CS303 (Spring 2008)— Solutions to Assignment 10

Problem 1

(a) A problem is a set (of numbers, strings, ...).

(b) A program P decides a problem X if given an input y, P returns 1 (or "Yes") whenever $y \in X$, and 0 (or "No") whenever $y \notin X$.

(c) A reduction from X to Y is a function f such that for every input x: if $x \in X$, then $f(x) \in Y$, and if $x \notin X$, then $f(x) \notin Y$.

(d) When we write $X \leq_m Y$, we mean that there is a reduction f from X to Y, which is itself total and recursive, i.e., is computed by a program which always terminates.

(e) If we have proved $X \leq_m Y$, then we have proved that if Y is decidable, so is X. Thus, either both are decidable, neither is decidable, or X is decidable while Y is not. We have ruled out that Y is decidable and X is not.

Problem 2

We will prove this by proving that HALT $\leq_m$ NONEMPTY. A reduction f from HALT to NONEMPTY would need to take a program P (input to HALT) and transform it to a program $f(P)$ (input to NONEMPTY), such that if $P(P) \downarrow$, then $f(P)$ terminates for at least one input x, and if $P(P) \uparrow$, then $f(P)$ does not terminate for any inputs x.

Here is what the reduction does: given a program P, it produces the following new program $f(P)$. $f(P)$ obtains an input x, but ignores it. It executes $P(P)$. If and when $P(P)$ terminates, it returns 0. First, we check that given the source code of P, we can clearly produce the source code of $f(P)$. If $P(P) \downarrow$, then $f(P)$ makes it through that part and terminates on each input. If $P(P) \uparrow$, then $f(P)$ will always get stuck in that part, regardless of the input, so it will never terminate. Thus, f is a correct reduction, and NONEMPTY is undecidable.

Problem 3

Similar to Problem 2, we will reduce from HALT to VARIED. We will produce a reduction f, which takes a program P (input to HALT) and transform it to a program $f(P)$ (input to VARIED), such that if $P(P) \downarrow$, then $f(P)$ outputs at least two different values, and if $P(P) \uparrow$, then $f(P)$ does not output at least two different values.

The reduction, given a program P, produces a program $f(P)$. That program gets as an input x. It first runs $P(P)$. If and when that terminates, it returns the value x itself. Again, this can clearly be computed by a total program. Now, if $P(P) \downarrow$, then for each input x, $f(P)$ will output x. In particular, it will output infinitely many different values, so at least two. On the other hand, if $P(P) \uparrow$, then $f(P)$ will not terminate on any input. In particular, it cannot output two different values. Thus, f is a correct reduction, and VARIED is undecidable.

Problem 4

We want to define a reduction from X to Y. Given a program P, we want to produce a program $f(P)$, such that if $P(42) \downarrow$ and $P(y) \uparrow$ for all $y \neq 42$, then $f(P)$ terminates for infinitely many inputs, but if $P(42) \uparrow$ or $P(y) \downarrow$ for some $y \neq 42$, then $f(P)$ terminates for only finitely many inputs. This is a bit more tricky, as *non-termination* of P should imply termination of $f(P)$.

The program $f(P)$, given input x, behaves as follows: first it runs $P(42)$. If and when $P(42)$ terminates, it runs each of $P(1), P(2), P(3), \ldots, P(x)$ for x steps (except $P(42)$, which we don't run). This will take $O(x^2)$ simulation steps (x steps each for $x - 1$ programs). If any one of these runs of P terminates in x steps, then $f(P)$ deliberately enters an infinite loop. Otherwise, it returns some value, such as 0.

First, we should convince ourselves that f can be computed. Notice that it just simulates P for a limited number of steps, so all we need to do is augment P by a counter or timer, and add a `for` loop. So f is total and recursive.

Now, if $P \in X$, then $P(42) \downarrow$, and $P(y) \uparrow$ for all $y \neq 42$. So the first step, running $P(42)$, will terminate for all inputs y. For the second part, because $P(y) \uparrow$ for all y, none of the simulations will terminate within the allotted x steps, for any x. So the second part will never enter the infinite loop. Thus, $f(P)$ will terminate on all inputs x, in particular on infinitely many.

On the other hand, if $P \notin X$, then either $P(42) \uparrow$, or $P(y) \downarrow$ for some $y \neq 42$. In the first case, running $P(42)$ will never terminate, so $f(P)$ will not terminate on any input. In particular, it will not terminate on infinitely many inputs. On the other hand, if $P(y) \downarrow$ for some $y \neq 42$, let t be the number of steps that $P(y)$ takes to terminate for that particular y. Then, for each $x \geq \max(t, y)$, $f(P)$ will find that $P(y)$ terminates within x steps, and will deliberately enter an infinite loop. Therefore, $f(P)$ will terminate at most on inputs $1, \ldots, \max(t, y)$, which is only finitely many. Thus, $f(P) \notin Y$.

This completes the proof that the reduction is correct, and $X \leq_m Y$.