
Answer: We will show that if $BB$ is a computable function then

$$H_{TM} = \{ \langle M \rangle \mid M \text{ halts on the empty string} \}$$

is decidable. However as we have seen in the class $H_{TM}$ is not decidable.

Let $M$ be a TM. From $M$ we construct a two-tape Turing Machine $N$. On the first tape, $N$ simulates $M$ on the input $\varepsilon$, and on the second tape, it writes a 1 at every step. So if $M$ terminates after $t$ steps, $1^t$ will be written on the second tape. We can simulate $N$ with a single-tape Turing Machine $T$ so that if $M$ terminates after $t$ steps, then eventually $T$ will terminate with $1^t$ written on the tape.

If $M$ has $k$ states, then we can calculate the number of states of $T$ that we constructed. Let us denote this number by $f(k)$. Now we claim that if $M$ does not halt after $BB(f(k))$ steps on the input $\varepsilon$, then we will know that it will never halt. Indeed if it halts after $t > BB(f(k))$ steps, then the Turing machine $T$ (which has $f(k)$ states) will terminate with more than $BB(f(k))$ number of 1’s on its tape and this contradicts the definition of the busy beaver function.

2. Answering Hilbert’s 10th problem, Matiyasevich showed that the following language is undecidable: $P$ is the set of all $\langle p \rangle$ such that $p$ is a multivariate polynomial with integer coefficients that evaluates to zero for some assignment of positive integers to its variables. For example $\langle x_1^2 + x_2^2 - 5 \rangle \in P$ as it evaluates to 0 if we set $x_1 = 1$ and $x_2 = 2$. On the other hand $\langle x_1^2 - 5 \rangle$ is not in $P$.

(a) (5 Points) Use Matiyasevich’s result to prove that the following language is undecidable: $L$ is the set of all $\langle p \rangle$ such that $p$ is a multivariate polynomial with integer coefficients that can evaluate to a prime number for some assignment of positive integers to its variables.

Answer: Note that $p$ evaluates to zero for some assignment of positive integers to its variables if and only if $2(1 + p^2)$ evaluates to a prime number (the number 2) for some assignment of positive integers to its variables. This gives a mapping reduction from the language $P$ to the language $L$. 
(b) (10 Points) Use Matiyasevich’s result to prove that the following language is undecidable: 
\( L \) is the set of all pairs of multivariate polynomial \((p, q)\), both with integer coefficients, such that for some assignment of integer values to the variables \( p(\cdot) > q(\cdot) \).

**Answer:** Note that \( p \) evaluates to zero for some assignment of positive integers to its variables if and only if \( \langle 1 - p^2, 0 \rangle \in L \). Again we have a mapping reduction from \( P \) to \( L \).

3. (10 Points) Show that the following language is undecidable relative to \( A_{TM} \).

\[ B_{TM} = \{ \langle M^{ATM}, w \rangle \mid M^{ATM} \text{ is an oracle TM that accepts } w \} \]

**Answer:** Suppose it is. Then there is an oracle Turing Machine \( N^{ATM} \) that decides \( B_{TM} \). Let \( M_1^{ATM}, M_2^{ATM}, \ldots \) be an enumeration of all oracle Turing Machines that have access to an oracle for \( A_{TM} \), and let \( w_1, w_2, \ldots \) be an enumeration of all the words in \( \Sigma^* \). Consider the following oracle Turing machine \( P^{ATM} \).

- On input \( w_i \):
  - Simulate \( N^{ATM} \) on \( \langle M_i, w_i \rangle \) using the oracle for \( A_{TM} \).
  - If \( \langle M_i, w_i \rangle \in B_{TM} \), then reject \( w_i \).
  - If \( \langle M_i, w_i \rangle \not\in B_{TM} \), then accept \( w_i \).

Note that \( P^{ATM} \) is an oracle Turing machine with access to an oracle for \( A_{TM} \), and hence must be the same as \( M_j^{ATM} \) for some value of \( j \). But this is not possible as \( P^{ATM} \) accepts \( w_j \) if and only if \( M_j^{ATM} \) rejects \( w_j \).

4. (10 Points) Prove that a language \( L \) is recursively enumerable if and only if it can be expressed as

\[ L = \{ x \mid \exists y \text{ such that } \langle x, y \rangle \in R \} \]

where \( R \) is a decidable language. You need to prove that every language of this form is recursively enumerable, and that every recursively enumerable language can be described as above for some decidable language \( R \).

