Note #1 on Chapter 11 of [Manber]: An NP-Completeness Proof

This note concerns the NP-completeness proof of the *dominating set problem* in Manber's book. The main purpose is to make clearer certain conditions that are omitted or implicitly assumed in the book. With the proof as an example, we also wish to clarify how the definition of polynomial-time reduction is followed in a typical NP-completeness proof. We start with the problem statement; other related definitions are appended at the end of this note.

The Dominating Set Problem: Given an undirected graph $G = (V, E)$ and an integer k, determine whether G has a dominating set containing $\leq k$ vertices. (A *dominating set* D of G is a subset of V such that every vertex of G is either in D or is adjacent to some vertex in D.)

Theorem. The dominating set problem is NP-complete.

Proof. The problem is obviously in NP, as we can guess a set of vertices and check in polynomial time whether the set is of size $\leq k$ and is indeed a dominating set of the given graph G. To prove that it is NP-hard, we demonstrate a polynomial-time reduction from the vertex-cover problem, which is known to be NP-hard. An input $(G_1 = (V_1, E_1), k_1)$, which is a pair of a graph and an integer, to the vertex cover problem can be converted to an input $(G_2 = (V_2, E_2), k_2)$ to the dominating set problem in the following manner:

To obtain G_2, we first remove all isolated vertices (which are not connected to any other vertex) from V_1. We then add, for each edge $\{u, v\}$ in E, a vertex uv and two edges $\{u, uv\}$ and $\{uv, v\}$. In other words, we transform every edge into a triangle. Finally, we make k_2 simply equal to k_1. This conversion apparently can be done by a deterministic algorithm in polynomial time.

We need to show that G_1 has a vertex cover of size $\leq k_1$ if and only if G_2 has a dominating set of size $\leq k_2$. But before doing so, we deviate to make a contrast with the definition of polynomial-time reduction (which can be found in the appendix). The input spaces U_{vc} and U_{ds} of the two problems are the same, namely the set of all possible pairs of a graph and an integer. The language L_{vc} of the vertex cover problem is the set of all (G, k) such that G has a vertex cover of size $\leq k$, while the language L_{ds} of the dominating set problem is the set of all (G, k) such that G has a dominating set of size $\leq k$. The proof obligation "G_1 has a vertex cover of size $\leq k_1$ if and only if G_2 has a dominating set of size $\leq k_2$" is derived from the statement "$(G_1, k_1) \in L_{vc}$ iff $(G_2, k_2) \in L_{ds}$". (End of Deviation)

The "only if" part: Suppose G_1 has a vertex cover C of size $\leq k_1$. Remove all isolated vertices in C to obtain another vertex cover C' of G_1 (isolated vertices are not usual for covering an edge).

C' is also a subset of V_2 and $|C'| \leq |C| \leq k_1 = k_2$. We claim that C' is a dominating set of G_2. Every vertex u in V_2 that comes from V_1 is an end vertex of some edge $\{u, v\} \in E_1$. Since $\{u, v\}$ is covered by C', either u or v must be in C', implying that u is dominated by C', i.e., u is either in C' or adjacent to some vertex (namely v) in C'. Every new vertex uv that was added for edge $\{u, v\}$ is adjacent to both u and v and is also dominated, as again one of u and v must be in C'.

The "if" part: Suppose G_2 has a dominating set D of size $\leq k_2$. D may not be a subset of V_1, as D may contain vertices that were added in the conversion. Replace every vertex uv in D, which was added for edge $\{u, v\}$, by either u or v to obtain a new set D'. Since every replaced vertex is adjacent to the replacing vertex, D' remains a dominating set of G_2. D' is a subset of V_1 and $|D'| \leq |D| \leq k_2 = k_1$ ($|D'|$ is not necessarily equal to $|D|$). We claim that D' is also a vertex cover of G_1. For every edge $\{u, v\}$ in E_1, either u or v is in D'; otherwise, the added vertex uv in V_2 corresponding to $\{u, v\}$ would not be dominated by D'. Therefore, every edge of G_1 is covered by D'. □

Appendix

The Vertex Cover Problem: Given an undirected graph $G = (V, E)$ and an integer k, determine whether G has a vertex cover containing $\leq k$ vertices. (A *vertex cover* C of G is a subset of V such that every edge in G is incident to at least one vertex in C.)

Polynomial-Time Reduction: Let L_1 and L_2 be two languages from the input spaces U_1 and U_2. We say that L_1 is *polynomially reducible* to L_2 if there exists a polynomial-time algorithm that converts each input $u_1 \in U_1$ to another input $u_2 \in U_2$ such that $u_1 \in L_1$ if and only if $u_2 \in L_2$.