Tutorial 7

(Un)Decidability, Rice's Theorem

Guidelines: Solve all problems in the class. *Do not search for solutions online.*

1. Show that the set $\{\mathcal{M} \mid \mathcal{M}$ is a DFA not accepting any string with odd number of 1's$\}$ is decidable.
   Hint: For a DFA $\mathcal{M}$, the problem of whether or not $L(\mathcal{M}) = \emptyset$ is decidable.

   Solution: We will use the fact that the emptiness problem for DFAs is decidable (just compute the set of states that are reachable from the start state and accept if any final state belongs to this set). In other words, there exists a TM $\mathcal{N}$ that decides $\{\mathcal{M} \mid \mathcal{M}$ is a DFA and $L(\mathcal{M}) = \emptyset\}$. Let

   $$B = \{\mathcal{M} \mid \mathcal{M} \text{ is a DFA not accepting any string with odd number of 1's}\}.$$

   Define a set $A = \{x \in \Sigma^* | x$ contains an odd number of 1's$\}$ and let $\mathcal{M}_A$ denote a DFA recognising A.

   Define a TM $\mathcal{K}$ that, on input a DFA $\mathcal{M}$, does the following:

   - Construct a DFA $\mathcal{M}'$ that accepts $L(\mathcal{M}) \cap L(\mathcal{M}_A)$ (here, the description of $\mathcal{M}_A$ is hardwired in $\mathcal{K}$'s finite control).
   - Run $\mathcal{N}$ on input $\mathcal{M}'$.
   - Accept if $\mathcal{N}$ accepts; reject otherwise.

   $\mathcal{K}$ accepts B since $L(\mathcal{M}) \cap A$ is empty iff $\mathcal{M}$ does not accept any string with odd number of 1's. Since both A and $L(\mathcal{M})$ are regular (the latter follows from the fact that $\mathcal{M}$ is a DFA) and regular sets are closed under intersection, the construction of $\mathcal{M}'$ accepting $L(\mathcal{M}) \cap L(\mathcal{M}_A)$ can be done in finite time. To be more precise, such a *product construction* can be done in time $O(|\mathcal{M}| \cdot |\mathcal{M}_A|)$ time, where $|\mathcal{M}|$ denotes the size of representation of $\mathcal{M}$. Hence $\mathcal{K}$ is total and decides B.

2. Recall the definition of linear bounded automaton (LBA) and that the halting problem for LBA is decidable. Prove by diagonalisation that there exists a recursive set that is not accepted by any LBA.

   Solution: Similar to TMs, we can define an encoding using bit strings for LBAs. Let $\mathcal{M}_\varepsilon, \mathcal{M}_0, \mathcal{M}_1, \mathcal{M}_{00}, \mathcal{M}_{01}, \ldots$ denote LBA's described according to this encoding. If a string x is not a valid description of an LBA, set $\mathcal{M}_x$ to be a trivial LBA that accepts and halts on all inputs. Construct an infinite matrix $\mathbf{A} = (a_{x,y})_{x,y=\varepsilon,0,1,00,01,\ldots}$ as

   $$a_{x,y} = \begin{cases} 1 & \text{if } y \in \mathcal{L}(\mathcal{M}_x) \\ 0 & \text{if } y \notin \mathcal{L}(\mathcal{M}_x) \end{cases}$$

   Now consider a set B defined as $B = \{x \mid x \notin \mathcal{L}(\mathcal{M}_x)\}$. The characteristic function for this set corresponds to the complement of the diagonal of $\mathbf{A}$. As for each $x \in \{0,1\}^*$, B differs from

$\mathcal{L}(\mathcal{M}_x)$ atleast on x, no LBA accepts B. We now prove that B is recursive by building a total TM $\mathcal{N}$ accepting B. Below is the description of $\mathcal{N}$.

