Assignment 6

Nishan Dhoj Karki

10/09/2022

1 5.4) If $A \leq_m B$ and B is a regular language, does that imply that A is a regular language? Why or why not?

Proof sketch:

From **Definition 5.20**, we have
Language A is mapping reducible to language B, written A $\leq_m$ B, if there is a computable function f: $\Sigma^* \rightarrow \Sigma^*$, where for every w,

$$w \in A \Leftrightarrow f(w) \in B$$

.

The function f is called the reduction from A to B.

Here, we are given two languages A and B where B is regular and $A \leq_m B$.
To test if the implication that A is regular or not. Let us define language A as a context-free language. Then we will show that A can be reduced to B, which does not imply A is regular.

Proof:

Let us consider languages $A = \{0^n 1^n \mid n \geq 0\}$ as it is a context free language and $B = \{10\}$ as it is a regular language over alphabet $\Sigma = \{0, 1\}$.

THEOREM 4.9 Every context-free language is decidable.
Here, A is context-free language and is decidable.

Since A is a context-free language and B is a regular language. Both A and B are decidable. Hence, there exists a computable function, $f : \Sigma^* \rightarrow \Sigma^*$ as,

$$f(w) = \begin{cases} 10 & if w \in A, \\ 01 & if w \notin A. \end{cases}$$

Here, for every $w \in A$, $f(w) \in B$ and for every $w \notin A$, $f(w) \notin B$. The function f defines a reduction from A to B.

For the above languages A and B, we have shown that $A \leq_m B$.
B is regular language since it is finite, and A is a context-free language. Here, we have reduced a not regular but context-free language A to a regular language B. This shows that though B is regular and $A \leq_m B$, it does not imply A is regular.

2 5.30) Use Rice's theorem, which appears in Problem 5.28, to prove the undecidability of each of the following languages. However, do not use Rice's theorem. Instead, use a reduction from another undecidable language.
a. $INFINITE_{TM} = \{<M>|M$ **is a TM and** $L(M)$ **is an infinite language**$\}$.

Proof sketch:

We have from **THEOREM 5.1**, $HALT_{TM}$ is undecidable.
Also, From **COROLLARY 5.23**, If $A \leq_m B$ and A is undecidable, then B is undecidable.

We approach this proof by showing a reduction of $HALT_{TM}$ to $INFINITE_{TM}$ so as to conclude if $HALT_{TM}$ is undecidable so is $INFINITE_{TM}$. Let us construct a machine that shows a computable function of a reduction from $HALT_{TM}$ to $INFINITE_{TM}$.

Proof by Reduction from $HALT_{TM}$

Let us consider a mapping reduction from $HALT_{TM}$ to $INFINITE_{TM}$. Let f be the computable function defined as follows and F computes the reduction f:

F = " On input $\langle M, w\rangle$, where M is a TM, w is a string:
  Construct machine A on input w:
  A = "on input x:
    Run M on w for k steps where $k = \mid x \mid$
    if M halts on w, *accept*
    if M doesn't halt on w, *reject*"
  Output $\langle A\rangle$"

The computable function defined shows that $HALT_{TM} \leq_m INFINITE_{TM}$. We have from **THEOREM 5.1**, $HALT_{TM}$ is undecidable. Also, From **COROLLARY 5.23**, If $A \leq_m B$ and A is undecidable, then B is undecidable. By **COROLLARY 5.23**, $INFINITE_{TM}$ is undecidable since $HALT_{TM}$ is undecidable and $HALT_{TM} \leq_m INFINITE_{TM}$ is valid.

3 5.30) Use Rice's theorem, which appears in Problem 5.28, to prove the undecidability of each of the following languages. c. $ALL_{TM} = \{\langle M\rangle | M$ is a TM and $L(M) = \sum^*\}$. Do problem 5.30 (c) from your textbook. You should use Rice's theorem for this proof. You may follow the structure of the book's sample solution for 5.30 (a), but make sure you show each sub-part carefully.

ALL_{TM} is a language of TM descriptions. It satisfies the two conditions of Rice's Theorem.
First, It is nontrivial because some TMs accept all possible strings of an alphabet Σ and others do not. Let, T_1 and T_2 be two Turing Machines such that T_1 accepts all inputs and T_2 rejects all input. Here, $\langle T_1\rangle \in ALL_{TM}$ and $\langle T_2\rangle \notin ALL_{TM}$.
Secondly, if two TMs M_1 and M_2 recognize the same language then either both have descriptions in ALL_{TM} or neither do. If $\langle M_1\rangle$ and $\langle M_2\rangle$ both accepts all inputs then $L(M_1) = L(M_2)$. Either both $\langle M_1\rangle$ and $\langle M_2\rangle \in ALL_{TM}$ or both $\langle M_1\rangle$ and $\langle M_2\rangle \notin ALL_{TM}$.
Since all the conditions of Rice's theorem are satisfied, it implies that ALL_{TM} is undecidable.

4 Is the language from 5.30 (a) co-Turing-recognizable? Prove your answer. Try using Corollary 5.29. This should be easy if you used a mapping reduction in your earlier proof, or if you can show that your earlier reduction is a mapping reduction.

Proof: The language $INFINITE_{TM}$ is not co-Turing recognizable.

THEOREM 4.22 A language is decidable iff it is Turing-recognizable and co-Turing-recognizable.
COROLLARY 5.29 If $A \leq_m B$ and A is not Turing-recognizable, then B is not Turing-recognizable.

From problem 2, we have shown that $HALT_{TM} \leq_m INFINITE_{TM}$. From **THEOREM 5.1**, $HALT_{TM}$ is undecidable. Since, $HALT_{TM}$ is undecidable, from **THEOREM 4.22** $HALT_{TM}$ is not co-Turing recognizable. Since, $HALT_{TM}$ is not co-Turing recognizable then $\overline{HALT_{TM}}$ is not Turing recognizable. From definition of mapping reducibility $HALT_{TM} \leq_m INFINITE_{TM}$ implies to $\overline{HALT_{TM}} \leq_m \overline{INFINITE_{TM}}$. Using **COROLLARY 5.29**, we can say that $\overline{INFINITE_{TM}}$ is not Turing recognizable as $\overline{HALT_{TM}}$ is not Turing recognizable. Since, $\overline{INFINITE_{TM}}$ is not Turing recognizable, by definition, we can say that $INFINITE_{TM}$ is not co-Turing recognizable.

5 Extra credit (10 points): Is the language from 5.30 (a) Turing-recognizable? Prove your answer.

Following the description of **COROLLARY 5.29**, we can show a reduction from $\overline{HALT_{TM}}$ to $INFINITE_{TM}$ to prove that $INFINITE_{TM}$ is not Turing-recognizable. Let us show that a reduction from $\overline{HALT_{TM}}$ to $INFINITE_{TM}$. The reduction function f works as follows:

F = "on input $\langle M, w\rangle$, where M is a TM and w is a string:

Construct the following machine B
$B =$ "on input x:
Run M on w for k steps where $k = \mid x \mid$
if M does not halt, accept
if M halts, reject"
Print $\langle B \rangle$ "

The above construction of a machine F shows a computable function f for a mapping reduction of $\overline{HALT_{TM}}$ to $INFINITE_{TM}$.

We know that $HALT_{TM}$ is not co-Turing-recognizable. Since, $HALT_{TM}$ is not co-Turing-recognizable, by definition $\overline{HALT_{TM}}$ is not Turing recognizable. Hence, from corollary 5.29, $INFINITE_{TM}$ is not Turing recognizable.