6.892: Algorithmic Lower Bounds, Spring 2019

Prof. Erik Demaine, Jeffrey Bosboom, Jayson Lynch

Problem Set 6 Solution

Due: Tuesday, March 19, 2019 at noon

Problem 6.1 [Reduction Properties]. Suppose you have a polynomial-time one-call reduction r from NP search problem A to NP search problem B, which converts each problem-A instance x having a solutions into a problem-B instance $r(x)$ having $f(x, a)$ solutions.

(a) What must be true about f for r to preserve NP-hardness (i.e., A being NP-hard implies B is NP-hard)?

Solution: $f(x, a) > 0$ if and only if $a > 0$.

(b) What must be true about f for r to preserve #P-hardness (i.e., $\#A$ being #P-hard implies $\#B$ is #P-hard)?

Solution: The number of solutions a can be computed in polynomial time from x and $f(x, a)$. That is, there is a polynomial-time function g such that $g(x, f(x, a)) = a$.

(c) What must be true about f for ASP A being NP-hard to imply ASP B is NP-hard?

Solution: If $a = 1$, then $f(x, a) = 1$, and if $a > 1$, then $f(x, a) > 1$.

(d) Given an NP search problem A, define *AASP A* to be the following decision problem: given an instance of A and two distinct solutions of that instance, is there a third distinct solution? What must be true about f for AASP A being NP-hard to imply AASP B is NP-hard?

Solution: If $a = 2$, then $f(x, a) = 2$, and if $a > 2$, then $f(x, a) > 2$.

(e) Besides the function r converting instances of A into instances of B, we require another function to conclude that ASP B is NP-hard. What must the other function do? What about for AASP?

Solution: For ASP, we need a polynomial-time function g such that, if y is a solution to x, then $g(x, y)$ is a solution to $r(x)$.

For AASP, we need a polynomial-time function g such that, if y_1 and y_2 are distinct solutions to x, then $g(x, y_1, y_2)$ is a pair of distinct solutions to $r(x)$.

Variation where we want to actually find other solutions, instead of just decide whether there is another solution:

For ASP, we need a pair of polynomial-time functions g and h such that

- If y is a solution to x, then $g(x, y)$ is a solution to $r(x)$.
- If y' is a solution to $r(x)$, then $h(x, y')$ is a solution to x.
- Function h is injective: if $y'_1 \neq y'_2$ are solutions to $r(x)$, then $h(x, y'_1) \neq h(x, y'_2)$.

For AASP, we need these conditions and also that function g is injective: if $y_1 \neq y_2$ are solutions to x, then $g(y, p_1) \neq g(y, p_2)$.