计算理论导论

习题十: NP 完全问题

中国人民大学　信息学院　崔冠宇　2018202147

1. Let us define the "Safe Marriage" problem. There are n people (but there is no notion of gender in this problem). Each pair of people u and v either like or dislike each other. A "Safe Marriage of size k" is a set of pairs $\{u_1, v_1\}, \{u_2, v_2\}, \cdots, \{u_k, v_k\}$ such that:

- u_i and v_i like each other,
- u_i is the only person that v_i likes amongst $\{u_1, v_1, \cdots, u_k, v_k\}$,
- v_i is the only person that u_i likes amongst $\{u_1, v_1, \cdots, u_k, v_k\}$.

The objective is to decide if there is a Safe Marriage of size k. We can model this problem using an undirected graph G, where the vertices correspond to people, and the edges correspond to pairs who like each other. Let us define the decision problem

$$\text{SAFEMARRIAGE} = \{\, \langle G, k\rangle \mid G \text{ has a safe marriage of size } k\}.$$

Prove that SAFEMARRIAGE is NP-Hard. Hint: Try a reduction from Independent Set, which we have shown to be NP-Complete.

解: 往证 INDSET $\leq_P$ SAFEMARRIAGE。归约函数 f 如下定义：

对于 $G = (V, E)$（其中 $V = \{1, 2, \cdots, n\}$），先将 G 复制一份记为 $G_1 = (V_1, E_1)$（其中 $V_1 = \{1', 2', \cdots, n'\}$），然后将 G 和 G_1 对应顶点相连（即添加 (k, k')），最后若 $(x, y) \in E$，则添加 (x, y') 和 (x', y) 两条边。记生成的新图为 G'。容易看出，这个过程添加了 $|V|$ 个顶点，$3|E|$ 条边，因此是多项式时间可计算的。下面只需要说明归约的正确性：

1. 若 $\langle G, k\rangle \in$ INDSET，则 G 中存在大小为 k 的独立集 $\{i_1, i_2, \cdots, i_k\}$。断言 G' 中的 k 对顶点

$\{i_1, i'_1\}, \{i_2, i'_2\}, \cdots, \{i_k, i'_k\}$ 构成大小为 k 的 Safe Marriage，因为：

(a) 根据 G' 的构造方法，$(i_j, i'_j) \in E(G')$；

(b) 由于 $\{i_j\}$ 是独立集，即它们之间两两不相连，故 i_j 在 $\{i_1, i'_1, \cdots, i_k, i'_k\}$ 中只与 $\{i'_j\}$ 相连，反之亦然。

于是 $f(\langle G, k\rangle) = \langle G', k\rangle \in$ SAFEMARRIAGE。

2. 若 $f(\langle G, k\rangle) = \langle G', k\rangle \in$ SAFEMARRIAGE，即 G' 中存在大小为 k 的 Safe Marriage $\{u_1, v_1\}, \{u_2, v_2\}, \cdots, \{u_k, v_k\}$。若某 u_j, v_j 都在 G 中或都在 G_1 中，把其中一个顶点换成对应顶点，使二者一个位于 G 中，另一个位于 G_1 中，得到 $\{u_j, u'_j\}$（不妨令 u_j 总是在 G 中）。对所有点对都施行上述操作，得到的点对仍然是 Safe Marriage。断言 $\{u_j\}$ 构成了 G 上的独立集，因为若 $(u_i, u_j) \in E(G')$，还有 $(u_i, u'_i) \in E(G')$，则与 Safe Marriage 矛盾。于是 $\langle G, k\rangle \in$ INDSET。

因此这是多项式归约，故有 INDSET $\leq_P$ SAFEMARRIAGE，又因为 INDSET $\in$ NP-Complete，结论得证。

2. Prove that the following problem called DS (Dominating Set) is NP-Complete. A dominating set in a graph $G(V, E)$ is a set of vertices $S \subseteq V$ such that each vertex in V is either in S or has an edge to some vertex in S. The definition of the problem is the following:

INPUT: Graph $G(V, E)$ and integer k.

PROBLEM: Does G have a dominating set of size at most k?

