College of Computer & Information Science — Spring 2009
Northeastern University — Handout 9
CSG714: Theory of Computation — 9 February 2009

Problem of the Week – 5

Reductions and NP

This problem concerns two questions that came up in last class's discussions. First was whether there exist NP-hard problems that are not in NP. The answer is certainly yes, since there are problems in much higher complexity class to which every language in NP (many-one) reduces in polynomial time. In fact, as an extreme, we can take an undecidable problem, say the Halting Problem.

(a) Show that the Halting Problem is NP-hard.

We also discussed the difference between many-one reductions (also called Karp or mapping reductions, as defined in Sipser) and Turing reductions (also called Cook reductions). Let us consider many-one or mapping reductions first.

(b) Show that NP is closed under polynomial-time many-one reductions. That is, show that if L_2 is in NP and there is a polynomial-time many-one reduction from L_1 to L_2, then L_1 is in NP.

We say that there is a polynomial-time *Turing reduction* from language L_1 to language L_2 if given a Turing machine M_2 for deciding L_2 as an oracle (black box), we can construct a Turing machine M_1 that decides L_1 by making at most a polynomial number of calls to a given Turing machine that decides L_2.

(c) Show that P is closed under polynomial-time Turing reductions. Why does not a similar proof extend to NP?