Homework 3

Yutong Huang (yxh589)

Problem 1

Proof. Proof by contradiction.
Assume L_{343} is decidable $\implies$ There exists an algotithm that decides L_{343}.
Then assume algorithm $DecideL_{343}$ correctly decides L_{343}, and contruct the following algorithm:

Algorithm 1: $DecideHalt$

```
Input: M, w
Encode the following Turing machine:
TuringMachine M'(x)
    Run M on w
    if x=EECS343, or x=is, or x=fun then
        return true
    else
        return false
if DecideL343(M') then
    return true
else
    return false
```

Case 1: M halts on input w
$\implies$ M' halts and accept any string from $\{EECS343,\ is,\ fun\}$ and no other strings.
$\implies$ $DecideL_{343}$ returns true $\rightarrow$ $DecideL_{343}$ accepts.
$\implies$ $DecideHalt$ returns true $\rightarrow$ $DecideHalt$ accepts.
Case 2: M does not halt on input w
$\implies$ M' does not halt on any input
$\implies$ M' does not accept any string from $\{EECS343,\ is,\ fun\}$ and no other strings.
$\implies$ $DecideL_{343}$ returns false $\rightarrow$ $DecideL_{343}$ rejects.
$\implies$ $DecideHalt$ returns false $\rightarrow$ $DecideHalt$ rejects.

Therefore, the algorithm $DecideHalt$ correctly decides if M halts on input w, which is impossible.

Proof. Assume there is an algorithm $Halt(P, I)$ that correctly decides if an algorithm P halts on input I, and construct the following algorithm

Algorithm 2: Z

```
Input: x
if Halt(x, x) then
    loop forever
else
    return
```

Consider $Z(Z)$
Case 1: If Z halts on Z, then Z loops forever $\implies$ contradiction.
Case 2: If Z does not halt on Z, then Z returns $\implies$ contradiction.
Therefore, there is no algorithm that correctly decides if a program will halt on given input. ☐

By contradiction, then, $DecideL_{343}$ cannot exist $\implies$ L_{343} is undecidable. ☐

Problem 2

$L = \{\langle M, q\rangle | q \ is \ a \ useless \ state \ in \ TM \ M\}$
Prove L is undecidable.

Proof. Proof by contradiction, assume L is decidable and a machine $DecideL$ decides it. We know that for any Turing machine M, for which $L(M) = \phi$, q_{accept} is a useless state.
Reduce L to $E_{TM} = \{\langle M\rangle | M \ is \ a \ TM \ and \ L(M) = \phi\}$, and consider the following algorithm:

Algorithm 3: $DecideE_{TM}$

Input: M
Run $DecideL(\langle M, q_{accept}\rangle)$;
if $DecideL$ *accepts* **then**
    return true;
else
    return false;

Then we have an algorithm $DecideE_{TM}$ that decides E_{TM}, which is impossible according in Theorem 5.2 in the book.

Therefore, by contradiction, L is undecidable.

□

Problem 3

Consider $B = \{w | w = 0x \ for \ some \ x \in A_{TM}, or \ w = 1y \ for \ somey \in \overline{A_{TM}}\}$.

Then $\overline{B} = \{w | w = \epsilon, or \ w = 0x \ for \ some \ x \in \overline{A_{TM}}, or \ w = 1y \ for \ some \ y \in A_{TM}\}$

Mapping reducibility from B to $\overline{B}$:

$$f(w) = \begin{cases} 1x & (w = 0x) \\ 0x & (w = 1x) \end{cases}$$

Then $B \leq_m \overline{B}$.

Proof. $\forall w \in B \Rightarrow f(w) \in \overline{B}$:
Case 1: $w = 0x$, then $f(w) = 1x$, and $x \in A_{TM} \implies f(w) \in \overline{B}$
Case 2: $w = 1x$, then $f(w) = 0x$, and $x \in \overline{A_{TM}} \implies f(w) \in \overline{B}$
$\forall f(w) \in \overline{B} \Rightarrow w \in B$:
Case 1: $f(w) = 1x$, then $x \in A_{TM}$ and $w = 0x \implies w \in B$
Case 1: $f(w) = 0x$, then $x \in \overline{A_{TM}}$ and $w = 1x \implies w \in B$ □

B is undecidable.

Proof. It's obvious that $A_{TM} \leq_m \overline{B}$, and A_{TM} is unrecognizable, therefore $\overline{B}$ is unrecognizable.
Since $B \leq_m \overline{B}$, we can also deduce that B is unrecognizable, which entails that B is undecidable. □