$\mathcal{N}$: on input y

- Construct $\mathcal{M}_y$ from y.
- Simulate $\mathcal{M}_y$ on input y.
- If $\mathcal{M}_y$ accepts and halts, then reject and halt.
- If $\mathcal{M}_y$ rejects or loops, then accept and halt.

Using the fact that it is possible to check whether an LBA loops or not in finite time, we conclude that the TM $\mathcal{N}$ halts on all inputs and is hence total.

Note: Here I have indexed the automata with strings. You can use the natural numbers as index and refer to the LBA corresponding to a string x as $\mathcal{M}_{\eta(x)}$ where $\eta : \{0,1\}^* \to \mathbb{N}$ is an injective map. Such a map exists since $\{0,1\}^*$ is a countable set. □

3. True or False? It is decidable whether two given TMs accept the same set.

Solution: The problem is to show that the language $\mathsf{EQUIV} = \{(\mathcal{M}_1, \mathcal{M}_2) | \mathcal{L}(\mathcal{M}_1) = \mathcal{L}(\mathcal{M}_2)\}$ is undecidable. We show a reduction from the halting problem i.e., $\mathsf{HP} \leq_m \mathsf{EQUIV}$. Given an instance $(\mathcal{M}, x)$ of HP, construct an instance $(\mathcal{M}_1, \mathcal{M}_2)$ such that $\mathcal{M}_1$ and $\mathcal{M}_2$ have the following descriptions.

$\mathcal{M}_1$: on input y does the following

- Erase the input y.
- Accept and halt.

$\mathcal{M}_2$: on input y does the following

- Erase the input y.
- Run $\mathcal{M}$ on x.
- Accept and halt if $\mathcal{M}$ halts on x.

Clearly, $\mathcal{L}(\mathcal{M}_1) = \Sigma^*$ and

$$\mathcal{L}(\mathcal{M}_2) = \begin{cases} \Sigma^* & \text{if } \mathcal{M} \text{ halts on } x \\ \emptyset & \text{otherwise} \end{cases}$$

That is $(\mathcal{M}, x) \in \mathsf{HP}$ iff $(\mathcal{M}_1, \mathcal{M}_2) \in \mathsf{EQUIV}$. The descriptions of $\mathcal{M}_1, \mathcal{M}_2$ can be written down by a total TM. (Note that writing down descriptions of these TMs does not require running $\mathcal{M}$.) Since HP is undecidable, so is EQUIV.

4. Show that $\{\mathcal{M} | \mathcal{M}$ is a TM that halts on all inputs of length less than 300$\}$ is recursively enumerable but its complement is not.

Solution: Denote the language by T_{300}. Define a TM $\mathcal{N}$, that on input a TM $\mathcal{M}$, does the following:

- For each string w with $|w| < 300$, run $\mathcal{M}$ on w
- If $\mathcal{M}$ halts on all such strings w, then accept and halt

If $\mathcal{M} \in \mathsf{T}_{300}$, then it halts on all string of length less than 300 and so $\mathcal{N}$ accepts $\mathcal{M}$. Otherwise, $\mathcal{M}$ loops on some string w thus making $\mathcal{N}$ loop on $\mathcal{M}$. Hence $\mathcal{L}(\mathcal{N}) = \mathsf{T}_{300}$ establishing that T_{300} is recursively enumerable.

To show that $\neg\mathsf{T}_{300}$ is not *r.e.* , we provide a reduction from $\neg\mathsf{HP}$. Let $(\mathcal{M}, x)$ be an instance of $\neg\mathsf{HP}$. Define a TM $\mathcal{N}$, that does the following on input y:

- If $|y| >= 300$, then reject and halt (or accept or enter a trivial loop).
- Otherwise, run $\mathcal{M}$ on x.
- Accept and halt if $\mathcal{M}$ halts on x.

If $\mathcal{M}$ does not halt on x, then $\mathcal{N}$ loops on all inputs of length < 300. If $\mathcal{M}$ halts on x, then $\mathcal{N}$ halts on all inputs of length < 300. That is, $(\mathcal{M}, x) \in \neg\mathsf{HP} \Leftrightarrow \mathcal{N} \in \neg\mathsf{T}_{300}$.

