CS 121 Homework 5: Fall 2020

Some policies: (See the course policy at `http://madhu.seas.harvard.edu/courses/Fall2020/policy.html` for the full policies.)

- **Collaboration:** You can collaborate with other students that are currently enrolled in this course (or, in the case of homework zero, planning to enroll in this course) in brainstorming and thinking through approaches to solutions but you should write the solutions on your own and cannot share them with other students.
- **Owning your solution:** Always make sure that you "own" your solutions to this other problem sets. That is, you should always first grapple with the problems on your own, and even if you participate in brainstorming sessions, make sure that you completely understand the ideas and details underlying the solution. This is in your interest as it ensures you have a solid understanding of the course material, and will help in the midterms and final. Getting 80% of the problem set questions right on your own will be much better to both your understanding than getting 100% of the questions through gathering hints from others without true understanding.
- **Serious violations:** Sharing questions or solutions with anyone outside this course, including posting on outside websites, is a violation of the honor code policy. Collaborating with anyone except students currently taking this course or using material from past years from this or other courses is a violation of the honor code policy.
- **Submission Format:** The submitted PDF should be typed and in the same format and pagination as ours. Please include the text of the problems and write **Solution X:** before your solution. Please mark in gradescope the pages where the solution to each question appears. Points will be deducted if you submit in a different format.
- **Late Day Policy:** To give students some flexibility to manage your schedule, you are allowed a net total of **eight** late days through the semester, but you may not take more than **two** late days on any single problem set. No exceptions to this policy.

By writing my name here I affirm that I am aware of all policies and abided by them while working on this problem set:

Your name: (Write name and HUID here)

Collaborators: (List here names of anyone you discussed problems or ideas for solutions with)

No. of late days used on previous psets (not including Homework Zero):
No. of late days used after including this pset:

Questions

Please solve the following problems. Some of these might be harder than the others, so don't despair if they require more time to think or you can't do them all. Just do your best. Also, you should only attempt the bonus questions if you have the time to do so. If you don't have a proof for a certain statement, be upfront about it. You can always explain clearly what you are able to prove and the point at which you were stuck. You can always simply write **"I don't know"** and you will get 15 percent of the credit for the problem. If you are stuck on this problem set, you can use Piazza to send a private message to all staff.

Note on reading the textbook: If you are stuck on some of the problems, try consulting the book to 1) understand the concepts the question is referencing, and 2) review the way similar theorems are proved in the book.

Problem 1:

Problem 1.1 (4 points): Prove that the function $f_1 : \{0,1\}^* \to \{0,1\}$ for which $f_1(x) = 1$ iff $x = 1001111$ is computable.

Problem 1.2 (6 points): Prove that the function $f_2 : \{0,1\}^* \to \{0,1\}$ for which $f_2(x) = 1$ iff x is an encoding of a Turing Machine that calculates f_1 is uncomputable.

Problem 1.3 (5 points): Prove or disprove: There exists a constant C and an uncomputable function $f : \{0,1\}^* \to \{0,1\}$ such that the number of values of x for which $f(x) = 1$ is at most C.

Problem 2: For each of the following problems, write either "can be proven uncomputable by Rice's theorem" (i.e. the function is semantic and nontrivial) or "can't be proven uncomputable by Rice's theorem" for 2 points (no justification necessary for this part). Then, write either "computable" or "uncomputable", with brief justification, for a variable number of points. (If you answered "can be proven uncomputable by Rice's theorem", justification is why the function's semantic and nontrivial.)

Problem 2.1 (2+3 points): $f(M) = 1$ iff there is an input x such that $M(x) = 1$.

Problem 2.2 (2+3 points): $f(M) = 1$ iff any of M's instructions include "move left".

Problem 2.3 (2+Bonus 0 points): $f(M) = 1$ iff there exists an input for which M ever moves left.

Problem 2.4 (2+4 points): $f(M, n) = 1$ iff it is the case that for every input of length n, M runs in time at most 2^n.

Problem 2.5 (2+Bonus (hard) 0 points): $f(M) = 1$ iff it is the case that for every integer n and every input of length n, M runs in time at most 2^n.

Problem 2.6 (2+3 points): $f(M) = 1$ iff it is the case that for every x, $M(x) = 1$ if and only if $x = x^R$. (x^R denotes the reverse of x.)

Problem 3:

Problem 3.1 (5 points): The "configuration" of a Turing Machine at a time t when it is in the middle of a computation is information that determines its future operation. Specifically, for a machine M with k states and alphabet Σ, its configuration is the triple (q, i, x) where $q \in [k]$ is its current state, $i \in \mathbb{N}$ is the position of the tape head, and $x \in \Sigma^*$ is the string of cells on the tape between $\triangleright$ and the first ϕ. (Assume the TM never writes a ϕ or overwrites any but the leftmost ϕ.) Prove that if, on some input w, a machine M reaches the same configuration twice, i.e. $C_1 = (q_1, i_1, x_1)$ if the configuration at time t_1 and $C_2 = (q_2, i_2, x_2)$ is the configuration at time t_2 and $C_1 = C_2$, then M never halts on input w.

Problem 3.2 (15 points): Let SPACE-HOG$(M, x, s) = 1$ if M is a Turing Machine that on input x uses more than s bits of space (i.e., tape head ever reaches the $s + 1$th cell of the tape). Prove that SPACE-HOG is computable.

Problem 4 (20 points): Recall that HALT$(M, x) = 1$ if M is the encoding of a Turing Machine and M halts on x, and HALT$(M, x) = 0$ otherwise.

For integer C, let HALT$_C(M, x) = 1$ if and only if both HALT$(M, x) = 1$ and M is a Turing Machine with at most C states.

Prove that there exists a constant C such that HALT$_C$ is uncomputable.

Problem 5 (20 points): Let $f(M) = 1$ if M is a Turing Machine that ever writes the symbol @ on its tape on input the empty string. Let $f(M) = 0$ in all other cases. Show that $f(M)$ is uncomputable. (Hint: reduce from the halting problem.)

Problem 6:

Problem 6.1 (Bonus, 0 points): Give an example of functions f, g, F, G such that $f(n) = O(F(n))$, $g(n) = O(G(n))$ but $f(g(n)) \notin O(F(G(n)))$.

Problem 6.2 (10 points): Prove that if $f(n) = O(n^c)$ and $g(n) = O(n^d)$ for $c, d > 0$ then $f(g(n)) = O(n^{cd})$.

The contrast between 6.1 and 6.2 may mean that the following problem, which you solved in sections, was harder than you thought

Problem 6.3 (Bonus, 0 points): Prove that if $A : \{0,1\}^* \to \{0,1\}^*$ and $B : \{0,1\}^* \to \{0,1\}^*$ are polynomial time computable functions then their composition $A(B(x))$ is also polynomial time computable.