The Theory of Computation

Bernard M.E. Moret

Addison-Wesley-Longman, 1998

Solution Manual
Chapter 7, Problem 7.29

We prove this problem NP-complete (it is obviously in NP) by transforming *NAE3SAT* to it. The proof is a local replacement—with simple pieces for all but the truth-setting component. For each clause, we set up a simple triangle, where vertices correspond to literals. Cutting each triangle corresponds to setting at least one literal to true and one to false in each clause. For each variable, we set up a truth-setting component with two connecting vertices (corresponding to the uncomplemented and the complemented literals). The problem here is to ensure that these two vertices be placed on opposite sides of the partition; this is not as easy as it seems: see below for a suitable graph piece. Finally, we connect each clause literal to the corresponding attaching point in the truth-setting component; the connection piece ensures that its two extremities be placed on the same side of the partition. A suitable piece consists of a triangle and the two extremities, where each extremity is connected to every vertex of the triangle. (Notice that this is just a K_5 with one missing edge—between the extremities; notice also that K_5 is the smallest graph that is not partitionable.) The proof follows immediately. (Notice that the connection piece could be used for truth setting rather than the truth setting piece itself, following the lines of the next proof.)

A larger, but also simpler piece builds 3 pairs of triangles with the two vertices as bases: u, v, t_{i1} and u, v, t_{i2}, with additional edge $\{t_{i1}, t_{i2}\}$, for $i = 1, 2, 3$; then it adds a new triangle, say t_1, t_2, t_3, and connects t_i to both t_{i1} and t_{i2}. If vertices u and v have color 1, they force all t_{ij} to have color 2, which in turn forces all vertices t_i to have color 1, which forms a monochromatic triangle.

From *Positive NAE3SAT*, things are easier; we only need use our connector, setting up a triangle for each clause and a single vertex for each variable. That this construction works is obvious. Alternately, we can set up a triangle for each clause, one for each variable, and connect each clause "variable" to all three vertices of the corresponding variable triangle. (This is in fact the same thing, with a little simplification arising from dispensing with the variable vertex and re-using the connecting piece's triangle.)