Homework 7 – Due Thursday, November 3, 2022 at 11:59 PM

Reminder Collaboration is permitted, but you must write the solutions by yourself without assistance, and be ready to explain them orally to the course staff if asked. You must also identify your collaborators and write “Collaborators: none” if you worked by yourself. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Note You may use various generalizations of the Turing machine model we have seen in class, such as TMs with two-way infinite tapes, stay-put, or multiple tapes. If you choose to use such a generalization, state clearly and precisely what model you are using.

Problems There are 4 required problems and 1 bonus problem.

1. (Countable sets)
   (a) Whitespace [https://en.wikipedia.org/wiki/Whitespace_(programming_language)] is a programming language in which valid programs consist only of spaces (S), tabs (T), and linebreaks (L). Show that the set \( W \) of all (finite) Whitespace programs is countable. (For the sake of readability, you can use the symbols S, T, L to stand for legal Whitespace characters in your proof.)
   (b) A formal polynomial in variable \( x \) is an expression of the form \( c_0 + c_1x + c_2x^2 + \ldots + c_dx^d \), where \( d \geq 0 \) is an integer, and each coefficient \( c_i \) is an integer. Let \( P \) be the set of all formal polynomials in variable \( x \). Show that \( P \) is countable.
   (c) Show that if \( A \) and \( B \) are countable sets, then their union \( A \cup B \) is also countable.

2. (Uncountable sets)
   (a) A game of Tetris is an infinite sequence of “tetrominos” which can have one of seven types \( 1, \ldots, 7 \). For example, a Tetris game may look like \( (7, 4, 2, 2, \ldots) \). Use a proof by diagonalization to show that the set \( T \) of all Tetris games is uncountable.
   (b) An infinite sequence \( (x_1, x_2, x_3, \ldots) \) is collision-free if no number appears more than once in that sequence. For example, \( (1, 5, 2, 5, \ldots) \) is not collision-free because 5 appears at least twice. Use a proof by diagonalization to show that the set \( C \) of all collision-free infinite sequences of natural numbers is uncountable.
     Hint: When you construct a sequence contradicting the diagonal, make sure that it is indeed a member of \( C \).

3. (Reduction Mad-Libs) A language \( B \) is excited if every string in \( B \) takes the form \( ww \) for some \( w \in \{0,1\}^* \). For example, \( \{00, 1111, 1010\} \) is excited, but \( \{00, 10\} \) is not excited. The language \( EX_{TM} = \{\langle M \rangle \mid L(M) \text{ is excited}\} \) corresponds to the following computational problem: Given the encoding of a TM \( M \), does \( M \) recognize an excited language? This exercise will walk you through a proof, by reduction, that \( EX_{TM} \) is undecidable.
   Assume, for the sake of contradiction, that \( EX_{TM} \) is decidable by a TM \( R \). That is, there is a TM \( R \) that accepts \( \langle M \rangle \) whenever \( L(M) \) is excited, and rejects \( \langle M \rangle \) whenever \( L(M) \) is not excited. We will use \( R \) to construct a new TM \( S \) that decides the (undecidable) language \( A_{TM} \).
Algorithm: $S(⟨M, w⟩)$

Input : Encoding of a basic TM $M$ over input alphabet $\{0, 1\}$, string $w \in \{0, 1\}^*$

1. Construct the following TM $N$:
   
   $N = \text{"On input a string } x \in \{0, 1\}^*:\n   \text{If } x = 00, \text{ accept.}\n   \text{Else, run } M \text{ on input } w. \text{ If it accepts, accept. Otherwise, reject."}$

2. Run $R$ on input $⟨N⟩$. If it accepts, (i). Otherwise, (ii).

(a) Consider the machine $N$ constructed inside algorithm $S$. If $M$ accepts on input $w$, what is the language $L(N)$? Is $L(N)$ excited in this case?

(b) If $M$ does not accept on input $w$, what is the language $L(N)$? Is $L(N)$ excited in this case?

(c) Fill in the blanks labeled (i) and (ii) with accept or reject decisions to guarantee the following conditions: If $M$ accepts input $w$, then $S$ accepts input $⟨M, w⟩$, and if $M$ does not accept input $w$, then $S$ rejects input $⟨M, w⟩$. Use parts (a) and (b) to explain why these conditions hold for your choices of how to fill in the blanks.

(Your job is done now, but you may want to keep reading to see the exciting conclusion.) By part (c), the TM $M$ exactly decides the language $A_{TM}$. But this language is undecidable, which is a contradiction. Hence our assumption that $EX_{TM}$ was decidable is false, so we conclude that $EX_{TM}$ is an undecidable language.

4. (Unary acceptance) A Turing machine $M$ accepts a unary string if there exists a string $x \in \{1\}^*$ such that $M$ accepts on input $x$. Consider the problem of determining whether (the encoding of) a TM $M$ with input alphabet $\{0, 1\}$ accepts a unary string.

(a) Formulate this problem as a language $UA$. Caution: The only input to this computational problem is $⟨M⟩$ for a TM $M$.

(b) Consider the following “first attempt” at designing a TM $N$ recognizing the language $UA$.

Algorithm: $N(⟨M⟩)$

Input : Encoding of a basic TM $M$ over input alphabet $\{0, 1\}$

1. Let $x_n = 1^n$. For each $n = 0, 1, 2, 3, \ldots$:
2. Run $M$ on input $x_n$. If it accepts, accept. Otherwise, continue.

Explain why $N$ fails to recognize $UA$.

(c) Design and analyze a Turing machine that does recognize the language $UA$. Hint: Solved exercise 4.5 in Sipser illustrates the “dovetailing” trick that will be useful here.

(d) Prove that the language $UA$ is undecidable.

Hint: Give a reduction from the undecidable language $A_{TM}$. That is, you should assume for the sake of contradiction that $UA$ is decidable. Then under this assumption, construct a TM deciding $A_{TM}$, prove that this decider is correct, and as a result conclude that your assumption that $UA$ is decidable must have been false.

5. (Bonus problem) Define the language $XOR_{TM} = \{⟨M, w, v⟩ | M$ is a TM that accepts exactly one of the strings $w, v\}$. Prove that both $XOR_{TM}$ and its complement $\overline{XOR}_{TM}$ are unrecognizable.