

Tutorial 4

Computability Theory

Recursive and R.E. Languages, (Un)Decidability, Reduction and Rice's Theorem

1. For a language L over the alphabet $\{0, 1\}$, define the following two languages:

$$\text{HALF}_1(L) = \{x \mid x \in \Sigma^*, \text{ and there exists } y \in \Sigma^* \text{ such that } |x| = |y| \text{ and } xy \in L\}$$

$$\text{HALF}_2(L) = \{x \mid x \in \Sigma^*, \text{ and there exists } y \in \Sigma^* \text{ such that } |x| = |y| \text{ and } yx \in L\}$$

Prove/Disprove the following (for $k = 1$ and $k = 2$):

- (a) If L is R.E. (recursively enumerable), then $\text{HALF}_k(L)$ must be R.E..
 (b) If L is recursive, then $\text{HALF}_k(L)$ must be recursive.

2. Consider the following languages for (finite) $k \in \mathbb{N}$:

$$\text{LOOP}_{\text{LE}} = \{\mathcal{M} \mid \mathcal{M} \text{ is (the encoding of) a DTM that loops on at most } k \text{ input strings}\}$$

$$\text{LOOP}_{\text{GE}} = \{\mathcal{M} \mid \mathcal{M} \text{ is (the encoding of) a DTM that loops on at least } k \text{ input strings}\}$$

$$\text{LOOP}_{\text{LT}} = \{\mathcal{M} \mid \mathcal{M} \text{ is (the encoding of) a DTM that loops on less than } k \text{ input strings}\}$$

$$\text{LOOP}_{\text{GT}} = \{\mathcal{M} \mid \mathcal{M} \text{ is (the encoding of) a DTM that loops on more than } k \text{ input strings}\}$$

$$\text{LOOP}_{\text{EQ}} = \{\mathcal{M} \mid \mathcal{M} \text{ is (the encoding of) a DTM that loops on exactly } k \text{ input strings}\}$$

Determine whether the above languages (and the complements of these languages) are recursive, R.E., or non-R.E.

3. Consider the following languages for (finite) $k \in \mathbb{N}$:

$$\text{HALT}_{\text{LE}} = \{\mathcal{M} \mid \mathcal{M} \text{ is (the encoding of) a DTM that halts on at most } k \text{ input strings}\}$$

$$\text{HALT}_{\text{GE}} = \{\mathcal{M} \mid \mathcal{M} \text{ is (the encoding of) a DTM that halts on at least } k \text{ input strings}\}$$

$$\text{HALT}_{\text{LT}} = \{\mathcal{M} \mid \mathcal{M} \text{ is (the encoding of) a DTM that halts on less than } k \text{ input strings}\}$$

$$\text{HALT}_{\text{GT}} = \{\mathcal{M} \mid \mathcal{M} \text{ is (the encoding of) a DTM that halts on more than } k \text{ input strings}\}$$

$$\text{HALT}_{\text{EQ}} = \{\mathcal{M} \mid \mathcal{M} \text{ is (the encoding of) a DTM that halts on exactly } k \text{ input strings}\}$$

Determine whether the above languages (and the complements of these languages) are recursive, R.E., or non-R.E.

4. Let \mathcal{N} be a Non-deterministic Turing machine (NTM). We say that \mathcal{N} faces a dilemma if at some point in its working, it encounters a situation where the finite control is in the state p , the head scans the tape symbol a , and $\delta(p, a)$ offers multiple (two or more) possibilities, where p is neither the accept nor the reject state. Consider the following two languages.

$$\text{DILEMMA}_\epsilon = \{\mathcal{N} \mid \mathcal{N} \text{ is an NTM which faces a dilemma at least once on input } \epsilon\},$$

$$\text{DILEMMA}_\star = \{\mathcal{N} \mid \mathcal{N} \text{ is an NTM which faces a dilemma at least once on each input}\}.$$

Answer the following.

- (a) Prove that DILEMMA_ϵ is R.E., but not recursive.
 (b) Prove that DILEMMA_\star is non-R.E. (not recursively enumerable).