We suggest you use the following reduction from 3-SAT to DS. The reduction function takes as input a 3-SAT formula $F(X_1, X_2, \cdots, X_n)$ with clauses $C_1, \cdots, C_m$ such that $C_i = Z_{i1} \vee Z_{i2} \vee Z_{i3}$, where Z_{ij} denotes a literal. The output of the reduction is a graph $G(V, E)$ and $k = n$. The graph is constructed as follows. For each clause C_i there is a vertex c_i. For each variable X_i there are three vertices x_i, $\neg x_i$ and y_i. The three vertices for each variable are connected to each other to form a triangle. The vertex for a clause is connected by an edge to the three vertices, which correspond to its literals.

解: 分两方面证明:

1. DS $\in$ NP。

   显然，只需要取证书为一个节点集合 S，验证时只需要先确认集合中点的个数不超过 k，然后遍历节点，检查每个节点是否要么属于 S，要么有边与 S 中的点相连。显然验证过程是多项式可计算的。

2. DS $\in$ NP-Hard。

   往证 3-SAT $\leq_P$ DS。归约函数如上所述，由于归约时先构造了 $3|V|+|E|$ 个顶点，然后又构造了 $3|V|+3|E|$ 条边，显然是多项式时间可计算的。而且，

   (a) 若 $\langle F\rangle \in$ 3-SAT，则存在一组对变量 X_i 的赋值使得每个子句 (clause) 都为真。构造集合 S: 对每一个变量 X_i，若它的赋值为真，则取 x_i，否则取 $\neg x_i$。由于 F 可满足，所以每个子句中至少有一个文字 (literal) 为真，于是 c_i 至少与一个选中的节点相连；同时由于每个 $x_i, \neg x_i, y_i$ "三角形"中都有一个节点被选中，于是这 n 个节点的集合构成一个支配集，即 $f(\langle F\rangle) \in$ DS。

   (b) 若 $f(\langle F\rangle) \in$ DS，则在构造出的图中含有至多 n 个顶点的支配集。根据支配集的定义，每个节点要么属于 S，要么与 S 中的顶点相邻，而 y_i 只与 $x_i, \neg y_i$ 相邻，因此每个 $x_i, \neg x_i, y_i$ 中都至少有一个在 S 中，但是由于 $|S| = n$，于是每个 $x_i, \neg x_i, y_i$ 中都有且仅有一个在 S 中。构造 F 的成真赋值：对每一个 X_i，如果 $x_i \in S$，则 X_i 赋值为真，否则 X_i 赋值为假。考虑子句 C_j，由于点 c_j 不在支配集中，所以 S 中必有某节点 x_m 或 $\neg x_n$ 与之相连。若 x_m 与之相连，由于此时 X_m 被赋值为真，则 C_j 为真；若 $\neg x_n$ 与之相连，由于此时 X_n 赋值为假，于是 C_j 也为真，因此 $\langle F\rangle \in$ 3-SAT。

   因此这确实是一个多项式归约，故有 3-SAT $\leq_P$ DS。由于 3-SAT 是 NP-Complete 的，于是结论得证。

3. The SUBGRAPH-ISOMORPHISM problem is defined as follows: Given two graphs G_1 and G_2, does G_1 contain a copy of G_2 as a subgraph? Show that SUBGRAPH-ISOMORPHISM is NP-Complete. (Hint:

Think about the CLIQUE problem.)

解: 先将这个问题形式化成一个语言:

$$\text{SUBGRAPH-ISOMORPHISM} = \{\ \langle G_1, G_2\rangle \mid \exists G \subseteq G_1, G \cong G_2\}$$

分两方面证:

1. SUBGRAPH-ISOMORPHISM $\in$ NP。

   用"多项式时间可验证"的定义证。设 $V(G_1) = \{1, 2, \cdots, m\}, V(G_2) = \{1, 2, \cdots, n\} (m \geq n)$, 对于任意 $\langle G_1, G_2\rangle$, 当且仅当 $\langle G_1, G_2\rangle \in$ SUBGRAPH-ISOMORPHISM 时存在证书 (certificate) $y = \{(1, \sigma(1)), (2, \sigma(2)), \cdots, (n, \sigma(n))\}$ 满足 f 是 G_2 到 G_1 子图 G 的同构。证书 y 显然满足 $|y| \leq |\langle G_1, G_2\rangle|^k$, 而且可以多项式时间验证:

```
for i = 1 to n - 1:
    for j = i to n:
        if (i, j) 属于 E(G_2) 但 (f(i), f(j)) 不属于 E(G_1):
            reject
accept
```

