IIT PALAKKAD

Indian Institute of Technology Palakkad
Department of Computer Science and Engineering
CS5616 Computational Complexity
January May 2024

Problem Set – 2 | Total Points – 50
Name: Neeraj Krishna N | Given on 09 Feb
Roll no: 112101033 | Due on 16 Feb

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) [**Properties of** $\leq_m$] Show that $\leq_m$ relation is reflexive and transitive over languages in Σ^*. Is $\leq_m$ symmetric ? Argue.

Solution:

1. To prove $\leq_m$ is reflexive :

   Let L be a language, we have to prove $L \leq_m L$. Let M be a turing machine which accepts L

   Now we have to prove that if L is decidable, then L is decidable, which is trivial

2. To prove $\leq_m$ is transitive :

   Let L_1, L_2, L_3 be 3 languages such that $L_1 \leq_m L_2$ and $L_2 \leq_m L_3$, we have to prove that $L_1 \leq_m L_3$

   Let M_{12} be the reduction machine which converts an instance x of L_1 to an equivalent instance y of L_2 and M_{23} be the reduction machine which converts an instance y of L_2 to an equivalent instance z of L_3 (by definition of reduction machines both M_{12} and M_{23} are total)

   We will construct a reduction machine M_{13} which converts an instance x of L_1 to an equivalent instance z of L_3

   $M_{13} =$

   "

   On input x,

   (a) Run M_{12} on x and let y denote the output
   (b) Run M_{23} on y and let z denote the output
   (c) Return z

   "

   This machine M_{13} is total since both M_{12} and M_{23} are total

Let M_3 be a turing machine which accepts L_3

Now to prove $x \in L_1 \Leftrightarrow z \in L_3$

$x \in L_1 \Leftrightarrow y \in L_2$ since M_{12} converts instance x of L_1 to an equivalent instance y of L_2

and

$y \in L_2 \Leftrightarrow y \in L_3$ since M_{23} converts instance y of L_2 to an equivalent instance z of L_3

Hence

$x \in L_1 \Leftrightarrow z \in L_3$

Therefore the machine M_{13} converts an instance x of L_1 to an equivalent instance z of L_3. Hence this is the required reduction machine

3. $\leq_m$ is not symmetric

   Example : Let $L = 0^*$, we know that L is decidable, Let M be the total turing machine which accepts L

   We also know that $HP \not\leq_m L$, since then it would be the case that HP is decidable which is not the case

   To prove $L \leq_m HP$

   Let the reduction machine be M_σ which on input x outputs $\langle M', x\rangle$ where desciption of M' is as follows

   $M' =$

   "

   On input y,

   (a) Run M on x
   (b) If M accepts, halt and accepts
   (c) If M rejects, loop

   "

   we will prove that $x \in L \Leftrightarrow y \in HP$

   $x \in L$

   $\Rightarrow M$ accepts x

   $\Rightarrow M'$ halts on every input y, in particular x

   $\Rightarrow \langle M', x\rangle \in HP$

   $x \notin L$

   $\Rightarrow M$ will reject every input x (since M is total)

   $\Rightarrow M'$ will loop on every input y, in particular x

   $\Rightarrow \langle M', x\rangle \notin HP$

   Hence we have showed two languages L and HP such that

   $L \leq_m HP$ but $HP \not\leq_m L$

   Therefore, $\leq_m$ is not symmetric

2. (20 points) [**Reduction by containment !**] Let $L_1, L_2 \subseteq \Sigma^*$ where $L_1 \subseteq L_2$. Consider the statements **(1)** "$L_2 \leq_m L_1$" **(2)** "$L_1 \leq_m L_2$"

   (a) (5 points) Give a pair of languages where **(1)** is true. Provide appropriate justification if no such pair exists.

   (b) (5 points) Give a pair of languages where **(2)** is true. Provide appropriate justification if no such pair exists.

   (c) (10 points) Prove or disprove the following statements (a), (b).

      (a) "for any L_1, L_2 with $L_1 \subseteq L_2$, **(1)** is true."

      (b) "for any L_1, L_2 with $L_1 \subseteq L_2$, **(2)** is true."

