CS172 Computability & Complexity, Spring 2021

Homework 7

Out: 12 Mar. Due: 19 Mar.

Instructions: *Submit your solutions in pdf format on Gradescope by* **5pm on Friday, March 19**. *Solutions may be written either in LaTeX (with either machine-drawn or hand-drawn diagrams) or* **legibly** *by hand. (The LaTeX source for this homework is provided in case you want to use it as a template.) Please be sure to begin the solution for each problem on a new page, and to tag each of your solutions to the correct problem! Per course policy, no late solutions will be accepted. Take time to write* **clear** *and* **concise** *answers; confused and long-winded solutions may be penalized. You are encouraged to form small groups (two to four people) to work through the homework, but you* **must** *write up all your solutions on your own. Depending on grading resources, we reserve the right to grade a random subset of the problems and check off the rest; so you are advised to attempt all the problems.*

1. Classify each of the following languages as either decidable, recognizable but not decidable, or not recognizable. In each case, justify your answer with a proof.

(a) Given a Turing machine M and a state q, does M enter state q more than once on any input?

(b) Given a Turing machine M, does M enter any state more than once on any input?

(c) Given a Turing machine M, is the language recognized by M the set of all 0-1 strings with an equal number of 0's and 1's?

(d) $PCP_1 = \{\langle P \rangle : P$ is an instance of the Post Correspondence Problem with unary alphabet $\{1\}$, and P has a match$\}$.

2. The *Planar Tiling Problem* (PTP) is defined as follows. The input is a finite set S of square tiles with unit side lengths, each of whose four sides is labeled with a string from Σ^*. One of the tiles is designated as a *corner* tile. The problem is to arrange the tiles in non-overlapping fashion so that they cover the entire infinite upper right-hand quadrant of the plane (i.e., the region $\{(x, y) : x \geq 0, y \geq 0\}$), with the corner tile sitting in the bottom left corner (i.e., with its bottom left corner at the origin). The constraint is that two tiles may be placed next to each other if and only if the labels on their adjacent sides are equal. Of course, as in the PCP discussed in class, tiles may be reused as often as desired.

Show that the PTP is undecidable (in fact, not even recognizable) by giving a mapping reduction from $\overline{\text{HALT}_{\text{TM}}}$, the complement of the Halting Problem.

[HINTS: This is similar in spirit — but different in detail — to the reduction we saw in class, and in Sipser 5.2, from A_{TM} to the PCP. Try to be systematic and follow the design process we saw for PCP. Given an input $\langle M, w \rangle$ for $\overline{\text{HALT}_{\text{TM}}}$, design your tiles so that successive horizontal rows of tiles represent successive configurations of M, with the bottom row being the initial configuration of M on w. (Note that the rows must be infinitely long.) You will need tiles that simply copy tape symbols from one configuration to the next, and tiles that implement left and right head moves. This will be easier if you represent the head position by a composite symbol (q, a) on a single tile. Use the top and bottom labels of tiles to transfer information from one configuration to the next, and the left and right labels to make sure that the head moves are correctly handled. You will need some special tiles to set up the initial configuration.]

[continued on next page]

3. A function $f : \Sigma^* \rightarrow \Sigma^*$ is a *computable function* if there is a Turing machine that, on each input $w \in \Sigma^*$, halts with just $f(w)$ on its tape.

(a) Show, using a countability argument, that there exist functions from $\{0,1\}^*$ to $\{0,1\}^*$ that are not computable.

(b) If f is a computable function, show that the range of f is Turing recognizable.

[NOTE: The *range* of a function f is its set of possible values, i.e., the set of strings z such that $f(w) = z$ for some $w \in \Sigma^*$.]

(c) Give an explicit example of a computable function f whose range is not decidable.

[HINT: Let L be any Turing recognizable language that is not decidable. Can you construct a computable function f whose range is L?]