5. Is the set $\{\mathcal{M} \mid \mathcal{L}(\mathcal{M})$ contains atmost 300 elements$\}$ *r.e.* ?

   Solution: This is a non-monotone property of *r.e.* sets. If an *r.e.* set A contains atmost 300 elements, then it is not necessary that all its supersets contain atmost 300 elements. Hence, it follows by Rice's theorem that the given set is not *r.e.*

6. Show that none of the following languages or their complements are *r.e.*

   (a) $\mathsf{REG} = \{\mathcal{M} \mid \mathcal{L}(\mathcal{M})$ is a regular set$\}$.

   Solution: We use Rice's theorem to show that REG is undecidable. Let $\mathscr{P}_{\mathsf{REG}}$ denote the property on *r.e.* sets defined as

$$\mathscr{P}_{\mathsf{REG}}(A) = \begin{cases} \mathsf{T} & \text{if } A \text{ is regular} \\ \mathsf{F} & \text{otherwise} \end{cases}$$

   Then deciding this property is equivalent to deciding REG. If a set A is regular it is not necessary that all its supersets are regular. In other words, there exist A, B such that $A \subseteq B$ and $\mathscr{P}_{\mathsf{REG}}(A) = \mathsf{T}$, $\mathscr{P}_{\mathsf{REG}}(B) = \mathsf{F}$. For instance, we can take $A = \phi$ and $B = \{0^n1^n \mid n \geq 0\}$. This shows that $\mathscr{P}_{\mathsf{REG}}$ is a non-monotone property. By Rice's theorem, $\mathscr{P}_{\mathsf{REG}}$ is not *r.e.* or equivalently REG is not *r.e.* Similarly, we can show that $\neg$REG or equivalently,

$$\mathscr{P}_{\neg\mathsf{REG}}(A) = \begin{cases} \mathsf{T} & \text{if } A \text{ is not regular} \\ \mathsf{F} & \text{otherwise} \end{cases}$$

   is not *r.e.* by proving that $\mathscr{P}_{\neg\mathsf{REG}}$ is a non-monotone property. We only need to exhibit two sets A, B with $A \subseteq B$ such that $\mathscr{P}_{\neg\mathsf{REG}}(A) = \mathsf{T}$ and $\mathscr{P}_{\neg\mathsf{REG}}(B) = \mathsf{F}$. Taking $A = \{0^n1^n \mid n \geq 0\}$ and $B = \{0^*1^*\}$ suffices.

   (b) $\mathsf{TOT} = \{\mathcal{M} \mid \mathcal{M}$ halts on all inputs$\}$.

   Solution: We show TOT is not *r.e.* via a reduction from $\neg\mathsf{HP}$. Given an instance $(\mathcal{M}, x)$ of $\neg\mathsf{HP}$, we construct a TM $\mathcal{N}$ such that $(\mathcal{M}, x) \in \neg\mathsf{HP}$ iff $\mathcal{N} \in \mathsf{TOT}$. Below is a description of $\mathcal{N}$.

   $\mathcal{N}$: on input y does the following.

- Store y on a separate track of the tape.
- Run $\mathcal{M}$ on x for $|y|$ steps. ($\mathcal{M}$ and x are hardwired in $\mathcal{N}$'s finite control.)
- If $\mathcal{M}$ does not halt within $|y|$ steps, then halt and either accept or reject.
- Otherwise, enter a trivial loop.

