Computability and Complexity

Exercises

Reducibility

Exercise 1 : ***Accepting the reverse.***
Prove that $L = \{< M > \mid M$ is a Turing machine and M accepts $w^R \Leftrightarrow M$ accepts $w\}$ is undecidable.
Answer:
Assume by contradiction that there is a Turing machine D_L that decides L. We construct the following Turing machine:

$S :=$ *On input $< M, w >$:*

1. *Construct the following Turing machine:*

   $A :=$ *On input x:*

   a. *If $x \notin \{ab, ba\}$ then reject*
   b. *If $x = ab$ then accept*
   c. *Run M on w and accept if M accepts w*

2. *Run D_L on $< A >$. If D_L accepts, then accept; otherwise, reject.*

It is easy to see that D_L will accept A if and only if M accepts w. Thus, S decides A_{TM}, which is a contradiction since A_{TM} is undecidable.

□

Exercise 2 : ***Two halts make a whole.***
Prove that $L = \{< M_1, M_2 > \mid M_1$ and M_2 are Turing machines, and $\exists w$ s.t. M_1 and M_2 accept $w\}$ is undecidable.
Answer:
Assume by contradiction that L is decidable, and let D_L be a Turing machine that decides it. We construct the following Turing machine:

$S :=$ *On input $< M, w >$:*

1. *Construct the following Turing machine:*

   $M_w :=$ *On input x:*

   a. *If $x \neq w$, then loop forever*
   b. *Run M on w and accept if M accepts w*

2. *Run D_L on $< M, M_w >$. If D_L accepts, then accept; otherwise, reject.*

It is easy to see that D_L will accept $< M, M_w >$ if and only if M accepts w, since M_w loops on every input that is not w and will accept w if and only if M accepts w. Thus, S decides A_{TM}, a contradiction.

□

Exercise 3 : ***Rice's Theorem is not about eating.***
We say that P is a *nontrivial* property if it is neither true nor false for every computable function.
Prove the following:
Rice's Theorem *If P is a nontrivial property of the language of a Turing machine, then determining whether a given Turing machine's language has property P is undecidable.*
More formally: let P be a language consisting of Turing machine descriptions with the following properties:

1. There exist Turing machines M_1 and M_2 such that $M_1 \in P$ and $M_2 \notin P$ (*i.e.*, P is nontrivial);
2. For all Turing machines M_1 and M_2, if $L(M_1) = L(M_2)$, then $< M_1 > \in P \Leftrightarrow < M_2 > \in P$ (*i.e.*, P is a property of the Turing machines' languages).

Prove that P is undecidable.

Answer:

Assume by contradiction that P is a nontrivial property of the language of a Turing machine, and that P is decidable. Call D_P a Turing machine that decides P. We construct a Turing machine S that decides A_{TM} by using D_P.

*First, let $T_\emptyset$ be a Turing machine that accepts nothing (*i.e., *$L(T_\emptyset) = \emptyset$). Without loss of generality, assume $< T_\emptyset > \notin P$ (otherwise, we can proceed with $\overline{P}$). Now, P is nontrivial, so there exists a Turing machine T such that $< T > \in P$. We construct S the following way:*

$S :=$ *On input $< M, w >$:*

1. *Construct the following Turing machine:*

   $M_w :=$ *On input x:*

   a. *Simulate M on w. If it halts and rejects, then reject. If it halts and accepts, then go to step b.*

   b. *Simulate T on x. If it accepts, then accept.*

2. *Use D_P to decide whether $M_w \in P$. If it is the case, then accept; otherwise, reject.*

Now, M_w simulates T if M accepts w. Hence, $L(M_w) = L(T)$ if M accepts w, and $L(M_w) = \emptyset$ otherwise. This implies that $M_w \in P$ if and only if M accepts w. Thus, deciding P allows us to decide A_{TM}, a contradiction.

□

Exercise 4 : *Eating Rice.*

Use Rice's Theorem to prove that the following languages are undecidable:

1. $L_1 = \{< M > \mid M$ is a Turing machine and $L(M)$ is infinite$\}$
2. $L_2 = \{< M > \mid M$ is a Turing machine and $|L(M)| \geq 3\}$
3. $L_3 = \{< M > \mid M$ is a Turing machine and $L(M) = \Sigma^*\}$

Answer:

In all cases, we need to prove the two conditions of Rice's Theorem, that is: the property is nontrivial, and it depends only on the language of the Turing machines.

1. *It is easy to see that $L(M)$ being infinite is nontrivial (there are Turing machines with a finite language and Turing machines with an infinite language). Furthermore, it depends only on the language: if two Turing machines recognize the same language, then either both have their description in L_1 or none of them do. Hence, Rice's Theorem immplies that L_1 is undecidable.*
2. *Again, $L(M)$ having size at least 3 is nontrivial, and it depends only on the language. Rice's Theorem implies that L_2 is undecidable.*
3. *This is the same.*

□

Exercise 5 : *Poisoned Rice - do not eat.*

A *useless state* in a Turing machine is a state that is never entered on any input string.

Let $L = \{< M > \mid M$ has a useless state$\}$. Prove that L is undecidable. Can you use Rice's Theorem?

Answer:

*We cannot use Rice's Theorem since the propety of having a useless state is **not** does not depend only on the language: there can be two Turing machines M_1 and M_2 such that $L(M_1) = L(M_2)$ but M_1 has a useless state and M_2 does not.*
However, there is a direct reduction from the halting problem: if a Turing machine does not halt on input x, then its accepting state is useless. Assume by contradiction that D_u decides L, we construct the following Turing machine:

$S :=$ *On input $< M, w >$:*

1. *Construct the following Turing machine:*

   $M_w :=$ *On input x:*

   a. *Execute M on w. If it halts, then accept.*

2. *Execute D_u on $< M_w >$. If it accepts, then reject. Otherwise, accept.*

It is easy to see that $M_w \in L$ if and only if M does not halt on input w (since in this case, the accept state of M_w will never be used and thus is useless). This implies that deciding L allows us to decide $HALT_{TM}$, a contradiction.

□

Exercise 6 : *Rice with kayak.*
Prove that $L = \{< M > \mid M \text{ is a Turing machine and } L(M) \text{ contains a palindrome}\}$ is undecidable, first by using Rice's Theorem and then without it.
Answer:
Containing a palindrome is a nontrivial property (some Turing machines recognize palindromes, some do not), and it depends only on the languages (if two Turing machines accept the same language, then either both of their representations are in L or none is). Using Rice's Theorem, we deduce that L is undecidable.
Another proof is a reduction. Assume by contradiction that L is decidable and let D_p be a Turing machine that decides it. We construct the following Turing machine:

$S :=$ *On input $< M, w >$:*

1. *Construct the following Turing machine:*

   $M_w :=$ *On input x:*

   a. *If $x \neq aba$, then reject*

   b. *Run M on w. If it accepts, then accept.*

2. *Run D_p on $< M_w >$. If it accepts, then accept; otherwise, reject.*

It is easy to see that if M accepts w, then $L(M_w) = \{aba\}$ and thus D_p will accept $< M_w >$, which means that S will accept $< M, w >$. Conversely, if M rejects w (or loops), then $L(M_w) = \emptyset$ and thus D_p will reject $< M_w >$, which means that S will reject $< M, w >$. Hence, S decides A_{TM}, a contradiction.

□