Problem 1

Recall the language:
\[ K = \{ \langle M \rangle \mid M \text{ accepts } \langle M \rangle \} \]

Prove that \( \text{HALT} \leq_m K \).

**Definition**

\( A \leq_m B \) (there is a reduction from \( A \) to \( B \)) if there is a computable function \( f : \Sigma^* \to \Sigma^* \) such that for every \( w \in \Sigma^* \), \( w \in A \iff f(w) \in B \).

**Proof idea**

We characterize each set in the following table.

<table>
<thead>
<tr>
<th></th>
<th>( \text{HALT}, \langle M, w \rangle )</th>
<th>( K, \langle M \rangle )</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>( M \text{ accepts or rejects on input } w )</td>
<td>( M \text{ accepts on input } \langle M \rangle )</td>
</tr>
<tr>
<td>OUT</td>
<td>( M \text{ loops on input } w )</td>
<td>( M \text{ rejects or loops on input } \langle M \rangle )</td>
</tr>
</tbody>
</table>

We define a machine \( M' \) which accepts on every input if and only if \( M \) halts on \( w \), for a given \( M \) and \( w \). We define \( f(\langle M, w \rangle) = M' \). Since the construction of \( M' \) can be performed by a Turing machine, and \( M' \) accepts everything (including its own code \( \langle M' \rangle \)) if and only if \( M \) accepts \( w \), \( \langle M, w \rangle \in \text{HALT} \iff f(\langle M, w \rangle) \in K \).

**Proof.** The function \( f(\langle M, x \rangle) = M' \) defined as follows:

\( M' : " \text{on input } \langle x \rangle \)

i. Run \( M \) on \( w \).

(a) If \( M \) accepts, ACCEPT.
(b) If \( M \) rejects, ACCEPT."

\( f \) is computable. Since the set of Turing machines is computable, there is an ordered enumeration \( E = M_1, M_2, M_3, \ldots \) of these machines. Given \( E \), the function \( g(\langle M \rangle) = i \) is computable, where \( i \) is the index of \( M \) in \( E \). Since \( f(\langle M, w \rangle) = M_g(\langle M' \rangle) \) and \( g(\langle M' \rangle) \) is computable, \( f \) is computable.

Suppose \( \langle M, w \rangle \in \text{HALT} \). Then \( M \) halts and either accepts or rejects on input \( w \). In either case, \( M' \) accepts on every input. So, if \( M \) accepts on \( w \), then \( M' \) accepts on every input and so accepts \( \langle M \rangle \). And, if \( M \) rejects on \( w \), then \( M' \) accepts on every input and so accepts \( \langle M \rangle \). Therefore, if \( \langle M, w \rangle \in \text{HALT} \) then \( f(\langle M, x \rangle) \in K \).

Suppose \( \langle M, w \rangle \not\in \text{HALT} \). Then \( M \) does not halt on \( w \), and \( M' \) halts on, and so accepts, no input. So, if \( M \) does not halt on \( w \), then \( M' \) does not accept anything, and so does not accept \( \langle M \rangle \). Therefore, if \( \langle M, w \rangle \not\in \text{HALT} \) then \( f(\langle M, x \rangle) \not\in K \) □
Problem 2

Prove that $K \leq_m \text{HALT}$.

Proof idea

We want to show that for some computable function $f$, for any $\langle M \rangle \in \Sigma^*$,

$$\langle M \rangle \in K \iff f(\langle M \rangle) \in \text{HALT}$$

The sets are characterized in the table in Problem 1.

We define a machine $M'$ which halts on an input $x$ iff $M$ accepts on $\langle M \rangle$. By the same reasoning as in Problem 1, this proves that $K \leq_m \text{HALT}$

Proof. The function $f(\langle M \rangle) = \langle M', \langle 1 \rangle \rangle$, with $M'$ defined as follows:

$M' :$ "on input $x$

1. Run $M$ on $\langle M \rangle$.

   (a) If $M$ accepts, ACCEPT.

   (b) If $M$ rejects, LOOP."

By the same reasoning as in Problem 1, $f(\langle M \rangle)$ is computable.

Suppose $\langle M \rangle \in K$. Then $M$ accepts (and halts) on input $\langle M \rangle$, so $M'$ accepts (and halts) on all inputs, and, so, halts on $\langle 1 \rangle$. Therefore, $f(\langle M \rangle) \in \text{HALT}$.

Suppose $\langle M \rangle \notin K$. Then $M$ either rejects or loops on input $\langle M \rangle$, so $M'$ loops on all inputs and, so, loops on $\langle 1 \rangle$. Therefore, $f(\langle M \rangle) \notin \text{HALT}$.

Therefore, $\langle M \rangle \in K \iff f(\langle M \rangle) \in \text{HALT}$. \qed
**Problem 3**

Consider the language:

\[ Q = \{ \langle M, w, q \rangle | M \text{ enters state } q \text{ at least once during its computation on } w \} \]

Prove that HALT \( \leq_m Q \).

**Proof idea**

We want to show that for some computable function \( f \), for any \( \langle M, w \rangle \in \Sigma^* \),

\[ (M, w) \in HALT \iff f(\langle M, w \rangle) \in Q \]

We characterize each set in the following table.

<table>
<thead>
<tr>
<th></th>
<th>HALT, ( \langle M, w \rangle )</th>
<th>Q, ( \langle M, w, q \rangle )</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>( \langle M, w \rangle )</td>
<td>( \langle M, w, q \rangle )</td>
</tr>
<tr>
<td></td>
<td>( M ) accepts or rejects on</td>
<td>( M ) enters state ( q ) at</td>
</tr>
<tr>
<td></td>
<td>( w )</td>
<td>least once in its computation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>on ( w )</td>
</tr>
<tr>
<td>OUT</td>
<td>( M ) loops on ( w )</td>
<td>( M ) never enters state ( q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>in its computation on ( w )</td>
</tr>
</tbody>
</table>

We define a machine which enters state \( q \) in its computation on \( w \) iff \( M \) halts on \( w \). By the same reasoning in Problem 1, this shows that HALT \( \leq_m Q \).