2. SUBGRAPH-ISOMORPHISM $\in$ NP-Hard。

   往证 CLIQUE $\leq_P$ SUBGRAPH-ISOMORPHISM。归约函数 g 满足 $g(\langle G, k\rangle) = \langle G, K_k\rangle$, 其中 K_k 是 k 阶完全图。显然这个归约函数是多项式时间可计算的, 而且 $\langle G, k\rangle \in$ CLIQUE $\Leftrightarrow$ G 含有一个 k 团 $\Leftrightarrow \exists K_k \subseteq G \Leftrightarrow \langle G, K_k\rangle \in$ SUBGRAPH-ISOMORPHISM, 因此这确实是一个多项式归约, 故有 CLIQUE $\leq_P$ SUBGRAPH-ISOMORPHISM, 而 CLIQUE 是 NP-Complete 的, 于是结论得证。

4. Prove that the following problem, called TRUE-SAT, is NP-Complete. The definition of the problem is the following:

INPUT: A Boolean formula $F(X_1, X_2, \cdots, X_n)$ such that $F(T, T, \cdots, T) = T$. In other words, F can be

satisfied by setting each variable X_i to TRUE.

PROBLEM: Does F have a satisfying truth assignment in which at least one of the variables is set to FALSE? We propose the following reduction from 3-SAT to TRUE-SAT. Given a 3-SAT formula F with clauses $C_1, \cdots, C_m$ and the variables $X_1, \cdots, X_n$, we construct a CNF formula G which is an instance of the TRUE-SAT problem. The new formula G has one new variable called Y. For each clause C_i in F, we add the clause $C_i \vee Y$ to G. We also add to G clauses of the form $\neg Y \vee X_i$, for each X_i. For example, if $F = (X_1 \vee \neg X_2 \vee X_3) \wedge (\neg X_1 \vee X_2 \vee X_3)$ then $G = (Y \vee X_1 \vee \neg X_2 \vee X_3) \wedge (Y \vee \neg X_1 \vee X_2 \vee X_3) \wedge (\neg Y \vee X_1) \wedge (\neg Y \vee X_2) \wedge (\neg Y \vee X_3)$.

解: 分两方面证：

1. TRUE-SAT $\in$ NP。

   用“多项式时间可验证”的定义证。对于任意满足 $F(T, T, \cdots, T) = T$ 的布尔公式，证书 y 的形式为 n 个布尔值，分别代表对 $x_1, x_2, \cdots, x_n$ 的赋值。当且仅当 $\langle F \rangle \in$ TRUE-SAT 时存在至少含一个 F 的证书。证书 y 显然满足 $|y| \leq |\langle F \rangle|^k$，而且可以多项式时间验证，只需要代入验证即可。

2. TRUE-SAT $\in$ NP-Hard。

   往证 3-SAT $\leq_P$ TRUE-SAT。归约函数如上所述。由于归约时添加了 $m+n$ 个 Y 以及 n 个 X_i，这是多项式时间可计算的。下面需要证明归约的正确性。

   (a) 若 $\langle F \rangle \in$ 3-SAT，则存在一组对变量 X_i 的赋值使得 F 为真。对 $f(\langle F \rangle)$ 而言，当所有变量为 TRUE 时，显然每个子句都为真，整个公式为真；若令 Y 为 FALSE，则新公式与 F 等价，取满足 F 的变量赋值，可以使公式为真，即存在至少一个变量赋值为 FALSE 的成真赋值。于是 $f(\langle F \rangle) \in$ TRUE-SAT。

