Lab12-Solution

CS363-Computability Theory, Xiaofeng Gao, Spring 2016

1. A *dominating set* for a graph $G = (V, E)$ is a subset D of V such that every vertex not in D is adjacent to at least one vertex in D. The domination number $\gamma(G)$ is the number of vertices in a smallest dominating set for G. The *Dominating Set* (DS) problem concerns finding a minimum $\gamma(G)$ for a given graph G.

   Prove that: SET-COVER$\equiv_p$ DOMINATING-SET.

   Solution. Assume the sets in SET-COVER problem are $S_1, \dots, S_n$ where $S_1 \cup \dots \cup S_n = \{e_1, e_2, \dots, e_m\}$.

   - SET-COVER $\leq_p$ DOMINATING-SET
     (a) Firstly, we define two vertices sets V_S and V_e. We add a vertex v_{S_i} into V_S for each set S_i and add a vertex v_j into V_e for each element e_j.
     (b) Secondly, when the set S_i contains a element e_j, we add a edge $\{v_{S_i}, v_j\}$ into the edge set E.
     (c) Finally, add edges $\{v_{S_i}, v_{S_j}\}$ for each two different vertices pair in V_S. And we get a new graph $G = (V = (V_S \cup V_e), E)$.

     Considering an optimal solution OPT for the DOMINATING-SET problem in the constructed graph. If it contains a vertex u in V_e, we can see that the set consists with its neighbors must be a subset of V_S and none of its element is contained in the OPT. Then we can replace u by any element among its neighbors and get another optimal solution. We can see that there exists a solution with k sets in SET-Cover problem iff there exists a solution contains k vertices in the DOMINATING-SET. Thus SET-COVER $\leq_p$ DOMINATING-SET.
   - DOMINATING-SET $\leq_p$ SET-COVER
     (a) For each vertex u, we define a set S_u and an element e_u.
     (b) For each vertex u, if a vertex v(contains u itself) is dominated by u, we add e_v to S_u.

     It is obvious that there exists a solution contains k vertices in the DOMINATING-SET if there exists a solution with k sets in SET-Cover problem. Thus DOMINATING-SET $\leq_p$ SET-COVER.

   $\square$

2. Let A, B, C, be sets. Prove that
   (a) If A is B-recursive and B is C-recursive, then A is C-recursive.
   (b) If A is B-r.e. and B is C-recursive, then A is C-r.e.
   (c) If A is B-recursive and B is C-r.e., then A is not necessarily C-r.e.

   Solution.

   (a) As it is mentioned in the problem, $c_A \in \mathscr{C}^{c_B}$ and $c_B \in \mathscr{C}^{c_C}$.Then whenever we need to use $c_B(x)$ we can always build a $c_C - program$ for it. Therefore, we know that $c_A \in \mathscr{C}^{c_C}$ then c_A is c_C computable, i.e. A is C-recursive.

(b) Similar to the solution for (a), we can deduce that $c_A \in \mathscr{C}^{c_C}$ then c_A is c_C computable,i.e. A is C-recursive.

(c) Notice that A is B-recursive means we need to know both whether $x \in B$ or $x \notin B$. But B is C-r.e. can only tell us whether $x \in B$. So A is not necessarily C-r.e.

$\square$

3. Let A, B be any sets.

(a) Show that $A \leq_T B$ iff $K^A \leq_m K^B$, and $A \equiv_T B$ iff $K^A \equiv_m K^B$.

(b) Show that the previous question can be made effective in the following sense: there is a total computable function f such that if $c_A = \phi_e^B$, then $\phi_{f(e)} : K^A \leq_m K^B$. (*Hint.* Find total computable functions g, h such that (1) if $c_A = \phi_e^B$ then $K^A = W_{g(e)}^B$, (2) $\phi_{h(e)} : W_e^B \leq_m K^B$ for all e.)

Solution.

(a)
- "$\Rightarrow$": If $A \leq_T B$, then since K^A is A-r.e. it is also B-r.e.. Define

$$f(x,y) = \begin{cases} 1 & \text{if } x \in K^A \\ \uparrow & \text{otherwise} \end{cases}$$

  Since K^A is B-r.e. f is B-recursive. According the relativised s-m-n theorem, there is a total computable function k such that $f(x,y) \simeq \phi_{k(x)}^B(y)$.

$$x \in K^A \Rightarrow \phi_{k(x)}^B(k(x)) = 1 \Rightarrow k(x) \in K^B$$

$$x \notin K^A \Rightarrow \phi_{k(x)}^B(k(x)) \text{ is undefined} \Rightarrow k(x) \notin K^B$$

  Therefore $K^A \leq_m K^B$.
- "$\Leftarrow$": Since K^B is B-r.e., there exists a B-computable g function such that $K^B = Dom(g)$. Suppose we have a total computable function h: $K^A \leq_m K^B$, then $K^A = Dom(h \circ g)$. $h \circ g$ is a B-computable function therefore K^A is B-r.e. and $A \leq_T B$.

Now we have proved that $A \leq_T B$ iff $K^A \leq_m K^B$. $A \equiv_T B$ iff $K^A \equiv_m K^B$ follows directly.

(b)
- Define

$$f(x,y) = \begin{cases} 1 & \text{if } y \in K^{\phi_x^B} \\ \uparrow & \text{otherwise} \end{cases}$$

  f is B-computable, then according to the relativised s-m-n theorem, there exists a total computable function g such that $f(x,y) \simeq \phi_{g(x)}^B(y)$. If $c_A = \phi_e^B$, then

$$\phi_{g(e)}(y) = \begin{cases} 1 & \text{if } y \in K^{c_A} \\ \uparrow & \text{otherwise} \end{cases}$$

  Therefore we have $K^A = W_{g(e)}^B$.
- ϕ_e^B is a way to enumerate all the B-computable function, and thus W_e^B is an enumeration of all the B-r.e. sets. Therefore, consider a computable function $f^B(e,x,y)$ defined by

$$f^B(e,x,y) = \begin{cases} 1, & x \in W_e^B, \\ \uparrow, & \text{otherwise.} \end{cases}$$

According to the relativised s-m-n theorem, there are total computable functions s and h such that $f^B(e,x,y) \simeq \phi^B_{s(e,x)}(y) \simeq \phi^B_{\phi_{h(e)}(x)}(y)$. Then $x \in W^B_e \Leftrightarrow f^B$ is defined $\Leftrightarrow \phi^B_{\phi_{h(e)}(x)}(\phi_{h(e)}(x))$ is defined $\Leftrightarrow \phi_{h(e)}(x) \in K^B$. Therefore, we have $\phi_{h(e)} : W^B_e \leq_m K^B$.

Combine the 2 parts together we can get the final solution: $\phi_{h(g(e))} : K^A \leq_m K^B$.

$\square$