**Proof.** The function \( f(\langle M, w \rangle) = \langle M', (1), q_0 \rangle \), with \( M' \) defined as follows:

\( M' : \) "on input \( x \)

i. Run \( M \) on \( w \).

(a) If \( M \) accepts or rejects,

i. Transition to state \( q_0 \)
ii. ACCEPT.

Where \( q_0 \) is the final halting and accepting state in any computation of \( M' \).

\( f \) is computable. Since the set of Turing machines is computable, there is an ordered enumeration \( E = M_1, M_2, M_3, \ldots \) of these machines. Claim: for some \( M_i \) in \( E \), \( M_i = M' \). We know this because if we take a TM \( M'' \) which is equivalent to \( M' \), but in which the final state is not \( q_0 \), we can construct \( M' \) from \( M'' \) by changing the final accepting state of \( M'' \) to \( q_0 \) (if \( q_0 \) is already a state of \( M'' \), then we swap it with the final accepting state). So, by the same reasoning as in Problem 1, \( f \) is computable.

Suppose \( \langle M, w \rangle \in HALT \). Then \( M \) halts (either accepting or rejecting) on input \( w \) and, so, \( M' \) accepts all inputs (including \( (1) \)) and \( q_0 \) in the the computation of those inputs. Therefore, \( f(\langle M, w \rangle) = \langle M', (1), q_0 \rangle \in Q \).

Suppose \( \langle M, w \rangle \notin HALT \). Then \( M \) does not halt on input \( w \) and, so, \( M' \) loops on all inputs (including \( (1) \)) and never enters \( q_0 \) in the the computation of those inputs. Therefore, \( f(\langle M, w \rangle) \notin Q \).

Therefore, HALT \( \leq_m Q \). \( \square \)
Problem 4
Consider the language:
\[
2P = \{ \langle M \rangle | M \text{ accepts at least 2 palindromes, } \Sigma = \{0, 1\} \}
\]
Prove that $\text{HALT} \leq_m 2P$.

Proof idea
We want to show that for some computable function $f$, for any $\langle M, w \rangle \in \Sigma^*$,
\[
\langle M, w \rangle \in \text{HALT} \iff f(\langle M, w \rangle) \in 2P
\]
We characterize each set in the following table.

<table>
<thead>
<tr>
<th>$\text{HALT, } \langle M, w \rangle$</th>
<th>$\text{2P, } \langle M \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{IN } w$</td>
<td>$\text{M accepts or rejects on } w$</td>
</tr>
<tr>
<td>$\text{OUT } w$</td>
<td>$\text{M loops on } w$</td>
</tr>
</tbody>
</table>

We define a machine which accepts two palindromes iff $M$ halts on $w$. By the same reasoning
in Problem 1, this shows that $\text{HALT} \leq_m 2P$.

Proof. The function $f(\langle M, w \rangle) = \langle M' \rangle$, with $M'$ defined as follows:

$M'$ : "on input $x$

i. Run $M$ on $w$.

(a) If $M$ accepts, ACCEPT.
(b) If $M$ rejects, ACCEPT.

By the same reasoning in Problem 1, $f$ is computable.
Suppose $\langle M, w \rangle \in \text{HALT}$. Then $M$ halts, either accepting or rejecting, on input $w$. So $M'$
accepts on all inputs, including two palindromes. So $f(\langle M, w \rangle) \in 2P$.
Suppose $\langle M, w \rangle \notin \text{HALT}$. Then $M$ does not halt on input $w$, so $M'$ loops on all inputs and
accepts nothing. So $M'$ does not accept at least two palindromes. Therefore, $f(\langle M, w \rangle) \notin 2P$.

Therefore, $\text{HALT} \leq_m 2P$. \qed
Problem 5
Prove that $2P \leq_m \text{HALT}$.

Proof idea
We want to show that for some computable function $f$, for any $\langle M \rangle \in \Sigma^*$,

$$\langle M \rangle \in 2P \iff f(\langle M \rangle) \in \text{HALT}$$

The sets are characterized in the table above.

We define a machine which halts on an input iff $M$ accepts two palindromes. By the same reasoning in Problem 1, this shows that $2P \leq_m \text{HALT}$.

Proof. The function $f(\langle M \rangle) = \langle M', (1) \rangle$, with $M'$ defined as follows:

$M' : \ "on \ input \ x$

i. Run $M$ on serially on each possible input.

(a) If, at any point, $M$ accepts two palindromes, ACCEPT.

$f$ is computable. Using the same algorithm as in Homework 3, we can iterate through every input. Starting with $i_0$, for every $i_x \in \Sigma^*$, and for every $y \leq x$, run $M$ on $i_y$ for $x$ steps. Since $M'$ picks out a Turing machine, by the same reasoning as in Problem 1, $f$ is computable.

Suppose $\langle M \rangle \in 2P$. Then $M$ accepts at least two palindromes, and $M$ halts on all inputs. So $f(\langle M \rangle) \in \text{HALT}$.

Suppose $\langle M \rangle \not\in 2P$. Then $M$ accepts either zero or one palindrome. So, $M'$ loops on all inputs, including $(1)$. Therefore, $f(\langle M \rangle) \not\in \text{HALT}$.

Therefore, $2P \leq_m \text{HALT}$.

$\Box$