Solutions of Assignment 10

> Theory of Computation, Fall 2022

Keys

Q1

Proof:

For every recursively enumerable language R, there exists a Turing machine M_R that semidecides it.

Then there is a recursive function $f(w) = "M_R""w"$ such that for any $x \in \Sigma^*$, $x \in R \iff f(x) \in H$.

Therefore, $f(w) = "M_R""w"$ is a reduction from R to H. Thus every recursively enumerable language can be reduced to H.

Q2

Proof:

$A \leq B$ implies that there is a reduction $f : \Sigma^* \to \Sigma^*$ from A to B such that for any $x \in \Sigma^*$, $x \in A \iff f(x) \in B$.

That is, for any $x \in \Sigma^*$, $x \in \overline{A} \iff f(x) \in \overline{B}$.

Therefore, f is also a reduction from $\overline{A}$ to $\overline{B}$. Thus $\overline{A} \leq \overline{B}$.

Q3

According to the conclusion of **Q2**, there is $A \leq \overline{A} \implies \overline{A} \leq A$.

According to the conclusion of **Q5 (a)** in ***Assignment 9*** (If $A \leq B$ and that B is recursively enumerable, then A is also recursively enumerable), there is
A is recursively enumerable $\implies$ $\overline{A}$ is recursively enumerable.

Since A and $\overline{A}$ are recursively enumerable, A is recursive.

> **Note:** To show that A is recursive if A and $\overline{A}$ are recursively enumerable, we can construct 2 Turing machines M_A and $M_{\overline{A}}$ that semidecide A and $\overline{A}$ respectively. Then we construct a Turing machine M as follows:
>
> $$\begin{array}{rll} M = & & \text{on input } w: \\ & 0. & \text{For } i = 1, 2, \ldots \text{ loop} \\ & 1. & \quad \text{Run } M_A \text{ on } w \text{ for } i \text{ steps} \\ & 2. & \quad \text{If } M_A \text{ halts on } w \text{ in } i \text{ steps} \\ & 3. & \qquad \text{Accept } w \\ & 4. & \quad \text{Run } M_{\overline{A}} \text{ on } w \text{ for } i \text{ steps} \\ & 5. & \quad \text{If } M_{\overline{A}} \text{ halts on } w \text{ in } i \text{ steps} \\ & 6. & \qquad \text{Reject } w \end{array} \tag{25}$$
>
> On an input w, either M_A or $M_{\overline{A}}$ halts, hence M accepts w if M_A halts on w, i.e. $w \in A$, and rejects w if $M_{\overline{A}}$ halts on w, i.e. $w \in \overline{A}$. Therefore, M decides A. Thus A is recursive.

Q4

No.

Consider 2 languages $A = \{a^n b^n : n \geq 0\}$ and $B = \{a\}$ on $\Sigma = \{a, b\}$. Obviously A is not regular while B is. Besides, they are both recursive, so there exists a Turing machine M_A that decides A.

We can construct a function $f : \Sigma^* \to \Sigma^*$ as follows:

$$f(w) = \begin{cases} a, & w \in A \\ b, & w \in \overline{A} \end{cases} \tag{26}$$

To show that f is recursive, we can construct a Turing machine M_f that computes f as follows:

$$
\begin{array}{rrl}
M = & & \text{on input } w: \\
& 1. & \text{Run } M_A \text{ on } w \\
& 2. & \text{If } M_A \text{ accepts } w \\
& 3. & \quad \text{Output } a \\
& 4. & \text{Else } (M_A \text{ rejects } w) \\
& 5. & \quad \text{Output } b
\end{array} \tag{27}
$$

Now we have that f is recursive, and $w \in A \iff f(w) \in B$, so f is a reduction from A to B, hence $A \leq B$.

This implies that A is not necessarily regular even if $A \leq B$ and B is regular.

> **Note: We should keep an eye on whether f is recursive when constructing a reduction.**

Q5

$\overline{A}$ is not recursively enumerable.

According to the conclusion of **Q2**, there is $H \leq A \implies \overline{H} \leq \overline{A}$.

We have proved that $\overline{H}$ is not recursively enumerable. Thus $\overline{A}$ is not recursively enumerable either.

> **Note1:** We can find that if a language A is recursively enumerable but not recursive, then $\overline{A}$ is not recursively enumerable.

> **Note2:** The last step is based on the fact that if $A \leq B$ and A is not recursively enumerable, then B is not recursively enumerable either. You can refer to the solution of **Q5 (b)** of ***Assignment 9*** to write the proof of this statement.

Q6

Proof:

Given that

$$
\begin{aligned}
A &= \{"M" : M \text{ is a Turing machine that halts on some input}\} \\
L &= \{"M_1""M_2""k" : M_1 \text{ and } M_2 \text{ are two Turing machines with } |L(M_1) \cap L(M_2)| \geq k\}
\end{aligned} \tag{28}
$$

We have proved that A is not recursive, so we can construct a reduction from A to L to show that L is not recursive.

Firstly, construct a Turing machine M_{all} as follows.

$$M_{all} = \begin{array}{ll} & \text{on input } w: \\ 1. & \text{Accept } w \end{array} \tag{29}$$

We get that $L(M_{all}) = \Sigma^*$, and $L(M) \cap L(M_{all}) = L(M)$.

Then we construct a recursive function f as follows.

$$f(''M'') = ''M''''M_{all}''''1'' \tag{30}$$

We will show that it is a reduction from A to L.

> **Note1:** Here $f(''M'') = ''M''''M''''1''$ is also a correct reduction from A to L.

For any Turing machine M,

$$\begin{gathered} ''M'' \in A \iff M \text{ halts on some input } \iff \\ |L(M)| = |L(M) \cap L(M_0)| \geq 1 \iff f(x) \in L \end{gathered} \tag{31}$$

Therefore, $f(''M'') = ''M''''M_{all}''''1''$ is a reduction from A to L. Thus L is not recursive.

> **Note2:** The proof below is incorrect.
>
> > $$\begin{gathered} D = \{''M_1''''M_2'' : M_1 \text{ and } M_2 \text{ are 2 Turing machines with } L(M_1) = L(M_2)\} \\ f(''M_1''''M_2'') = ''M_1''''M_2'''' \max\{|L(M_1)|, |L(M_2)|\}'' \end{gathered} \tag{32}$$
> >
> > D is not recursive, and f is a reduction from D to L.
>
> Here f is not a recursive function since $|L(M_1)|$ and $|L(M_2)|$ are not computable and neither is $\max\{|L(M_1)|, |L(M_2)|\}$.
>
> A reduction must be a recursive function first, so f is **NOT** a reduction from D to L.

Grading

100 pts. in total

- Q1. 15 pts.
- Q2. 15 pts.
- Q3. 15 pts.
- Q4. 20 pts.
- Q5. 15 pts.
- Q6. 20 pts.