**Answer:** First note that if \( L \) can be expressed in the above form, then it is r.e. Indeed a TM can enumerate all the possible strings for \( y \), say \( y_1, y_2, \ldots \) and see if \( R \) accepts any of \( \langle x, y_i \rangle \). If it does, it will accept and halt. Now if \( x \in L \), then by the description of \( L \), there must be some \( y_i \) for which \( R \) accepts \( \langle x, y_i \rangle \). On the other hand if \( x \not\in L \), then no \( \langle x, y_i \rangle \) will be accepted and thus the constructed TM will loop.

To prove the other direction, suppose that \( L \) is r.e. and thus recognized by some Turing Machine \( M \). Now let \( R \) be a TM that takes \( \langle x, y \rangle \) and simulates \( M \) on \( x \) for \( |y| \) many steps. If \( M \) accepts \( x \) in those many steps it accepts, and otherwise it rejects. Obviously if \( x \in L \), then \( M \) must accept \( x \) after certain number of steps; so any sufficiently long \( y \) will satisfy \( \langle x, y \rangle \in R \). On the other hand if \( x \not\in L \), \( R \) will not accept \( \langle x, y \rangle \) for any \( y \).

5. (15 Points) You are allowed to use Rice’s theorem (See Problem 5.28 of the textbook) to answer this question. For each one of the following three languages, either prove that they are decidable, or prove that they are undecidable.

\[ L_r = \{ \langle M \rangle \mid L(M) \text{ is a regular language} \} \]

\[ L_{re} = \{ \langle M \rangle \mid L(M) \text{ is a recursively enumerable language} \} \]
\[ L_d = \{ \langle M \rangle \mid L(M) \text{ is a decidable language} \}. \]

**Answer:** \( L_{re} \) is decidable, as by definition \( L(M) \) is a recursively enumerable language for any Turing Machine \( M \). Hence
\[ L_{re} = \{ \langle M \rangle \mid M \text{ is a description of a TM} \}, \]
and this can be verified easily.

The other two languages are undecidable by Rice’s Theorem. To see that \( L_r \) is undecidable, note that obviously there are Turing Machines whose languages are not regular (say \( \{0^n1^n : n\} \)), and also there are Turing machines whose languages are regular (say \( \emptyset \)).

Similarly to see that \( L_d \) is undecidable, note that there are Turing Machines whose languages are not decidable (say \( A_{TM} \)), and also there are Turing machines whose languages are decidable (say \( \emptyset \)).

6. (5 Points) Prove that every recursively enumerable language \( L \) satisfies \( L \leq_m A_{TM} \).

**Answer:** Suppose that \( L \) is recursively enumerable and thus recognized by some Turing Machine \( M \). For every \( w \in L \), let \( f(w) = \langle M, w \rangle \). Note that
\[ w \in L \iff f(w) \in A_{TM}. \]

7. (5 Points) Prove that for any two languages \( A \) and \( B \), there exists a language \( J \) such that \( A \leq_T J \) and \( B \leq_T J \).

**Answer:** Let \( J = \{0\} \cdot A \cup \{1\} \cdot B \), where \( \cdot \) denotes the concatenation. That is \( J \) is the set of all words \( 0w \) where \( w \in A \) together with the set of all words \( 1w \) where \( w \in B \). To show \( A \leq_T J \), we can use the following oracle Turing machine.
- On input \( w \):
  - Use the oracle to see if \( 0w \in A \), and if yes accept else reject.

Similarly one can show that \( B \leq_T J \) by querying \( 1w \).

8. (10 Points) Prove that for any languages \( A \), there exists a language \( J \) such that \( A \leq_T J \) and \( J \not\leq_T A \).

**Answer:** Let
\[ J = \{ \langle M^A, w \rangle \mid M^A \text{ is an oracle TM that accepts } w \}. \]

The proof that \( J \not\leq_T A \) is identical to the diagonalization proof presented in Question 3, replacing \( A_{TM} \) with \( A \). It remains to show that \( A \leq_T J \). Consider the following oracle Turing Machine \( N^A \):
- On input \( w \):
  - Query the oracle to see if \( w \in A \); if yes accept else reject.

Note that \( w \in A \) if and only if \( \langle N^A, w \rangle \in J \). Hence \( A \leq_m J \) which in particular implies \( A \leq_T J \).

9. (5 Points) Show that the Post Correspondence Problem is decidable relative to \( A_{TM} \).