5. Show that the following language is decidable.

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

(Hint: For a DFA M , the problem of whether or not $\mathcal{L}(M) = \phi$ is decidable.)

6. Recall that the halting problem for linear bounded automaton (LBA) is decidable. Prove by diagonalization that there exists a recursive set that is not accepted by any LBA.

7. Show, using reductions, that none of the following languages nor their complements are R.E.

$$\begin{array}{ll} \text{(a) REG} = \{M \mid \mathcal{L}(M) \text{ is a regular}\} & \text{(c) REC} = \{M \mid \mathcal{L}(M) \text{ is a recursive}\} \\ \text{(b) CFL} = \{M \mid \mathcal{L}(M) \text{ is a context-free}\} & \text{(d) TOT} = \{M \mid M \text{ halts on all inputs}\} \end{array}$$

Now, use Rice's theorems to prove the same (as asked above).

8. 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 $C(x)$ as the sequence $(x, f(x), f(f(x)), \dots)$ which terminates if and when it hits 1. For example, if $x = 7$, then

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

Computer tests have shown that $C(x)$ hits 1 eventually for x ranging from 1 to $2^{68} \approx 2.95 \times 10^{20}$ (as of 2026). But, the question of whether $C(x)$ 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 TM \mathcal{K} . Use \mathcal{K} to describe a TM that is guaranteed to prove or disprove Collatz conjecture.

9. Prove or Disprove: The language $\{M \mid M \text{ is a DTM that runs in time } O(n^3)\}$ is undecidable.

10. For a language $A \subseteq \Sigma^*$ (with $|\Sigma| \geq 2$), define $A^R = \{w^R \mid w \in A\}$, where w^R denotes the reverse of the string w . Is it decidable, for a given TM \mathcal{M} , whether $\mathcal{L}(\mathcal{M}) = \mathcal{L}(\mathcal{M})^R$?

11. Prove/Disprove whether the following problems on a TM \mathcal{M} are decidable for (finite) $k \in \mathbb{N}$.

- Decide whether \mathcal{M} halts on some input within k steps.
- Decide whether \mathcal{M} halts on some input beyond k steps.
- Decide whether \mathcal{M} halts on all inputs within k steps.
- Decide whether \mathcal{M} halts on all inputs beyond k steps.
- Decide whether \mathcal{M} runs for at least k^k steps for input a^k .
- Decide whether \mathcal{M} runs for at most k^k steps for input a^k .
- Decide whether \mathcal{M} on input ϵ moves left at least k times.
- Decide whether \mathcal{M} on a given input w moves left at least k times.

12. Is the problem whether a TM on any input re-enters the start state decidable or not? Prove.

13. Prove that the following languages are not recursive.

$$\begin{array}{l} \text{(a) TMB} = \{M \# w \mid M \text{ writes the blank symbol at some point of time on input } w\} \\ \text{(b) TMS} = \{M \# w \# \$ \mid M \text{ writes the symbol } \$ \in \Gamma \text{ at some point of time on input } w\} \end{array}$$

14. Determine whether the following languages are recursive, R.E. or neither? Justify the answer.

$$\begin{array}{l} \text{(a) } L_{GE} = \{M \mid \mathcal{L}(M) \text{ contains at least 2026 elements}\} \\ \text{(b) } L_{LE} = \{M \mid \mathcal{L}(M) \text{ contains at most 2026 elements}\} \\ \text{(c) } L_{EQ} = \{M \mid \mathcal{L}(M) \text{ contains exactly 2026 elements}\} \end{array}$$