If $\mathcal{M}$ does not halt on x, then $\mathcal{N}$ halts on all input strings. Suppose that $\mathcal{M}$ halts on x in n steps. Then for any string y with $|y| < n$, $\mathcal{N}$ accepts y since $\mathcal{M}$ does not halt on x within $|y|$ steps. On the other hand, for any string y with $|y| \geq n$, $\mathcal{N}$ does not halt. That is,

$$\mathcal{M} \text{ does not halt on } x \implies \mathcal{N} \text{ halts on all input strings} \implies \mathcal{N} \in \mathsf{TOT}$$

$$\mathcal{M} \text{ halts on } x \implies \mathcal{N} \text{ does not halt on some input strings} \implies \mathcal{N} \notin \mathsf{TOT}$$

Since $\neg\mathsf{HP}$ is not *r.e.* , TOT is not *r.e.*

Next, we prove that $\neg\mathsf{TOT}$ is not *r.e.* through a reduction from $\neg\mathsf{HP}$. Given an instance $(\mathcal{M}, x)$ of $\neg\mathsf{HP}$, we construct a TM $\mathcal{N}$ such that $(\mathcal{M}, x) \in \neg\mathsf{HP}$ iff $\mathcal{N} \in \neg\mathsf{TOT}$. Below is a description of $\mathcal{N}$.

$\mathcal{N}$: on input y does the following.

- Store y on a separate track of the tape.
- If $|y| \leq |x|$, halt and accept.
- Run $\mathcal{M}$ on x.
- If $\mathcal{M}$ halts on y, then halt and accept.

If $\mathcal{M}$ does not halt on x, then $\mathcal{N}$ halts on all input strings of length at most $|x|$. If $\mathcal{M}$ halts on x, then $\mathcal{N}$ accepts any string y. That is,

$$\mathcal{M} \text{ does not halt on } x \implies \mathcal{N} \text{ does not halt on some inputs} \implies \mathcal{N} \in \neg\mathsf{TOT}$$

$$\mathcal{M} \text{ halts on } x \implies \mathcal{N} \text{ halts on all input strings} \implies \mathcal{N} \notin \neg\mathsf{TOT}$$

Since $\neg\mathsf{HP}$ is not *r.e.* , $\neg\mathsf{TOT}$ is not *r.e.*

7. Let

$$f(x) = \begin{cases} 3x+1 & \text{if } x \text{ is odd} \\ x/2 & \text{if } x \text{ is even} \end{cases}$$

for any natural number x. Define $\mathsf{C}(x)$ as the sequence $x, f(x), f(f(x)), \ldots$, which terminates if and when it hits 1. For example, if $x = 7$, then

$$\mathsf{C}(x) = (7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1).$$

Computer tests have shown that $\mathsf{C}(x)$ hits 1 eventually for x ranging from 1 to 87×2^{60} (as of 2017). But, the question of whether $\mathsf{C}(x)$ ends at 1 for all $x \in \mathbb{N}$ is not proven. This is believed to be true and known as the Collatz conjecture. Suppose that MP were decidable by a Turing machine $\mathcal{K}$. Use $\mathcal{K}$ to describe a TM that is guaranteed to prove or disprove Collatz conjecture.

Solution: Let $\mathcal{N}$ be a TM, that on input x, does the following.

- while $x \neq 1$,

- if x is even, $x \leftarrow x/2$
- otherwise $x \leftarrow 3x + 1$

- accept and halt

Let $\mathcal{M}$ be a TM, that on input y, does the following.

- erase y
- set $x \leftarrow 1$ and repeat the following
  - use $\mathcal{K}$ to determine whether $\mathcal{N}$ accepts x
  - if not, accept and halt
  - otherwise, $x \leftarrow x + 1$

If the Collatz conjecture is true, then $\mathcal{M}$ runs forever; otherwise $\mathcal{M}$ halts after finding a counter example (in fact, the smallest counter example).

We now use $\mathcal{K}$ to determine whether or not $\mathcal{M}$ accepts an arbitrary input y to decide whether or not Collatz conjecture is true.

8. (a) Show that the language

$$\{(\mathcal{M},\mathcal{N}) \mid \mathcal{M},\mathcal{N} \text{ are Turing machines and } L(\mathcal{M}) \cap L(\mathcal{N}) = \emptyset\}$$

is undecidable via reduction.