**Answer:** Post Correspondence Problem is recursively enumerable (just keep generating the arrangements and if you see a match, accept). Now the answer follows from Question 6.
10. (15 Points) Determine whether the following language is decidable or undecidable

\[ X = \{ \langle M \rangle \mid \text{On every input } w, M \text{ eventually leaves the start state} \} \].

**Answer:** It is decidable. Let \( M \) be a Turing machine. Consider the following properties of Turing Machines:

- **Property P1:** There exists \( a_1, \ldots, a_k \) in the alphabet, and self-loops on the start state with labels:
  \[(a_1 \to a_2, L), (a_2 \to a_3, L), \ldots, (a_k \to a_1, L)\].

- **Property P2:** There exists \( a_1, \ldots, a_k \) in the alphabet with \( a_k = \sqcup \) or \( a_k = a_j \) for some \( 1 \leq j < k \), and self-loops on the start state with labels:
  \[(\sqcup \to a_1, \ast_1), (a_1 \to a_2, \ast_2), \ldots, (a_{k-1} \to a_k, \ast_k),\]
  where \( \ast_1, \ldots, \ast_k \in \{ L, R \} \).

- **Property P2:** There exists distinct \( a_1, \ldots, a_k \) in the alphabet, and self-loops on the start state with labels:
  \[(\sqcup \to a_1, \ast_1), (a_1 \to a_2, \ast_2), \ldots, (a_{k-1} \to a_k, \ast_k),\]
  such that (strictly) more than half of the \( \ast \)'s are \( R \).

Note that if \( M \) has property P1 then it will loop on the input \( w = a_1 \), and it will never leave the start state. If it has property P2, then it will never terminate on the \( \varepsilon \) input, as for this input, all the symbols that will be written on the tape will be from \( \sqcup, a_1, \ldots, a_k \) and none of these will make the TM leave the start state. Finally if the TM has property P3, then again on the \( \varepsilon \) input it will never leave the start state. This case is slightly more complicated but it is not hard to see that in this case, it will keep moving left and right on the tape, but each time eventually moving further right and writing more on the tape. This will continue forever. To see this, note that if the machine uses only \( r \) cells on the tape in the first \( t \) steps where \( t \) is sufficiently large, then the head has to move many more times to right than to left (the difference going to be more than \( r \) for a large enough \( t \)). This would mean that the tape could not stay in the first \( r \) steps forever.

Note that if a TM does not satisfy P1, P2, or P3 then it must satisfy the following.

- **Property P4:** Either there is an arrow \( \sqcup \to b, \ast \) that leaves the start state, or there exists distinct \( a_1, \ldots, a_k \) in the alphabet, and self-loops on the start state with labels:
  \[(\sqcup \to a_1, \ast_1), (a_1 \to a_2, \ast_2), \ldots, (a_{k-1} \to a_k, \ast_k),\]
  such that for every \( i \leq k \), at least half of the \( \ast_1, \ldots, \ast_j \)'s are \( L \), and moreover there is an arrow \( a_k \to b, \ast \) that leaves the start state.

**Claim 1.** Consider a Turing Machine \( M \) that does not have property P1, and let \( w \) be the shortest string on which \( M \) never leaves the start state. Then \( w = \varepsilon \).

**Proof.** Since \( M \) does not have property P1, the head cannot get stuck in the first memory cell: Every time the head goes to the first cell, after a few steps it will go to the second cell. Suppose \( w \neq \varepsilon \), and let \( w = w_1w_2 \ldots w_l \) be its letters. Consider \( u = w_2 \ldots w_l \), which is the shorter string obtained by dropping the first letter of \( w \). Note that by the above observation \( M \) will loop on \( u \) as well. This contradicts the assumption that \( w \) was the shortest such string. \( \square \)
Claim 2. If a Turing Machine $M$ has property $P_4$, then it will eventually leave the start state on the input $w = \varepsilon$.

Proof. We only sketch the proof. The head moves to the left more often than to the right, and thus it eventually we will end up with the head on $a_k$ which makes the TM leave the start state.

Now we can decide $X$ in the following manner. On input $\langle M \rangle$ check to see if $M$ satisfies any of the properties $P_1$, $P_2$, or $P_3$. If it does reject it, otherwise accept it.

11. (0 Points) Problems 6.16 and 6.17 from the textbook (3rd Edition). First problem asks you to prove the existence of languages $A, B$ such that $A \not\leq_T B$ and $B \not\leq_T A$. The second problem asks you to describe two disjoint recursively enumerable languages $A$ and $B$ such that there is no decidable $C$ with $A \subseteq C$ and $B \cap C = \emptyset$. 