Solution:

(a) We know that for all $L_1, L_2 \in$ decidable languages, $L_1 \leq_m L_2$

Nevertheless, for this question, let $L_1 = 0^*$ and $L_2 = 1^*0^*$

Here, $L_1 \subseteq L_2$

The reduction machine on input x, an instance of L_1 produces an instance y of L_2

To prove : $L_2 \leq_m L_1$

Let M_1 be the total turing machine which accepts L_1, we know it exists, because L_1 is **regular**

The reduction machine M_σ Is

"

On input x

(a) Run M_1 on x

(b) If M_1 accepts, output 10

(c) If M_1 rejects, output 01

"

Now, we have to prove $x \in L_1 \Leftrightarrow M_\sigma(x) \in L_2$

$x \in L_1$

$\Rightarrow M_1$ accepts x

$\Rightarrow M_\sigma$ outputs 10 which is in L_2

$\Rightarrow 10 \in L_2$

$\Rightarrow M_\sigma(x) \in L_2$

$x \notin L_1$

$\Rightarrow M_1$ rejects x, since M_1 is total

$\Rightarrow M_\sigma(x) = 10$ which is not in L_2

$\Rightarrow M_\sigma(x) \notin L_2$

Hence given $L_1 = 0^*$ and $L_2 = 1^*0^*$ such that $L_1 \subseteq L_2$ and $L_1 \leq_m L_2$

(b) Let $L_1 = 0^*$ and $L_2 = 1^*0^*$

Here, $L_1 \subseteq L_2$

The reduction machine on input x, an instance of L_2 produces an instance y of L_1

To prove : $L_1 \leq_m L_2$

Let M_2 be the total turing machine which accepts L_2, we know it exists, because L_2 is **regular**

The reduction machine M_σ Is

"

On input x

(a) Run M_2 on x
(b) If M_2 accepts, output 0
(c) If M_2 rejects, output 1

"

Now, we have to prove $x \in L_2 \Leftrightarrow M_\sigma(x) \in L_1$

$x \in L_2$

$\Rightarrow M_2$ accepts x

$\Rightarrow M_\sigma$ outputs 0 which is in L_1

$\Rightarrow 0 \in L_1$

$\Rightarrow M_\sigma(x) \in L_1$

$x \notin L_2$

$\Rightarrow M_2$ rejects x, since M_2 is total

$\Rightarrow M_\sigma(x) = 1$ which is not in L_1

$\Rightarrow M_\sigma(x) \notin L_1$

Hence given $L_1 = 0^*$ and $L_2 = 1^*0^*$ such that $L_1 \subseteq L_2$ and $L_2 \leq_m L_1$

(c) (a) This is false, because, let $L_1 = \phi$ and $L_2 = HP$, we know that L_1 is decidable whereas L_2 is not decidable, (and also $L_1 \subseteq L_2$)
$L_2 \nleq_m L_1$ because if $L_2 \leq_m L_1$, then $L_2 = HP$ would have been decidable, which we know it is not the case

(b) This is also false, because, let $L_1 = HP$ and $L_2 = \Sigma^*$
$L_1 \subseteq L_2$ and L_1 is undecidable and L_2 is decidable
$L_1 \nleq_m L_2$ because if $L_1 \leq_m L_2$, then $L_1 = HP$ would have been decidable, which we know it is not the case

3. (10 points) [**Proof of Rice's theorem 1**] In the proof of Rice's theorem for showing undecidability done in class, we assumed that the non-trivial property $\mathcal{P}$ is false for $\emptyset$. Explain how can this assumption be removed. [*Hint: Use* $\overline{\mathsf{HP}}$ *!*]

Solution:

Suppose $\mathcal{P}$ is non-trivial and $\mathcal{P}(\phi) = 1$, we want to prove that $\mathcal{P}$ is undecidable

Assume $\mathcal{P}$ is decidable, and M be the total turing machine accepting $\mathcal{P}$

Since $\mathcal{P}$ is decidable, $\overline{\mathcal{P}}$ has to be decidable, hence if $\mathcal{P}(\phi) = 1$, then $\overline{\mathcal{P}}(\phi) = 0$ and $\overline{\mathcal{P}}$ is also non-trivial

Now, $\exists L_1$ such that $\overline{\mathcal{P}}(L_1) = 1$ and L_1 will not be ϕ since $\overline{\mathcal{P}}(\phi) = 0$. Let M_1 be a machine accepting L_1 (existence of M_1 is guaranteed since L_1 is semi-decidable)

We will prove that $\mathsf{HP} \leq_m \overline{\mathcal{P}}$

Let the reduction algorithm on input $\langle M, x\rangle$, an instance of HP output M', an instance of $\overline{\mathcal{P}}$

Description of M' is as follows

$M' =$

"

On input y,

1. Run M on x
2. If M halts on x
   (a) Run M_1 on y
   (b) If M_1 accepts, accept
   (c) If M_1 rejects, reject

"

