CSE200 Lecture Notes – NP-complete problems (II)

Lecture by Russell Impagliazzo
Notes by Jiawei Gao

February 11, 2016

In this lecture we prove 3-coloring and Subset Sum are NP-complete.

1 3-coloring

3-coloring problem: given a graph $G = (V, E)$ find a solution $\chi : V \to \{\mathbf{R}, \mathbf{G}, \mathbf{B}\}$, satisfying $\{u, v\} \in E \to \chi(u) \neq \chi(v)$.

To reduce from 3SAT, we define an intermediate problem:

> $\text{CSP}_{2,3}$ (Constraint satisfaction problem, where constraints are binary relations, and each variable can take 3 possible values)
>
> - Input: a graph $G = (V, E)$, $\forall e \in E$, List of possible colors $L_e \subseteq \{\mathbf{R}, \mathbf{G}, \mathbf{B}\} \times \{\mathbf{R}, \mathbf{G}, \mathbf{B}\}$.
> - Solution: $\chi : V \to \{\mathbf{R}, \mathbf{G}, \mathbf{B}\}$.
> - Constraint: $\forall \{u, v\} \in E, (\chi(u), \chi(v)) \in L_{\{u,v\}}$.

1.1 Reduction from 3SAT to $\text{CSP}_{2,3}$

For each clause, we create a vertex.

For example consider the following clauses:
v_1 corresponds to $C_1 = x \vee y \vee z$
v_2 corresponds to $C_2 = \bar{x} \vee y \vee w$
v_3 corresponds to $C_3 = \bar{x} \vee \bar{y} \vee \bar{z}$
v_4 corresponds to $C_4 = \bar{w} \vee z$

For clause $C_i = \ell_1 \vee \ell_2 \vee \ell_3$, we establish these relations:
ℓ_1 is set to true $\iff$ C_i is colored red.
ℓ_2 is set to true $\iff$ C_i is colored green.
ℓ_3 is set to true $\iff$ C_i is colored blue.

C_1 contains x and C_2 contains $\bar{x}$. Because x and $\bar{x}$ cannot both be true, v_1 and v_2 cannot both be red. Thus we add an illegal pair of colors for edge $\{v_1, v_2\}$: $\{(\mathbf{R}, \mathbf{R})\}$.

For simplicity, we rename $\mathbf{R} = 1, \mathbf{G} = 2, \mathbf{B} = 3$. Given CNF $\varphi = C_1 \wedge \cdots \wedge C_m$, we will construct a graph and lists of possible colors as follows:
$V = \{v_1, \ldots, v_m\}$.
$E = \{\{u, v\} \mid u, v \in V\}$.
$L_{\{u,v\}} = \{(a, b) \mid a \leq |C_u|, b \leq |C_v|, \ell_{u,a} \neq \neg \ell_{v,b}\}$.

Proof of corectness (sketch):

- Assume $x_1, \ldots, x_n$ is a satisfying assignment for φ, and in the graph, vertex v_i corresponds to $C_i = \ell_{i1} \vee \ell_{i2} \vee \ell_{i3}$.
  Let $\chi(v_i)$ be the first j such that $\ell_{i,j} = \text{TRUE}$. For each edge u, v colored $(\chi(u), \chi(v))$, $\chi(u) \leq |C(u)|, \chi(v) \leq |C(v)|$, and since $\ell_{i,\chi(u)} = \text{TRUE} = \ell_{v,\chi(v)}$, $\ell_{u,\chi(u)} \neq \ell_{v,\chi(v)}$. So edge (u, v) is colored correctly.
- Assume χ is a legal coloring of G.
  Set $x_i =$ true, if $\exists u$ s.t. $x_i = \ell_{u,\chi(u)}$
  false, if $\exists u$ s.t. $\neg x_i = \ell_{u,\chi(u)}$
  false, otherwise.
  If $\exists u, v$ such that $\ell_{u,\chi(u)} = x_i, \ell_{v,\chi(v)} = \neg x_i$ is satisfied, $(\chi(u), \chi(v)) \notin L_{\{u,v\}}$. By the assignment, every clause is satisfied.

1.2 From $\text{CSP}_{2,3}$ to 3-coloring

Gadgets for forbidden single color

We add a triangle whose vertices are v_R, v_G, v_B. Because the colors are interchangable, we will assume v_R is colored red, v_G is green and v_B is blue.

Similar as edges, define L_u to be the list of allowed colors for vertex u.

