CS3331 – Assignment 4
due Dec. 3, 2019, (latest to submit: Dec. 6, 11:55pm)

1. (80pt) For each of the following languages, prove, without using Rice's Theorem, whether it is (i) in D, (ii) in SD but not in D, or (iii) not in SD.

   1 $L_1 = \{<M> | \{\varepsilon, \mathtt{ab}, \mathtt{abab}\} \subseteq L(M)\}$

   Solution. SD − D.

   SD: Construct M' to semidecide L_1:

   1. run M on ε
   3. If it accepts, then run M on ab
   5. If it accepts, then run M on abab
   7. If it accepts, then accept

   ¬D: Reduction from H:

   $R(<M, w>) =$

   1. construct $M_\#$:
      1.1 erase tape
      1.2 write w on tape
      1.3 run M on w
      1.4 accept
   2. return $<M_\#>$

   Assume *Oracle* for L_1.

   $<M, w> \in H$: M halts on w, so $M_\#$ accepts everything, in particular $\{\varepsilon, \mathtt{ab}, \mathtt{abab}\}$. *Oracle* accepts.

   $<M, w> \notin H$: M does not halts on w, so $M_\#$ accepts nothing. *Oracle* rejects.

   2 $L_2 = \{<M> | \{L(M) \cap (\mathtt{ab})^*$ is infinite$\}$

   Solution. ¬SD. Reduction from ¬H:

   $R(<M, w>) =$

   1. construct $M_\#$:
      1.1 save its input x
      1.2 erase tape
      1.3 write w on tape
      1.4 run M on w for $|x|$ steps or until it halts
      1.5 if M halted, then loop
      1.6 else accept
   2. return $<M_\#>$

   Assume *Oracle* that semidecides L_2.

   $< M, w > \in \neg H$: M does not halt on w, so $M_\#$ always makes it to 1.6 and accepts everything, in particular, infinitely many strings from $(\mathtt{ab})^*$. *Oracle* accepts.

   $<M, w> \notin \neg H$: M halts on w in, say, n steps, so $M_\#$ accepts only the strings x with $|x| < n$, which is a finite set. *Oracle* does not accept.

3 $L_3 = \{<M>| \; \{L(M) \cap (\texttt{ab})^* \text{ is finite}\}$

Solution. $\neg$SD. Reduction from $\neg H$:

$R(<M, w>) =$

1. construct $M_\#$:
   1.1 save its input x
   1.2 erase tape
   1.3 write w on tape
   1.4 run M on w
   1.6 accept
2. return $<M_\#>$

Assume *Oracle* that semidecides L_3.

$<M, w> \in \neg H$: M does not halt on w, so $M_\#$ never makes it to 1.6 and accepts nothing. In particular, $L(M) \cap (\texttt{ab})^*$ is finite. *Oracle* accepts.

$<M, w> \notin \neg H$: M halts on w, so $M_\#$ makes it to 1.6 and accepts everything. $L(M) \cap (\texttt{ab})^*$ is infinite. *Oracle* does not accept.

4 $L_4 = \{<M>| \; \{L(M) \cap (\texttt{ab})^* = \emptyset\}$

Solution. $\neg$SD. The same reduction as for L_3.

5 $L_5 = \{<M>| \; \{L(M) \cap (\texttt{ab})^* \neq \emptyset\}$

Solution. SD $-$ D

SD: Construct M' to semidecide L_5:

1. run M on all strings of $(\texttt{ab})^*$ in dovetailing mode
2. if M accepts a string, then accept

$\neg$D: Reduction from H:

$R(<M, w>) =$

1. construct $M_\#$:
   1.1 erase tape
   1.2 write w on tape
   1.3 run M on w
   1.4 accept
2. return $<M_\#>$

Assume *Oracle* for L_5.

$<M, w> \in H$: M halts on w, so $M_\#$ accepts everything, in particular, $L(M) \cap (\texttt{ab})^* \neq \emptyset$. *Oracle* accepts.

$<M, w> \notin H$: M does not halts on w, so $M_\#$ accepts nothing, in particular, $L(M) \cap (\texttt{ab})^* = \emptyset$. *Oracle* rejects.

6 $L_6 = \{<M>| \; L(M) \neq L(M')$ for any other TM $M'\}$

Solution. D. $L_6 = \emptyset$ since any language semidecided by a Turing machine is semidecided by infinitely many Turing machines.

7 $L_7 = \{<M> \mid \neg L(M) \in D\}$.

Solution. $\neg$SD. Reduction from $\neg H$.

$R(<M, w>) =$

1. construct $M_{\#}$:
   - 1.1 on input x, save x for later
   - 1.2 erase tape
   - 1.3 write w on tape
   - 1.4 run M on w
   - 1.5 put x back on the tape
   - 1.6 if x is not a correct encoding $<M', w'>$, then reject
   - 1.7 run M' on w'
   - 1.8 accept
2. return $<M_{\#}>$

Assume *Oracle* that semidecides L_7.

$<M, w> \in \neg H$: M does not halt on w, so M never makes it to 1.5 and hence $M_{\#}$ accepts $\emptyset$. Since $\neg\emptyset = \Sigma^* \in$ D, *Oracle* accepts.

$<M, w> \notin \neg H$: M halts on w, so $M_{\#}$ accepts precisely encodings $<M', w'>$ such that M' halts on w', which means $L(M_{\#}) = H$. Since $\neg H \notin$ D, *Oracle* does not accept.

8 $L_8 = \{<M> \mid L(M) \in \text{SD}\}$.

Solution. D. $L(M) \in$ SD holds true for any M, so L_8 consists of all correct encodings of Turing Machines, which is in D.

2. (20pt) For each of the languages in question 1, indicate whether Rice's Theorem can be used or not to prove that the corresponding language is not in D. Explain why.

   Solution. All but L_6 and L_8 where the property is trivial.

Note Submit your solution as a pdf file on `owl.uwo.ca`.