We want to prove that $\langle M, x\rangle \in \mathsf{HP} \Leftrightarrow M' \in \overline{\mathcal{P}}$

$\langle M, x\rangle \in \mathsf{HP}$

$\Rightarrow M$ halts on x

$\Rightarrow M'$ accepts y only if M_1 accepts y

$\Rightarrow L(M') = L(M_1)$

$\Rightarrow \overline{\mathcal{P}}(L(M')) = \overline{\mathcal{P}}(L(M_1)) = 1$

$\Rightarrow M' \in \overline{\mathcal{P}}$

$\langle M, x\rangle \notin \mathsf{HP}$

$\Rightarrow M$ does not halt on x

$\Rightarrow M'$ loops on every input y

$\Rightarrow L(M') = \phi$

$\Rightarrow \overline{\mathcal{P}}(L(M')) = \overline{\mathcal{P}}(\phi) = 0$

$\Rightarrow M' \notin \overline{\mathcal{P}}$

Hence we have proved that $\mathsf{HP} \leq_m \overline{\mathcal{P}} \Rightarrow \overline{\mathsf{HP}} \leq_m \mathcal{P}$

Therefore, since $\overline{HP}$ is undecidable, $\mathcal{P}$ is also undecidable

4. (10 points) [**Rice's theorem ?**] Define the language $\mathsf{TOTAL} = \{M \mid M \text{ halts on all inputs}\}$.
   (a) Describe the language $\overline{\mathsf{TOTAL}}$ in a way similar to the language TOTAL.
   (b) Show that the languages TOTAL as well as $\overline{\mathsf{TOTAL}}$ are not semi-decidable.

Solution:

(a) $\overline{\mathsf{TOTAL}} = \{M \mid \exists y \in \Sigma^*, M \text{ does not halt on } y\}$

(b) (a) We will show that if TOTAL is decidable, then $\overline{\mathsf{HP}}$ is decidable i.e $\overline{HP} \leq_m \mathsf{TOTAL}$

The reduction algorithm, on input $\langle M, x\rangle$ an instance of $\overline{\mathsf{HP}}$ outputs, M' an instance of TOTAL

$M' =$

"

On input y,

  i. Run M on x
  ii. A. If M halts on x in $|y|+1$ steps, loop
      B. Else halt and accept

"

To prove : $\langle M, x\rangle \in \overline{\mathsf{HP}} \Leftrightarrow M' \in \mathsf{TOTAL}$

$\langle M, x\rangle \in \overline{\mathsf{HP}}$

$\Rightarrow M$ never halts on x

$\Rightarrow \forall y \in \Sigma^*$, M will not halt on x in $|y|+1$ steps

$\Rightarrow M'$ will accept all y

$\Rightarrow M'$ halts on all inputs

$\Rightarrow M' \in \mathsf{TOTAL}$

$\langle M, x\rangle \notin \overline{\mathsf{HP}}$

$\Rightarrow M$ halts on x in less than $c+1$ steps for some $c \geq 0$

$\Rightarrow \forall y \in \Sigma^*, |y| < c+1$, M' will loop on y (the reason for $|y|+1$ in the algorithm, is because, then it guarantees, that M' will loop on atleast the string ϵ, so that it satisfies the condition of $\overline{\mathsf{TOTAL}}$)

$\Rightarrow M' \notin \mathsf{TOTAL}$

Hence $\overline{HP} \leq_m \mathsf{TOTAL}$, and hence TOTAL is not semi-decidable

(b) We will show $\mathsf{HP} \leq_m \mathsf{TOTAL}$ which will imply $\overline{\mathsf{HP}} \leq_m \overline{\mathsf{TOTAL}}$

The reduction machine on input, $\langle M, x\rangle$, an instance of HP, outputs M', an instance of TOTAL

$M' =$

"

On input y

  i. Run M on x
  ii. A. If M halts on x, accept

”

To prove : $\langle M, x\rangle \in \mathsf{HP} \Leftrightarrow M' \in \mathsf{TOTAL}$

$\langle M, x\rangle \in \mathsf{HP}$

$\Rightarrow M$ halts on x

$\Rightarrow M'$ halts and accepts every input

$\Rightarrow M' \in \mathsf{TOTAL}$

$\langle M, x\rangle \notin \mathsf{HP}$

$\Rightarrow M$ loops on x

$\Rightarrow M'$ loops on every input

$\Rightarrow M' \notin \mathsf{TOTAL}$

Hence proved that $\mathsf{HP} \leq_m \mathsf{TOTAL}$ which implies

$\overline{HP} \leq_m \overline{\mathsf{TOTAL}}$