Solution: Let EI-TM the above langauge. We show that $\neg\mathsf{HP} \leq_m$ EI-TM. Let $(\mathcal{K}, x)$ be an instance of $\neg\mathsf{HP}$. Construct $\mathcal{M}$ so that it accepts any input. Clearly $L(\mathcal{M}) = \Sigma^*$. Let $\mathcal{N}$ be defined so that, on input y it does the following: store y on a separate track; simulate $\mathcal{K}$ on x. If it halts, then accept y. If $(\mathcal{K}, x) \in \neg\mathsf{HP}$, then $L(\mathcal{N}) = \emptyset$ and $L(\mathcal{M}) \cap L(\mathcal{N}) = \emptyset$. If $(\mathcal{K}, x) \notin \neg\mathsf{HP}$, then $L(\mathcal{N}) = \Sigma^*$ and as a result $\mathcal{M} \cap \mathcal{N} \neq \emptyset$. Since $\neg\mathsf{HP}$ is undecidable, so is EI-TM.

(b) Prove the following extension of Rice's theorem (of which part (a) is a special case):

Every non-trivial property of pairs of r.e. sets is undecidable.

More formally, let $\mathscr{P} : \{r.e.\text{ sets}\} \times \{r.e.\text{ sets}\} \to \{\top, \bot\}$ be a non-trivial property on pairs of *r.e.* sets. Then show that

$$T_{\mathscr{P}} = \{(\mathcal{M},\mathcal{N}) \mid \mathcal{M} \text{ and } \mathcal{N} \text{ are TMs and } \mathscr{P}(L(\mathcal{M}), L(\mathcal{N})) = \top\}$$

is undecidable.

Solution: Let $\mathscr{P}$ be a non-trivial property of pairs of *r.e.* sets. Assume w.l.g. that $\mathscr{P}(\emptyset, \emptyset) = \bot$. Since $\mathscr{P}$ is non-trivial, there exist *r.e.* sets L_1, L_2 such that $\mathscr{P}(L_1, L_2) = \top$. Let $\mathcal{M}_1$, $\mathcal{M}_2$ be TMs recognising L_1, L_2 repectively. We show that $\mathsf{HP} \leq_m T_{\mathscr{P}}$. (If $\mathscr{P}(\emptyset, \emptyset) = \top$, then we can show that $\neg\mathsf{HP} \leq_m T_{\mathscr{P}}$ via a similar argument.) Let $\mathcal{N}, x$ be an instance of HP. Construct $\mathcal{M}_1'$ that on input w does the following:

- Run $\mathcal{N}$ on x ($\mathcal{N}, x$ are hardwired in the finite control of $\mathcal{M}_1'$).
- If $\mathcal{N}$ halts, simulate $\mathcal{M}_1$ on w.
- Accept if $\mathcal{M}_1$ accepts.

Similarly, construct $\mathcal{M}_2'$ that on input w does the following:

- Simulate $\mathcal{N}$ on x
- If $\mathcal{N}$ halts, simulate $\mathcal{M}_2$ on w.
- Accept if $\mathcal{M}_2$ accepts.

If $\mathcal{N}$ halts on x, then $L(\mathcal{M}_1') = L(\mathcal{M}_1)$ and $L(\mathcal{M}_2') = L(\mathcal{M}_2)$. As a result, $\mathscr{P}(L(\mathcal{M}_1'), L(\mathcal{M}_2')) = \top$. If $\mathcal{N}$ does not halt on x, then $L(\mathcal{M}_1') = L(\mathcal{M}_2') = \emptyset$ whence $\mathscr{P}(L(\mathcal{M}_1'), L(\mathcal{M}_2')) = \bot$. In other words, $(\mathcal{N}, x) \in \mathsf{HP} \Leftrightarrow (\mathcal{M}_1', \mathcal{M}_2') \in T_{\mathscr{P}}$. Therefore, $T_{\mathscr{P}}$ is undecidable.