15. Determine whether the following languages are recursive, R.E. or neither? Justify the answer.
- $L_{AM} = \{M \mid M \text{ accepts at most 2026 input strings}\}$
 - $L_{AL} = \{M \mid M \text{ accepts at least 2026 input strings}\}$
 - $L_{AA} = \{M \mid M \text{ accepts all strings of length } \leq 2026\}$
 - $L_{AS} = \{M \mid M \text{ accepts some strings of length } \geq 2026\}$
 - $L_{NA} = \{M \mid M \text{ does not accept all strings of length } \geq 2026\}$
 - $L_{NS} = \{M \mid M \text{ does not accept some string of length } \leq 2026\}$
16. For two TMs (M, N) , determine whether the following are *decidable*, *semi-decidable*, or *not*.
- M takes more steps than N on input ϵ .
 - M does not take more steps than N on input ϵ .
17. Let $\text{nsteps}(M, w)$ denote the number of steps taken by M on w . If M loops on w , take $\text{nsteps}(M, w) = \infty$. If N also loops on v , take $\text{nsteps}(M, w) = \text{nsteps}(N, v)$. Prove whether the following languages are *Recursive*, or *R.E. but not Recursive*, or *non-R.E.*
- $L_1 = \{M \# N \mid \text{nsteps}(M, \epsilon) < \text{nsteps}(N, \epsilon)\}$
 - $L_2 = \{M \# N \mid \text{nsteps}(M, \epsilon) \leq \text{nsteps}(N, \epsilon)\}$
 - $L_3 = \{M \# N \mid \text{nsteps}(M, w) < \text{nsteps}(N, v) \text{ for some } w, v\}$
 - $L_4 = \{M \# N \mid \text{nsteps}(M, w) < \text{nsteps}(N, v) \text{ for all } w, v\}$
18. Using reductions, prove that the following languages are not recursive (undecidable).
- $L_a = \{M \# N \mid \mathcal{L}(M) = \mathcal{L}(N)\}$
 - $L_b = \{M \# N \mid \mathcal{L}(M) \subseteq \mathcal{L}(N)\}$
 - $L_c = \{M \# N \mid \mathcal{L}(M) \cap \mathcal{L}(N) = \phi\}$
 - $L_d = \{M \# N \mid \mathcal{L}(M) \cap \mathcal{L}(N) \text{ is finite}\}$
 - $L_e = \{M \# N \mid \mathcal{L}(M) \cap \mathcal{L}(N) \text{ is regular}\}$
 - $L_f = \{M \# N \mid \mathcal{L}(M) \cap \mathcal{L}(N) \text{ is context-free}\}$
 - $L_g = \{M \# N \mid \mathcal{L}(M) \cap \mathcal{L}(N) \text{ is recursive}\}$
 - $L_h = \{M \# N \# \mathcal{P} \mid \mathcal{L}(M) \cap \mathcal{L}(N) = \mathcal{L}(\mathcal{P})\}$
19. Prove/Disprove: *No non-trivial property of recursively enumerable languages is semidecidable.*
20. **[Generalization of Rice's Theorem for Pairs of R.E. Languages]** Consider the set of pairs of R.E. languages: $RE^2 = \{(L, L') \mid L, L' \in RE\}$. Answer the following.
- Define a property of pairs of R.E. languages.
 - How do you specify a property of a pair of R.E. languages?
 - Which properties of pairs of R.E. languages should be called non-trivial?
 - Prove that every non-trivial property of pairs of R.E. languages is undecidable.
21. Use the previous exercise to prove that the following problems about pairs of R.E. languages are undecidable.
- $\mathcal{L}(M) = \mathcal{L}(N)$
 - $\mathcal{L}(M) \subseteq \mathcal{L}(N)$
 - $\mathcal{L}(M) \cap \mathcal{L}(N) = \phi$
 - $\mathcal{L}(M) \cap \mathcal{L}(N)$ is finite
 - $\mathcal{L}(M) \cap \mathcal{L}(N)$ is context-free
 - $\mathcal{L}(M) \cap \mathcal{L}(N)$ is recursive
 - $\mathcal{L}(M) \cup \mathcal{L}(N) = \Sigma^*$
 - $\mathcal{L}(M) \cup \mathcal{L}(N) = \phi$
 - $\mathcal{L}(M) \cup \mathcal{L}(N)$ is finite
 - $\mathcal{L}(M) \cup \mathcal{L}(N)$ is recursive
22. Generalize Rice's theorem, Part 2 (monotone-property related), for pairs of R.E. languages.