Assignment 4 solution

Netanel Albert

May 9, 2020

Problem 1:

We want to show that given an algorithm that determine if some graph has a 3-coloring in polynomial time, SEARCH-3-COL can also be solved in a polynomial time.

Let A be an algorithm determining 3-COL in polynomial time. That is given a graph G, $A(G) = 1$ if $G \in$ 3-COL and 0 otherwise. The following algorithm resolves SEARCH-3-COL in polynomial time. Let G be a given instance. We will write $G[x + y]$ to donate the graph we get by marge the vertices x and y.

1. If $A(G) = 0$, then stop and reject the input.
2. Let $v \in V(G)$ be an arbitrary vertex.
3. For each $u \in G$ s.t. $u \notin N_G(v)$ do:
4. If $A(G[v + u]) = 1$, then
5. (a) Set $\varphi(u) = 3$.
6. (b) Set $G = G[v + u]$.
7. end-if.
8. enf-for.
9. Set $G = G \setminus v$.
10. Color the remaining vertices with the colors $1, 2$ (it's easy because the graph is now bipartite).
11. . Return φ.

■

To show that some language $L \in NP$ we need to show:
a. A verification algorithm in polynomial time to show that $L \in NP$.
b. $\forall L_i \in NP : L_i \leq L$ and it's enough to find some $L^* \in NPC\ s.t.\ L^* \leq L$.
And for b. we need a polynomial time reduction from L^* to L. We use those a,b in the next 3 proves.

Problem 2:

a. Given a pair (G, k) and legal coloring ψ return true if $\sum_{v \in V(G)} \psi(v) \leq k$, false otherwise.

b. Let $L_2 := \{(G, k) : \sigma(G) \leq k\}$. We'll prove that $L \in NPC$ by showing $3 - COL \leq L_2$ Given graph G we'll build new graph G' as follow, $V(G') = \{v_i : v \in V(G), i \in [3]\}$ and $E(G') = \{\{v_i, v_j\} : i < j \in [3]\} \cup \{\{v_i, u_i\} : \{v, u\} \in E(G), i \in [3]\}$. In words, every vertex in G become a triangle in G'.
The building time is polynomial because we need $3 * v(G)$ vertices and $3 * v(G) + 3 * e(G)$ edges.
Now we need to prove:

$$G \text{ has a 3-coloring} \Leftrightarrow \sigma(G') \leq 6 * v(G)$$

If G has a 3-coloring ψ so the coloring $\psi'(v_i = \psi(v) + i(mod\, 3)$ is legal for G' and $\sigma_{\psi'}(G') = 6 * v(G)$.
If $\sigma(G') \leq 6 * v(G)$, note that $\sigma(G') \geq 6 * v(G)$ for every valid coloring, because we have $v(G)$ triangles and each one has to use 3 different colors, so the minimum is $1, 2, 3$ and $1 + 2 + 3 = 6$. therefore $\sigma(G') = 6 * v(G)$

and it follows that G' has a 3-coloring, otherwise at least one of the triangles sum was bigger than 6 and we would get $\sigma(G') > 6 * v(G)$. Let ψ' be that coloring so $\psi(v) = \psi'(v_1)$ is a valid 3-coloring for G ■

Problem 3:

Lemma: Given graphs G and $G' := G \cup \{v', v'u : u \in V(G)\}$ i.e. we add a new vertex and connect it to each vertex in G. So G has a $k - CLIQUE \Leftrightarrow G'$ has a $(k+1) - CLIQUE$.
Prove: If G has a $k - CLIQUE\ \ K$ so in G', $K \cup v'$ is a $(k+1) - CLIQUE$. If G' has a $(k+1) - CLIQUE\ \ K$ so for any vertex $x \in V(K')$, $K' \setminus x$ is a $k - CLIQUE$. If $v' \in V(K')$ take $x = v'$, otherwise just take a random one. ■

a. Given graph G and clique $K \subseteq G$ return true if $v(K) \geq \sqrt{v(G)}$, false otherwise.

b. Let $L_3 := \{G : \omega(G) \geq \sqrt{v(G)}\}$. We'll prove that $L \in NPC$ by showing $k - CLIQUE \leq L_3$.
Given (G, k) Case 1. $k > \sqrt{v(G)}$ so add some isolated vertices until $k = \sqrt{v(G)}$ (This number will be $k^2 - v(G)$).
Case 2. $k < \sqrt{v(G)}$ so add 1 vertex and connect it to each vertex in G, and set $k = k + 1$. Keep doing it until $k \geq \sqrt{v(G)}$. go back to case 1 if need.
Case 3. $k = \sqrt{v(G)}$ do nothing.
Now name the new graph G' and we need to prove:

$$(G, k) \in k - CLIQUE \Leftrightarrow G' \in L_3.$$

Prove: Case 1: adding isolated vertices can't create or destroy a complete subgraph. Case 2: we showed in the lemma. Case 3: trivial. ■

Problem 4:

a. Given a Boolean formula φ and two assignments a, b Return $\varphi(a) \wedge \varphi(b)$.

b. Let $L_4 := \{\varphi : \varphi$ is a CNF-formula admitting at least two satisfying assignments$\}$. We'll prove that $L_4 \in NPC$ by showing $SAT \leq L_4$.
Given a Boolean formula φ, define $f(\varphi) = \varphi \wedge (x \vee \neg x)$. It's easy to see that f is computed in polynomial time (actually is constant). Now, if $\varphi \in SAT$ so it has an assignments I that satisfying it, hence $I \cup \{x\}, I \cup \{\neg x\}$ satisfying $f(\varphi)$ so $f(\varphi) \in L_4$. Else, φ is always false, so $f(\varphi) = \varphi \wedge \textit{something}$ so it's also false, so it doesn't have even 1 satisfying assignments so $f(\varphi) \notin L_4$ ■.