3515ICT Theory of Computation

Take-home test 4 answers

1. Straightforward.

2. Omitted.

3. (a) There are 8 possible truth assignments to 3 variables. Every clause with 3 literals is false for exactly 1 of the 8 possible assignments. So if a formula has 7 or fewer clauses, there must be one assignment that does not make any clause false, and this assignment makes the formula true.

   But if a formula has 8 or more clauses, there is no satisfying truth assignment. In fact, there is a unique (up to reordering) formula with 8 clauses:

   $$(x + y + z)(x + y + \overline{z})(x + \overline{y} + x) \cdots$$

   (b) With just 2 variables (not required, but helpful in understanding the problem), the smallest nae-unsatisfiable formula is this:

   $$(x + y)(x + \overline{y})(\overline{x} + \overline{y})$$

   If an assignment nae-satisfies a clause, then so does its complement. So there are only 4 distinct assignments to 3 variables wrt nae-satisfiability. And only 1 of these 4 distinct assignments makes any given clause nae-unsatisfiable. So a formula with 4 clauses each of which fails to satisfy 1 of the 4 distinct assignments can be nae-unsatisfiable. It follows that any formula with 3 or fewer clauses is nae-satisfiable.

   Here is one nae-unsatisfiable formula with 4 clauses:

   $$(x + y + z)(x + \overline{y} + \overline{z})(\overline{x} + y + \overline{z})(\overline{x} + \overline{y} + z)$$

   (c) As argued above, any formula with 3 variables and 7 clauses is satisfiable, hence any formulas with 4 variables and 7 clause is (even more) satisfiable.

   Not every formula with 4 variables and 8 clauses is unsatisfiable, but many are. They can be found by analysis or by using a computer program. Here is one solution:

   $$(x + y + \overline{z})(w + y + z)(\overline{w} + x + z)(\overline{w} + \overline{x} + y)(\overline{x} + \overline{y} + z)(\overline{w} + \overline{y} + \overline{z})$$
   $$(w + \overline{x} + \overline{z})(w + x + \overline{y})$$

   Here is another, perhaps easier to understand, solution:

   $$(x + y + z)(x + y + \overline{z})(\overline{x} + y + z)(\overline{x} + y + \overline{z}) \qquad \text{(forces } y \text{ to be true)}$$
   $$(w + x + \overline{y})(\overline{w} + x + \overline{y})(w + \overline{x} + \overline{y})(\overline{w} + \overline{x} + \overline{y}) \qquad \text{(forces } y \text{ to be false)}$$

4. (a) One Hamiltonian cycles goes almost all the way round the outer cycle, then almost all the wy round the inner cycle in the opposite direction, then returns to the start node.

   Another traverses alternate edges from the outer and inner cycles.

   (b) There are two maximal independent sets:

   {1, 12, 3, 14, 5, 16, . . . , 9, 20}

   {11, 2, 13, 4, 15, 6, . . . , 19, 10}

(c) There are two minimal vertex covers:

Each maximal independent set of this graph is a minimal vertex cover.

(d) There are three minimal edge covers:

$\{$(1,2),(12,13),(3,4),(14,15),...,(20,11)$\}$

$\{$(11,12),(12,2),(2,3),(3,13),(13,14),...,(10,1)$\}$

$\{$(1,11),(2,12),(3,13),...,(10,20)$\}$

5. The graph has one node for each of the 9 literals in the formula. There are edges between all pairs of noncomplementary literals in distinct clauses, *e.g.*, (x, y), (x, w), $(x, \overline{w})$.

   The parameter k is 3, the number of *clauses* in the formula.

   Because the given truth assignment is a "satisfying" truth assignment, it makes at least one literal in each clause true, *e.g.*, x, y and y. Then the subgraph consisting of these three nodes is a 3-clique.

6. One possible nae-satisfying truth assignment is the following:

$$\{x \to 1,\ y \to 0,\ z \to 1,\ w \to 1\}$$

   (There are many others.)

   The graph (which is an instance of 3-coloring) consists of the following groups of nodes:

   (a) One root node, q, say.

   (b) One pair of node for each variable in the formula: $(x, \overline{x})$, $(y, \overline{y})$, ... Each of these nodes is connected to the other node in its pair and to node q.

   (c) One triangle of nodes for each clause in the formula: $(x, y, \overline{z})$, ... Each of these nodes is connected to the other two nodes in its triangle and to the node with the same label (*i.e.*, to the same literal) in one of the pairs above.

   For the nae-assigment above, we assign q color 2, and we assign each literal in a pair the color determined by the truth assignment (x is assigned color 1, $\overline{x}$ color 0, y color 0, *etc.*). For each clause, choose one literal to be true and one to be false, *e.g.*, x and y, w and y, z and y. In the corresponding clauses, assign each of these literals the *opposite* color (*e.g.*, in the first clause, x is assigned color 0 and y color 1), and assign the third node in each clause color 2.

7. (a) First note that a graph G is 2-colorable if and only if every primitive cycle in G has an even number of nodes. (A cycle is *primitive* if it does not contain any node twice.) There is a polynomial-time algorithm to find all primitive cycles in a graph. (See any good algorithms text.) And there is a trivial linear-time algorithm to check the parity of each primitive cycle.

   (b) I'm no longer sure that 4-COLORING is in $\mathcal{P}$. I look forward to seeing your solutions.

8. First note that every vertex cover is a dominating set, but not every dominating set is a vertex cover. In particular, any one node of a triangle is a dominating set but not a vertex cover.

   Now, DOMINATING SET is clearly in $\mathcal{NP}$ because it is efficient to determine whether or not a given subset of nodes in a graph is a dominating set. To complete the proof that

DOMINATING SET is $\mathcal{NP}$-complete, we reduce VERTEX COVER to it. Clearly, the identity transformation doesn't work because of the above observation.

So, given a graph $G = (V, E)$, construct a new graph G' from G as follows: for each edge e in E, add a new node v to V and connect v to each endpoint of e, making a triangle t_e. Remove each isolated node in G from G'. Let the parameter k remain unchanged. Clearly this transformation can be performed in time polynomial in $|G|$. Further, G has a vertex cover of size k iff G' has a dominating set of size k'. If S is a vertex cover of G, then every edge e in E is covered by a node in S, and hence every node in the triangle e is either in S or is adjacent to a node in S, so S is a dominating set for G'. Conversely, if S is a dominating set for G', then every triangle t_e must contain an element s of S as no other nodes in G' are adjacent to the new node in t_e. If s is an endpoint of e, it covers e; otherwise, replace s by either endpoint of e. The resulting set, S', is cleaerly a vertex cover for G of size at most k.

Rodney Topor