   (b) 若 $f(\langle F \rangle) \in$ TRUE-SAT，则 $f(\langle F \rangle)$ 对所有变量赋值 TRUE 时值为真，而且存在至少一个变量赋值为 FALSE 的成真赋值。断言这种赋值中 Y 的赋值一定为 FALSE，用反证法，假设 Y 的赋值为 TRUE，由于至少一个变量赋值为 FALSE，若 X_i 为 FALSE，则子句 $(\neg Y \vee X_i)$ 为 FALSE，$f(\langle F \rangle)$ 为 FALSE，与成真赋值矛盾。在这种赋值下，由于 Y 赋值为 FALSE，$f(\langle F \rangle)$

与 F 等价，将 $f(\langle F\rangle)$ 中对于 X_i 的赋值搬到 F 中，可以满足 F。于是 $\langle F\rangle \in$ 3-SAT。

所以是多项式归约，故有 3-SAT $\leq_P$ TRUE-SAT。由于 3-SAT 是 NP-Complete 的，于是结论得证。

5. **7.21** Let G represent an undirected graph. Also let

$$\text{SPATH} = \{\ \langle G, a, b, k\rangle \mid G \text{ contains a simple path of length at most } k \text{ from } a \text{ to } b\},$$

and

$$\text{LPATH} = \{\ \langle G, a, b, k\rangle \mid G \text{ contains a simple path of length at least } k \text{ from } a \text{ to } b\}.$$

a. Show that SPATH $\in$ P.

b. Show that LPATH is NP-Complete.

解:

(a) 直接给出一个判定 SPATH 的 P 算法（设 $V(G) = \{1, 2, \cdots, n\}$，$|E(G)| = m$）：

```
d[a] = 0
for i = 0 to m:
    扫描每一条边 (s, t)，若 d[s] == i，则 d[t] = i + 1
if d[j] <= k:
    accept
else:
    reject
```

容易看出这个算法是多项式时间的。

(b) 分两方面证：

1. LPATH $\in$ NP。

显然，只需要取证书为一个节点序列，验证时只需要验证该序列是否构成简单路径，以及长度是

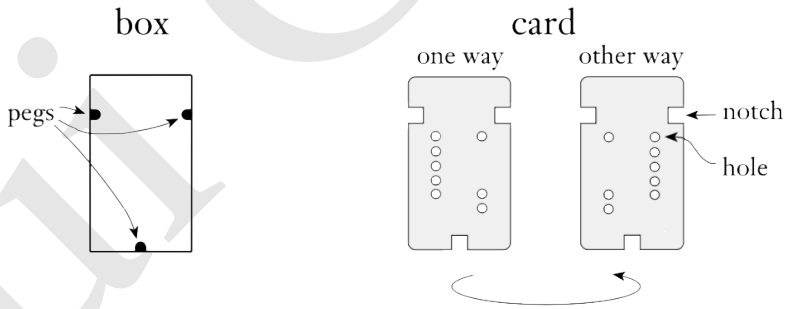
否至少为 k 即可。验证过程是多项式时间可计算的。

2. LPATH $\in$ NP-Hard。

往证 UHAMPATH $\leq_P$ LPATH。归约函数为 $f(\langle G, a, b\rangle) = \langle G, a, b, |V(G)| - 1\rangle$。

显然 $\langle G, a, b\rangle \in$ UHAMPATH $\Leftrightarrow$ 存在一条 a 到 b 的经过所有节点的简单路径 $\Leftrightarrow$ 存在一条 a 到 b 的长度为 $|V(G)|-1$ 的简单路径 $\Leftrightarrow \langle G, a, b, |V(G)|-1\rangle \in$ LPATH。容易看出，归约函数是多项式可计算的，因为只需要扫描各节点，求出节点个数。所以是多项式归约，故有 UHAMPATH $\leq_P$ LPATH，由于 UHAMPATH 是 NP-Complete 的，于是结论得证。

6. **7.28** You are given a box and a collection of cards as indicated in the following figure. Because of the pegs in the box and the notches in the cards, each card will fit in the box in either of two ways. Each card contains two columns of holes, some of which may not be punched out. The puzzle is solved by placing all the cards in the box so as to completely cover the bottom of the box (i.e., every hole position is blocked by at least one card that has no hole there). Let PUZZLE $= \{ \langle c_1, \cdots, c_k\rangle \mid$ each c_i represents a card and this collection of cards has a solution$\}$. Show that PUZZLE is NP-Complete.