If we don't allow vertex u to be colored red, we connect u to v_R. (And we deal with the other two colors respectively)

Gadgets for edges forbidding a pair of different colors

Assume $(\mathbf{R}, \mathbf{G}) \notin L_{\{u,v\}}$. Then create path $u - a - b - v$, and let
$L_a = \{(\mathbf{R}, \mathbf{B})\}$,
$L_b = \{(\mathbf{G}, \mathbf{B})\}$.

If u is colored red, then a can only be blue, and then b is forced to be green, so v cannot be green. And vice versa: if v is green, u is forced not to be colored green.

If u is green, then a can be either red or blue, b can be either green or blue, so the color of v can be any of the three colors. If u is blue, then a is red, b can be either green or blue, so the color of v can be any of the three colors. So this gadget allow all other colors for (u, v) other than $(\mathbf{R}, \mathbf{G})$.

Gadgets for edges forbidding a pair of the same color

Assume $(\mathbf{R}, \mathbf{R}) \notin L_{\{u,v\}}$. Then create path $u - a - b - c - v$, and let
$L_a = \{(\mathbf{R}, \mathbf{G})\}$,
$L_b = \{(\mathbf{G}, \mathbf{B})\}$,

$L_c = \{(\mathbf{R}, \mathbf{B})\}$.

If both u and v are red, then a can only be green, and c can only be blue. Thus b should be neither be green or blue.

If there are more than one pairs of colors not in $L_{\{u,v\}}$, we can create more than one paths between (u, v). There can be at most 9 paths created, which is a constant. So the total number of new vertices and edges is linear to the input size.

Formally, given V, E, and L_e for all $e \in E$, we construct V', E' as follows:

1. Add V to V'.
2. Add a triangle $\{v_R, v_G, v_B\}$ to V'.
3. For every $e = \{u, v\} \in E$ and every $(c_1, c_2) \notin L_e$,
   (a) we add nodes a_{e,c_1,c_2}, b_{e,c_1,c_2} to V as a path from u to v, if $c_1 \neq c_2$.
   (b) we add nodes a_{e,c_1,c_2}, b_{e,c_1,c_2}, c_{e,c_1,c_2} to V as a path from u to v, if $c_1 = c_2$.
4. And connect these nodes to the forbid edges as in the gadgets.

We skip the proof of correctness.

2 Subset Sum

Subset Sum

- Instance: $a_1, \ldots, a_n, T \in \mathbb{Z}$.
- Solution: $S \subseteq \{1, \ldots, n\}$.
- Constraint: $\sum_{i \in S} a_i = T$.

We define a intermediate problem,

Vector Subset Sum

- Instance: $\vec{a}_1, \ldots, \vec{a}_n, \vec{T} \in \mathbb{Z}$.
- Solution: $S \subseteq \{1, \ldots, n\}$.
- Constraint: $\sum_{i \in S} \vec{a}_i = \vec{T}$.

2.1 Reduction from Vector Subset Sum to Subset Sum

(Sketch:) For each vector, concatenate the values of all dimensions to form a large number. To avoid overflow, we put $\log n$ bits of padding zero between the values.

2.2 Reduction from Independent Set to Vector Subset Sum

Given an instane of Independent Set problem $G = (V, E), k$, whose solution is $I \subseteq V$, $|I| = k$, satisfying $\forall e, |e \cap I| \leq 1$.

Let the dimension of vectors be $d = |E| + 1$.

For $i < d$, the i^{th}dimension is to make sure we don't put both endpoints of edge i into the independent set. For each vertex a, we create a vector v_a, and put a 1 on the dimensions for each of its incident edges.

The last dimension counts the independent set size. For each vertex vector, always put a one on that dimension.

Finally, because the sum on each dimension are allowed to be either 0 or 1, we create a dummy vector for each edge so that each dimension can sum to exactly 1.

1. For every u, $v_u[j] = 1$ if $u \in e_j$, or $j = |E| + 1$.
2. For every e, $v_e[j] = 1$ if $e = e_j$, and 0 otherwise.
3. $T = (1, \ldots, 1, k)$.

Proof of correctness (sketch):

- Say that $I \subseteq V$ is an independent set of size k. We let $S = \{u \mid u \in I\} \cup \{e \mid |e \cap I| = 0\}$. Then $\sum_{u \in I} \vec{v}_u = (|e_j \cap I|, \ldots, |I|)$. $\sum_{u \in I} \vec{v}_u + \sum_{e, |e \cap I| = 0} \vec{v}_e = (1, 1, \ldots, 1, k) = \vec{T}$.
- Assume we have a subset S s.t. $\sum_{i \in S} \vec{v}_i = \vec{T}$. Then we let $I = \{u \mid u \in S\}$.