Problem

This problem is inspired by the single-player game *Minesweeper*, generalized to an arbitrary graph. Let G be an undirected graph, where each node either contains a single, hidden *mine* or is empty. The player chooses nodes, one by one. If the player chooses a node containing a mine, the player loses. If the player chooses an empty node, the player learns the number of neighboring nodes containing mines. (A neighboring node is one connected to the chosen node by an edge.) The player wins if and when all empty nodes have been so chosen.

In the *mine consistency problem*, you are given a graph G along with numbers labeling some of G's nodes. You must determine whether a placement of mines on the remaining nodes is possible, so that any node v that is labeled m has exactly m neighboring nodes containing mines. Formulate this problem as a language and show that it is NP complete.

Step-by-step solution

Step 1 of 5

Suppose that
$$MCP = \left\{ \begin{array}{l} \langle G, N\rangle \mid G \text{ is a graph } (V,E),\ N \text{ is labeling function } N: V \to \{X,0,1,2,\ldots\} \\ \text{and there exists mine's placement on the set of nodes labeledso that every} \\ \text{numbers are consistence} \end{array} \right\}$$

The language MCP is in NP, because in polynomial time one can easily test if a placement of mines is consistence. First users have to show $3SAT \le_P MCP$, to prove that it is also NP-complete.

Comment

Step 2 of 5

Consider φ is a Boolean formula, now user want to convert it to G, which is an instance of the MCP problem.

Here, suppose $c_1, c_{2,\ldots}$ are used to denote the clause appearing in φ. Now, suppose $x_1, x_{2,\ldots}$

Denotes the variable used in φ. Here, a variable x_i 'appears' in φ if one of x_i or $\overline{x}_i$ appear in minimum single clause in φ.

For all the variables x_i which appears in φ:

1. Three nodes are created as x_i, x_i^f and x_i^t.

2. Edges are added (x_i, x_i^t) and (x_i, x_i^f), and

3. Now, set $N(x_i)=1,\ N(x_i^t)=X$ and $N(x_i^f)=X$

Comment

Step 3 of 5

For all the clause c_i which appears in φ:

1. Three nodes are created as c_i, c_i^1 and c_i^2.

2. Edges are added from c_i to the nodes corresponding to the three literals in c_i

3. Now, set $N(c_i)=3,\ N(c_i^1)=X$ and $N(c_i^2)=X$

Comment

Step 4 of 5

All instances of the 3SAT problem can be reduced to an instance of the Circuit-SAT problem in polynomial time in a trivial manner by changing the Boole operators to a circuit of logic gates. The validity of this reduction needs to be proven.

• If there is an instance of the 3SAT problem it can be converted to an instance of the Circuit-SAT problem by simply mapping Boolean operators to logic gates and connections between the gates.

• Setting the corresponding logic inputs to the Boolean values that satisfy the 3SAT problem will satisfy this instance of the Circuit-SAT problem.

• Using an instance of the Circuit-SAT problem, an equivalent instance of the SAT problem is constructed by interchanging logic gates/wires with Boolean variables and operators. If the logical values that satisfy the instance of Circuit-SAT are changed into Boolean values, then the SAT instance will be satisfied.

Comment

Step 5 of 5

The Circuit-SAT problem is reducible in polynomial time to an NP-complete problem, so it is also in NP-complete. Hence, the MCP problem is NP-complete.

Comments (1)