解: 分两方面证。

1. PUZZLE $\in$ NP。

可以构造一台非确定性图灵机 N 在多项式时间内判定这个问题：对于每张卡片 c_i，非确定地选取它的方向，然后检查每一个孔是否都能被覆盖即可。容易看出非确定性图灵机能在多项式时间判定这个问题。

2. PUZZLE $\in$ NP-Hard。

往证 3-SAT $\leq_P$ PUZZLE。设一个 n 变量 m 子句的 3-CNF 为 $F(X_1, X_2, \cdots, X_n)$，为每一个变量 X_i 构造一张卡片 c_i，其中每张卡片上有 m 行打孔位置，按如下方式打孔（从卡片正面看）：

- 若 X_i 出现在子句 C_j 中，则第 j 行左侧不打孔，否则打孔；
- 若 $\neg X_i$ 出现在子句 C_j 中，则第 j 行右侧不打孔，否则打孔；

最后再构造一张卡片，它左侧 m 行均被打孔，右侧 m 行均不被打孔。

显然这个过程是多项式时间可计算的，因为只需要依次扫描各子句，不断构造/修改卡片即可。下面只需要证明归约的正确性：

(a) 若 $\langle F \rangle \in$ 3-SAT，即 $F(X_1, X_2, \cdots, X_n)$ 有一组成真赋值，则按如下方式放置各卡片：若在这组赋值中 X_i 为 TRUE，正面朝上放置 c_i 卡片，否则翻转卡片；正面朝上放置最后的特殊卡片。由于特殊卡片右侧 m 行都没有孔，右侧 m 行一定都被盖住，于是只需要考虑左半部分。断言这种放置一定构成了 PUZZLE 的一组解。用反证法，假设左侧第 j 行没有被盖住，根据卡片的打孔规则与放置规则，对于每张正放的卡片 c_i，它对应的 X_i 赋值为 TRUE，要么变量 X_i 不出现在子句 C_j 中，要么变量以 $\neg X_i$ 出现在子句中，但这两种情况都不能满足 C_j；反放的卡片类似。于是子句 C_j 没有满足，矛盾。

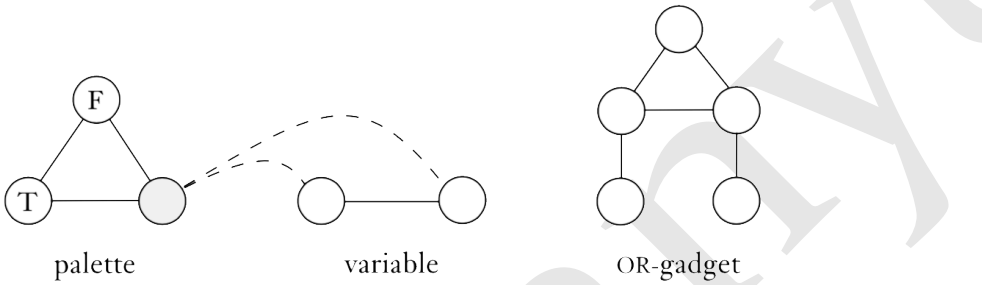
(b) 若 $f(\langle F \rangle)$ 有一组解，则按如下方式给 F 赋值：不妨设特殊卡片是正放的（反放的则操作相反），即它没有盖住左侧的所有 m 行。由于 PUZZLE 有解，对于每一行 j 都存在一张卡片 c_i 盖住了它，若这张卡片是正放的，则给 X_i 赋值为 TRUE；否则赋值为 FALSE。对每一行 j 重复该过程，若最终某 X_k 没有被赋值，则随意给它赋值。断言这种赋值一定满足了 F。因为对于每个子句 C_j，它所对应的 j 行被 c_i 盖住，要么是正放时 X_i 为 TRUE 满足了 C_j，要么反放时 $\neg X_i$ 为 TRUE 满足了 C_j，于是每个子句都被满足，F 被满足。

所以这是多项式归约，故有 3-SAT $\leq_P$ PUZZLE，由于 3-SAT 是 NP-Complete 的，于是结论得证。

7. 7.29 A ***coloring*** of a graph is an assignment of colors to its nodes so that no two adjacent nodes are assigned the same color. Let

$$\text{3COLOR} = \{\langle G\rangle \mid G \text{ is colorable with } 3 \text{ colors}\}.$$

Show that 3COLOR is NP-Complete. (Hint: Use the following three subgraphs.)



