{L_2}) \in \Pi_i$

$\Rightarrow \forall L_1', L_2' \in \Pi_i, (L_1' \cap L_2') \in \Pi_i$

$\Rightarrow \Pi_i$ closed under intersection

Proof of Claim 2 :

Σ_i closed under intersection

$\Rightarrow \forall L_1, L_2 \in \Sigma_i, L_1 \cap L_2 \in \Sigma_i$

$\Rightarrow \forall L_1, L_2 \in \Sigma_i, \overline{(\overline{L_1} \cup \overline{L_2})} \in \Sigma_i$, by De Morgan's Leftrightarrow

$\Rightarrow \forall \overline{L_1}, \overline{L_2} \in \Pi_i, (\overline{L_1} \cup \overline{L_2}) \in \Pi_i$

$\Rightarrow \forall L_1', L_2' \in \Pi_i, (L_1' \cup L_2') \in \Pi_i$

$\Rightarrow \Pi_i$ closed under union

Suffices to prove

Σ_i is closed under intersection and union

4. (20 points) (**Rice's theorem**) Identify if the following are (0) properties of SD languages, (1) non-trivial properties and (2) non-monotone properties. If (1) / (2) is true, apply Rice's theorems suitably and give your conclusions. A direct use of diagonalisation or reductions does not fetch any credit.
   (a) (4 points) given a Turing machine M, $L(M)$ is not regular ?
   (b) (4 points) given a Turing machine M, does there exist a non-empty regular set L' such that $L' \subseteq L(M)$?
   (c) (4 points) given M, does M represent a DFA that accepts some string with equal number of 0s and 1s ?
   (d) (4 points) given a Turing machine M, is $L(M) \in \Pi_{42}$?

Solution:

(a) $L = \{M \mid L(M) \text{ is not regular}\}$

is a property of SD language

If there are 2 machines M_1 and M_2 accepting the same language, then it cannot be the case that $L(M_1)$ is regular and $L(M_2)$ is not regular

This is non trivial since there are turing machines which accept regular languages and turing machines which accept non-regular languages

This is also non-monotone since, there exists L_1, L_2 such that $L_1 \subseteq L_2$ and $\mathcal{P}(L_1) = 1$ and $\mathcal{P}(L_2) = 0$

$L_2 = \Sigma^*$ and $L_1 = \{0^n 1^n \mid n \geq 0\}$

here L_1 is not regular which satisfies the property whereas L_2 is regular and a strict superset of L_1 which does not satisfy the property

Since the property is non-trivial, by Rice's Theorem 1, L is not decidable

Since the property is non-monotone, by Rice's Theorem 2, L is not semi-decidable

(b) $L = \{M \mid \exists L' \subseteq \Sigma^*, L' \neq \phi \text{ and} L' \subseteq L(M)\}$

This is equivalent to saying $L = \{M \mid L(M) \text{ is not empty}\}$ because

$L(M)$ is not empty

$\Rightarrow \exists L' \subseteq L(M)$ which is $L(M)$ itself, such that $L' \subseteq L(M)$ i.e $L' = L(M)$

$\exists L' \subseteq \Sigma^*, L' \neq \phi$ and$L' \subseteq L(M)$

$\Rightarrow L(M)$ is not empty

$L(M)$ is not empty is a property of SD languages since if there exists 2 machines M_1 and M_2 accepting the same language, it cannot be the case that

$L(M_1) = \phi$ and $L(M_2) \neq \phi$ or vice-versa

This is a non-trivial property, since there are turing machines accepting empty languages and there are turing machines accepting non-empty languages

This is a monotone property because

$\forall L_1, L_2 \in \Sigma^*, \mathcal{P}(L_1) = 1 \Rightarrow \mathcal{P}(L_2) = 1$

because if L_1 is non-empty, any superset of L_1 will be non-empty

Since, this property is non-trivial, by Rice's theorem 1, the given language is undecidable

We cannot conclude anything about semi-decidability from Rice's Theorem 2 because the language is not non-monotone

(c)

(d)