CPSC 421: Introduction to Theory of Computing
Practice Problem Set #3, Not to be handed in

1. Show that the following problems are decidable by describing a single-tape TM that decides it.

   (a) $L_1 = \{ 0^n 1^m : n, m \geq 1 \}$

   (b) $L_2 = \{ 0^n 1^n : n \geq 0 \}$

2. Informally and briefly describe a TM to decide the following languages. You may use multiple tapes if you wish.

   (a) $L_1 = \{ a^n b^n c^n : n \geq 0 \}$

   (b) $L_2 = \{ x\#y\#z : x + y = z \}$

   (c) $L_3 = \{ w \in \{0,1\}^* : w$ contains twice as many 0s as 1s $\}$

3. Show that the collection of decidable languages is closed under the operation of

   (a) union

   (b) concatenation

   (c) star

   (d) complementation

   (e) intersection

4. Show that the collection of Turing-recognizable languages is closed under the operation of

   (a) union

   (b) concatenation

   (c) star

   (d) intersection

   Give an example to show that the collection of Turing-recognizable languages is not closed under complementation.

5. Is the collection of decidable languages closed under countable unions? If so, give a proof. If not, give a counterexample.

6. (Another TM variant) A Turing machine with left reset is similar to an ordinary Turing machine, but the transition function has the form

$$\delta\colon Q \times \Gamma \leftarrow Q \times \Gamma \times \{R, RESET\}.$$

   If $\delta(q, a) = (r, b, RESET)$, when the machine is in state q reading an a, the machine's head jumps to the left-hand end of the tape after it writes b on the tape and enters state r. Note that these machines do not have the usual ability to move the head one symbol left. Show that Turing machines with left reset recognize the class of Turing-recognizable languages.

7. Let $INFINITE_{DFA} = \{ \langle A \rangle : A$ is a DFA and $L(A)$ is an infinite language $\}$. Show that $INFINITE_{DFA}$ is decidable.

8. Let $S = \{ \langle M \rangle : M \text{ is a DFA that accepts } w^R \text{ whenever it accepts } w \}$. Show that S is decidable. *Hint:* Use the fact that $EQ_{DFA} = \{ \langle A, B \rangle : A \text{ and } B \text{ are DFAs and } L(A) = L(B) \}$ is decidable.

9. Let $BLANKHALT = \{ \langle M \rangle : M \text{ halts on the empty input} \}$. Show that $BLANKHALT$ is undecidable.

10. Consider the problem of determining whether a single-tape Turing machine ever writes a blank symbol over a nonblank symbol during the course of its computation on any input string. Formulate this problem as a language, and show that it is undecidable.

11. Consider the problem of determining whether a Turing machine M on an input w ever attempts to move its head left when its head is on the left-most tape cell. Formulate this problem as a language and show that it is undecidable.

12. Consider the problem of determining whether a Turing machine M on an input w ever attempts to move its head at any point during its computation of w. Formulate this problem as a language and show that it is **decidable**.

13. When we are designing a Turing machine, it can often be desirable to use as few states as possible. A seemingly innocent heuristic to do this is to determine if a state will ever be used and, if not, to remove that state (and all its transitions). In a real program (say written using C or Python), this may refer to eliminating dead code and, thus, reduce the program size. Given a Turing machine M and a state q, show that it is undecidable to determine if M ever enters state q.

14. Let $f : \mathbb{N} \to \mathbb{N}$ be defined as

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

   Starting with a positive integer x and iterate f, you obtain the sequence, x, $f(x)$, $f(f(x))$, etc. The sequence stops whenever it hits 1. For example, if $x = 3$, we obtain the sequence $3, 10, 5, 16, 8, 4, 2, 1$. The Collatz conjecture asks if for any starting x, the sequence will eventually stop at 1.

   Suppose that H is a TM that solves the halting problem. Use this to design a TM that decides if the sequence starting at x will eventually hit 1 and stop.

15. Prove that $\{1\}^*$ contains an undecidable subset.

16. Describe two different Turing machines, M and N, that, when started on any input, M outputs $\langle N \rangle$ and N outputs $\langle M \rangle$.

17. Consider the following statement.

   "The smallest positive integer that cannot be described by less than fifteen words".

   Does such a number exist?

18. (*) Let $\Sigma = \{1\}$ and $A = \{ 1^n : \pi \text{ contains the string } 1^n \text{ in its decimal expansion} \}$. Is A decidable?

19. (*) Let $\Gamma = \{0, 1, \sqcup\}$ be the tape alphabet for all TMs. Define the busy beaver function $f : \mathbb{N} \to \mathbb{N}$ as follows. For each value of k, consider all k-state TMs that halt when started with a blank tape. Let $f(k)$ be the maximum number of 1s that remain on the tape among all of these machines. Show that f is not a computable function.

20. (*) We say that P is a nontrivial property of the language of a Turing machine if there exists TMs M_1 and M_2 such that $L(M_1)$ satisfies property P and $L(M_2)$ does not satisfy property P. For example, the property that a TM accepts no string is nontrivial (why?).

    Theorem 1 (Rice's Theorem). Let P be any nontrivial property. The language $L_P = \{ \langle M \rangle : L(M) \text{ satisfies } P \}$ is undecidable.

    (a) Prove Rice's Theorem.

    (b) Use Rice's Theorem to show that the language

    $$L = \left\{ \langle M \rangle : |L(M')| \text{ is prime} \right\}$$

    is undecidable.

21. (*) An oracle Turing machine is a Turing machine M with a magical read/write oracle tape along with three additional states q_{yes}, q_{no}, q_{ask}. The oracle tape is specified with a language O. Whenever M enters the state q_{ask}, M moves to state q_{yes} if $w \in O$ and q_{no} if $w \notin O$. Here, w denotes the contents of the oracle tape when M enters q_{ask}. Note that an oracle call counts only as a single computational step.

    We will denote a TM with oracle O by M^O. If we omit the superscript then the TM has no oracle.

    (a) Let $HALT = \{ \langle M, w \rangle : M \text{ is a TM and halts on input } w \}$. Show that $HALT$ is decidable by a TM with oracle $HALT$.

    (b) Let $HALT^{HALT} = \left\{ \langle M^{HALT}, w \rangle : M^{HALT} \text{ is a TM with oracle } HALT \text{ and halts on input } w \right\}$. Show that $HALT^{HALT}$ is undecidable by a TM with oracle $HALT$.

    Remark. Although oracle Turing machines seem fairly contrived, they have been used to prove that certain techniques for attacking the P vs NP problem will not work. This is known as the Baker-Gill-Solovay Theorem. We will cover P and NP later in the course, but not this particular theorem.