解: 分两方面证明:

1. 3-COLOR $\in$ NP。

   显然，只需要取证书为每一个节点的着色（0, 1, 2 之一），验证时只需要遍历每一条边，确认两个相邻顶点的着色不同即可（当然也必须是 0, 1, 2 之一），显然验证过程是多项式时间的。

2. 3-COLOR $\in$ NP-Hard。

   往证 3-SAT $\leq_P$ 3-COLOR。设 $\varphi(X_1, X_2, \cdots, X_n)$ 是布尔公式，按如下方式归约:

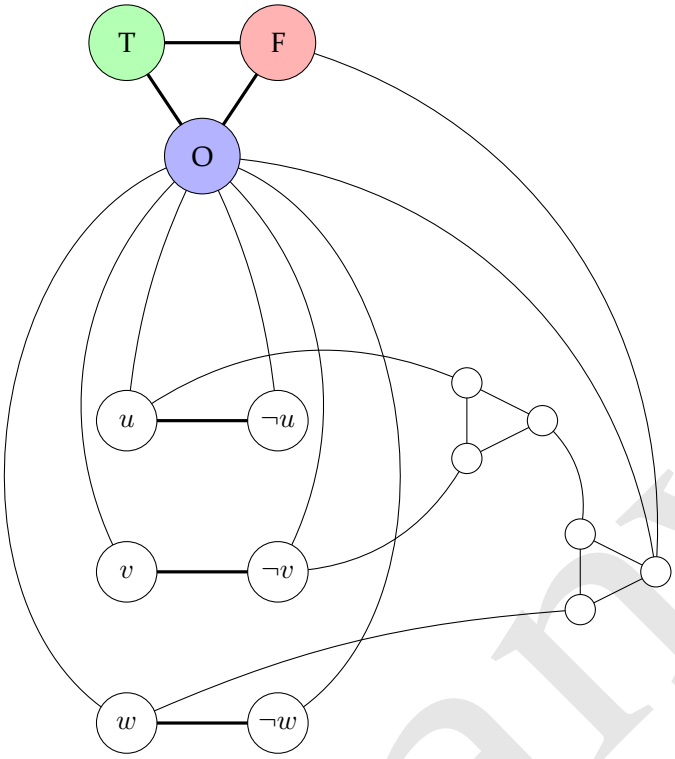
   (a) 按上图构造一个“palette”，且三个顶点分别染色为 0(FALSE/F)、1(TRUE/T)、2(OTHER/O);

   (b) 对每一个变量 X_i，构造两个相连的顶点 $x_i, \neg x_i$，然后分别将它们与“palette”的颜色为 2 的节点相连;

   (c) 对每一个子句 $C_j = (X \vee Y \vee Z)$，构造两个“OR-gadget”(一个三角形)，其中 X, Y 与第一个“OR-gadget”的两入引脚相连，它的出引脚以及 Z 与第二个“OR-gadget”的两入引脚相连;

   (d) 对每一个子句的第二个“OR-gadget”的出引脚分别与“palette”颜色为 0 和 2 的节点相连。

   比如 $\varphi = (u \vee \neg v \vee w)$ 构造的图如下:



公式 φ 归约产生出的图有以下几个特点：

- 由于每个变量的两个节点都与着色为 2 的节点相连，于是这两个节点的合法着色必然是一个 0 一个 1；
- 若某个“OR-gadget”的一个入引脚与一个着色为 1 的变量节点相连，则存在一种着色，使得它的出引脚着色也为 1；
- 若某个“OR-gadget”的两个入引脚均与着色为 0 的变量节点相连，则出引脚颜色必须为 0；
- 由于每个子句的“OR-gadget”的出引脚分别与颜色为 0、2 的节点相连，因此出引脚合法着色只能为 1。

