Homework 2

Due: 5:00pm, Thursday February 2

Remarks. In this homework, unless otherwise specified, you should describe Turing Machine algorithms at the level of detail Sipser uses in his Turing Machine descriptions (the ones that are in quotation marks and typically span a paragraph of half-a-dozen lines). Also, you may well find it useful to cite and use the theorem from class that a multitape Turing Machine running in time T can be simulated by a 1-tape Turing Machine running in time $O(T^2) + O(n)$.

1. **(Unary undecidability.)** (10 points.) A "unary language" (also called a "tally language") is a language over the alphabet $\{1\}$. (In other words, a unary language is any subset of $\{1\}^*$.) Show that there is an undecidable unary language.

   (I would like your solution to be relatively short, so for this problem you don't have to describe any Turing Machines in any detail. Instead, if you need to, you may "appeal to the Church–Turing Thesis"; i.e., if there is an algorithmic task that is pretty obviously doable by an algorithm, you may simply state that it can be done by a Turing Machine.)

2. **(An alarm clock.)**

   (a) (5 points.) Describe a 2-tape Turing Machine with the following behavior. On input string w, the machine should print the string 1^{2n^2} on its second tape, where $n = |w|$. That is, it should print out the string of length $2n^2$ consisting entirely of the symbol 1. (There's nothing overly special about $2n^2$ here; I just picked a moderately simple expression.) When your Turing Machine halts, the first tape should still just contain the string w, and both tape heads should be returned to their initial locations.

   (b) (5 points.) Suppose M is some 1-tape Turing Machine. Show that there is another 1-tape Turing Machine M' with the following property: On every input string w, if $M(w)$ accepts within $2|w|^2$ time steps, then $M'(w)$ accepts; and, if $M(w)$ fails to accept within $2|w|^2$ time steps, then $M'(w)$ rejects. Furthermore, M' should run in polynomial time.

   (Finally, you should convince yourself that there is a polynomial-time algorithm that, on input $\langle M \rangle$, outputs $\langle M' \rangle$. I decided it was too annoying to ask you to actually prove this, but as I say, you should convince yourself of it.)

3. **(Closure properties.)** In this problem, we fix some alphabet Σ and assume all languages called L or L_i are subsets of Σ^*.

   (a) (1 point.) Prove that the complexity class P is closed under complement. (That is, show that if $L \in \mathsf{P}$ then $L^c \in \mathsf{P}$, where $L^c = \Sigma^* \setminus L$.)

   (b) (3 points.) Prove that P is closed under union; i.e., if $L_1, L_2 \in \mathsf{P}$ then $L_1 \cup L_2 \in \mathsf{P}$.

   (c) (1 point.) Prove that P is closed under intersection; i.e., if $L_1, L_2 \in \mathsf{P}$ then $L_1 \cap L_2 \in \mathsf{P}$.

   (d) (1 point.) Prove that P is closed under "Majority-of-3"; i.e., if $L_1, L_2, L_3 \in \mathsf{P}$ then the following language is in P:

   $$L = \{w : w \text{ is in at least two out of } L_1, L_2, L_3\}.$$

(e) (4 points.) Prove that if $L \in \mathsf{P}$, then the following language is also in P:

$$K = \{\#w_1\#w_2\#w_3\#\cdots\#w_m : m \in \mathbb{N} \text{ and } w_1, \ldots, w_m \in L\}.$$

(Here the w_i's are strings in Σ^* and # is some symbol not in Σ. Thus, in contrast to previous problems, our language of concern K is a subset of $(\Sigma \cup \{\#\})^*$.)

4. **(Running times less than n.)** (10 points.) Show that $\mathrm{TIME}(\sqrt{n}) = \mathrm{TIME}(1)$.

(Remarks: The only thing special about $\sqrt{n}$ here is that it's noticeably less than n. I could have put $n^{.999}$ and had the same result, but I put $\sqrt{n}$ for simplicity. For this problem you will want to make sure you have a clear understanding of big-O notation: "$f(n)$ is $O(g(n))$" means there exist positive constants C and n_0 such that $f(n) \leq Cg(n)$ for all $n \geq n_0$.)