B62005Y-02 理论计算机科学基础	2017 年 6 月 2 日

Solutions to Homework # 6

主讲教师：杨光 | 张远航 2015K8009929045

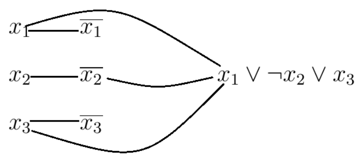
Sipser 7.16 We first show that the problem is in NP. The following NTM decides it:

$M =$ "On input $G = \langle V, E\rangle$:

1. Nondeterministically split V into two sets, C and $V \setminus C$;
2. Assign directions to each edge (c, w), where $c \in C$ such that each node c has either indegree 0 or outdegree zero;
3. Accept if every node in $V \setminus C$ has indegree at least one."

Clearly, this machine runs in $2^{O(n)}$ time, so the problem is in NP.

To show that the problem is NP-hard, we give a reduction from *3SAT*. Let the vertices in C be variables and the vertices in $V \setminus C$ be clauses. We add edges between each x_i and $\overline{x_i}$ and add an edge from the literals to clauses that contain them. Then the clause vertices have indegree at least one if and only if they are satisfied by a variable vertex. Here is an illustration:



We have found an reduction from *3SAT* which implies that the problem is NP-complete. □

Sipser 7.29 To prove *SET-SPLITTING* is NP-complete, we need to show *SET-SPLITTING* is in NP and *SET-SPLITTING* is NP-hard. The first claim is obvious since given a coloring, it is easy to verify in polynomial time that no set is monochromatic.

To prove that *SET-SPLITTING* is NP-hard, we reduce *3SAT* to it. Let ϕ be a 3-CNF formula with variable set V. Construct the following instance of *SET-SPLITTING*: the set of elements $S = \{a\} \cup V \cup \{\bar{x} \mid x \in V\}$, where a is a new element not related to any variable. Each other element corresponds to a literal.

We build the collection of sets C as follows. For each variable x in ϕ, construct a set $S_x = \{x, \bar{x}\}$. For each clause c, construct a set $S_c = \{x, y, \bar{z}, a\}$. Finally, let $C := \bigcup_x S_x \cup \bigcup_c S_c$. So the reduction is $f(\phi) = \langle S, C \rangle$ where S and C are as described above. Clearly the reduction is polynomial time.

To finish we prove that ϕ is satisfiable if and only if $\langle S, C \rangle$ can be colored so that no set is monochromatic.

($\Rightarrow$) Suppose ϕ is satisfiable. Fix some satisfying assignment. Consider the following coloring of the elements in S. Color the element a "red". For each variable x that is assigned "false", color the elements x and $\bar{x}$ "red" and "blue", respectively. For each variable x that is assigned "true", color the elements x and $\bar{x}$ "blue" and "red", respectively. As long as the assignment was satisfying, this coloring makes no set monochromatic. For each variable x, the set $S_x = \{x, \bar{x}\}$ has one red and one blue element. For each clause c, the set S_c has at least one red element a and, because some literal in the clause has a value of true, S_c has at least one "blue" element. Thus, $\langle S, C \rangle \in$ *SET-SPLITTING*.

($\Leftarrow$) Suppose $\langle S, C \rangle \in$ *SET-SPLITTING*. Fix some coloring of S with two colors such that every set has at least one element of both colors.

Consider the following assignment to the variables of ϕ. For each variable x, assign it "true" if its color differs from that of the element a. Assign x "false" if its color is the same as that of the element a. Then each clause c in ϕ is satisfied, because the set S_c has at least one element x or $\bar{x}$ that is colored differently than a. Thus, $\phi \in$ *SAT*. $\square$

Sipser 7.32 Clearly *DOMINATING-SET* is in NP. Given a dominating set, one can verify in polynomial time if that is a dominating set. This can be done by taking each vertex and checking if it is either in the given set or one of its edges travel into the set.

To show that is NP-complete, first of all notice that a dominating set has to include all isolated vertices (those which have no edges from them). So let us assume that our graph does not have any isolated vertices. We will show that Dominating Set is NP-complete using a reduction from *VERTEX-COVER*. Given a graph G, we will construct a graph G' as follows. G' has all edges and vertices of G. Also, for every edge $\{u, v\} \in G$, we add intermediate node on a parallel path in G'. Keeping $\{u, v\}$ intact in G', we add vertex w and edges $\{u, w\}$ and $\{w, v\}$ in G'. Now we will show that G has a vertex cover of size k if and only if G' has a dominating set of the same size.

If S is a vertex cover in G, we will show that S is a dominating set for $G'.S$ is a vertex cover, this means that every edge in G has at least one of its end points in S. Consider $v \in G'$. If v is an original node in G, then either $v \in S$ or there must be some edge connecting v to some other vertex u. Since S is a vertex cover, is $v \notin S$, then u must be in S, and hence there is an adjacent vertex of v in S. So v is covered by some element in S. However, if w is an additional node in G', then w has two adjacent vertices $u, v \in G$ and using the above argument at least one of them is in S. So the additional nodes are also covered by S. So if G has a vertex cover, then G' has a dominating set of at most the same size (in fact the same set itself would do).

If G' has a dominating set D of size k, then look at all the additional vertices $w \in D$. Notice that w must be connected to exactly 2 vertices $u, v \in G$. Now see that we can safely replace w by one of u or $v.w$ in D will help us dominate only $u, v, w \in G'$. But these three edges form a 3-cycle, and we can as well pick u or v and still dominate all the vertices that w used to dominate. So we can eliminate all the additional vertices as above. Since all the additional vertices correspond to one of the edges in G, and since all of the additional vertices are covered by the modified D, this means that all the edges in G are covered by the set. So if G' has a dominating set of size k, then G has a vertex cover of size at most k.

So we have proved both sides of equivalence. A dominating set of size k exists in G' if and only if a vertex cover of size k exists in G. Since we know that *VERTEX-COVER* is NP-complete, *DOMINATING-SET* is also NP-complete. □

Sipser 7.39 We show that the problem is in NP by showing that it is verifiable in polynomial time. To do this, we simply stack the cards according to the answer presented in the certificate to determine if they cover all the holes. This can be accomplished easily in polynomial time.

To show that it is NP-complete, we reduce *3SAT* to it. We create cards $\{x_1, x_2, \ldots\}$ for each variable in the *3SAT* formula and create a hole position in each column for each term of the formula. Then, we punch holes in the left column of the card in every position which corresponds to a term that does not contain that card's variable and in the right column for every term which does not contain that card's variable's complement. The *PUZZLE* problem is only satisfiable if every hole can be covered by one of the card which implies that every term in *3SAT* problem is satisfiable. □