接下来需要证明归约是正确的，分两个方面：

(a) 若 $\langle\varphi\rangle \in$ 3-SAT，则存在一组对 X_i 的赋值使得 φ 为真。对于每一个变量 X_i，若它赋值为真，则给 x_i 染 1，给 $\neg x_i$ 染 0；否则给 x_i 染 0，给 $\neg x_i$ 染 1。由于 φ 可满足，所以每个子句 C_j 中至少有一个文字为真，于是 C_j 可以被三着色使得出引脚颜色为 1。因此 $f(\langle\varphi\rangle) \in$ 3-COLOR。

(b) 若 $f(\langle\varphi\rangle) \in$ 3-COLOR，则该图存在三着色。对每一组变量节点，若 x_i 被着色 1，给 X_i 赋值

TRUE；否则赋值为 FALSE。断言每个子句至少有一个文字为真，否则若每个子句中三个文字都为假，则出引脚颜色必定为 0，与着色合法矛盾。由于每个子句至少有一个文字为真，所以 φ 可满足，即 $\langle\varphi\rangle \in$ 3-SAT。

于是 f 是多项式归约，故有 3-SAT $\leq_P$ 3-COLOR，由于 3-SAT 是 NP-Complete 的，于是结论得证。

8. 【Optional（选做）】**7.39** Show that if P = NP, you can factor integers in polynomial time. (See the note in **Problem 7.38**.)

解: 大致思路：（需要注意，整数 N 的编码 $\langle N\rangle$ 的长度 $n = O(\log N)$）

以下问题是 NP 的：

$$\text{FACTOR} = \{\, \langle N, r\rangle \mid \exists p(1 < p < r, p \mid N)\}$$

因为当且仅当 $\langle N, r\rangle \in$ FACTOR 时，可以提供一个证书 $\langle p\rangle$ 满足 $1 < p < N$ 且 $p \mid N$。显然证书长度 $|\langle p\rangle| = O(\log N)$ 是关于输入长度 $|\langle N, r\rangle| = O(\log N)$ 的多项式，而且验证过程（比如平凡的长除法，需要 $O(n^2) = O(\log^2 N)$）也是关于 $|\langle\langle N, r\rangle, p\rangle| = O(\log N)$ 的多项式。

如果 P = NP，则存在判定 FACTOR 的多项式算法 FACTOR(N, r)，可以利用它和二分搜索技术，在多项式时间内找到 N 的一个非平凡因子 p（若没有，则返回 1）：

```
FACTORIZE(N):
    low = 1, high = N
    mid = (low + high) / 2
    # O(log N) times.
    while low < high:
        # Divide, O(n^2) = O(log^2 N)
        if N % mid == 0 and mid != 1 and mid != N:
            return mid
```

```
        # A factor in (low, mid).
        # O(n^k) = O(log^k N) for some k if P = NP.
        if FACTOR(N, mid):
            high = mid
        else:
            low = mid
    return 1
```

容易看出算法的时间复杂度为 $O(\log N) \cdot (O(\log^2 N) + O(\log^k N)) = O(\log^{\max\{k+1,3\}} N)$，结论得证。

Remark:

Ladner 证明了若 P $\neq$ NP，则存在语言 L 满足 $L \in$ NP $-$ (P $\cup$ NP-Complete)（这样的问题被称作“NP-Intermediate”问题）。目前有两个 NP 问题，FACTOR（上面的因数分解）和 GRAPH-ISOMORPHISM（图同构），一直没有被明确划分到 NP-Complete，也没有找到多项式时间算法，因此人们猜测它们可能是 NP-Intermediate 问题。Shor 和 Kitaev 曾经分别试图寻找这两个问题的量子计算机上的多项式算法，希望借此说明量子计算的（可能的）优越性。最后，Shor 成功找到了分解质因数的 Shor 算法，增强了人们对于量子优越性的信心。