# The Theory of Computation
## Bernard M.E. Moret
## Addison-Wesley-Longman, 1998

## *Solution Manual*
## *Chapter 7, Problem 7.22*

This problem is sometimes called *Perfect Dominating Set (PDS)* or also *1-Perfect Codes on Graphs*. We prove it NP-complete (it is obviously in NP) by transforming *Positive 1in3SAT* to it.

Given an instance of *Positive 1in3SAT* with variables $V = \{x_1, \ldots, x_n\}$ and clauses $C = \{\{x_i, y_i, z_i\}, 1 \leq i \leq r\}$, we construct an instance of *PDS* with $cn + r$ vertices and $n + 3r$ edges as follows. Each variable will be represented as 2 vertices connected by an edge; one of the vertices is designated as the attaching point. Each clause will be represented by one vertex. The composition of a clause is denoted by edges (three for each clause vertex) connecting the clause vertex to the attaching point of corresponding variable components.

Now, a clause vertex cannot be in $V'$: putting it in $V'$ excludes its three literals, but then the other vertex of each literal component won't be dominated by anything. Thus all $r$ clause vertices must be in $V - V'$. In order to dominate them, exactly one variable attaching point per clause must be in $V'$; such variables are considered true and all others are considered false. The remaining vertex is in the dominating set if the variable is false, so that the variable's attaching point is dominated. It should now be obvious that a solution to *PDS* can easily be constructed from a satisfying truth assignment. This completes our proof.

From the standard *1in3SAT*, we use the same construction, where now the other end of each variable segment is the attaching point for complemented literals; we connect all clause vertices together in a complete subgraph. The latter prevents us from selecting any clause vertex for the dominating set, since it would allow only one and prevent domination of the variable vertices corresponding to the complements of the literals appearing in the selected clause.