

## I 15 - Vertex Cover is NP-Complete

对于一个 NP-C 问题，它需要：

1. 该问题必须在 NP 中，也就是说，它有一个多项式时间可验证的解决方案
2. 该问题必须是 NP-hard 的，也就是说，NP 中的每个问题都可以有效地归约到它

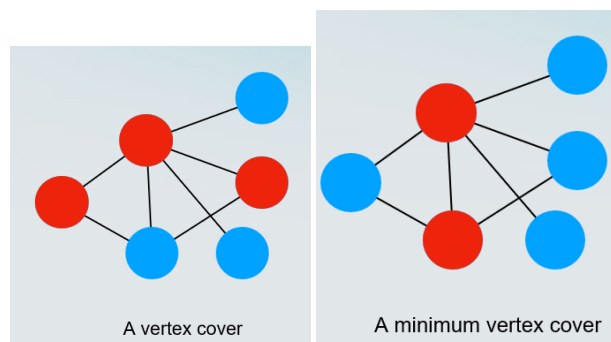
### I 顶点覆盖 (Vertex Cover)

图  $G = (V, E)$  的顶点覆盖  $C$  是顶点的一个子集，使得图中的每条边  $e$  至少有一个端点在  $C$  中

最小顶点覆盖是顶点覆盖中顶点数最少的一个

顶点覆盖问题：

- 输入：一个图
- 输出：一个最小的顶点覆盖



定点覆盖问题等价的判定形式 (decision version) :

- 输入：一个图  $G$  与一个数字  $k$
- 输出：是否存在一个小于  $k$  的顶点覆盖？

### I P/NP

- 顶点覆盖问题在 NP 中
  - 假设我们有一个顶点覆盖
    - 我们可以在多项式时间内检查其大小为  $k$ ，并验证它确实是一个顶点覆盖
- 顶点问题是 NP-Hard 的
  - 我们将从 3-SAT 构造一个多项式时间的归约
    - 即，我们将证明 3-SAT 可以多项式归约到顶点覆盖

### I 归约

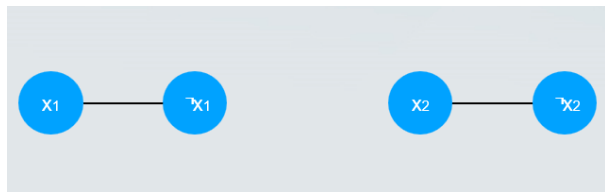
设  $\varphi$  是一个具有  $m$  个子句和  $d$  个变量的 3-CNF 公式

我们在多项式时间内构造一个顶点覆盖问题的实例  $\langle G, k \rangle$ ，使得

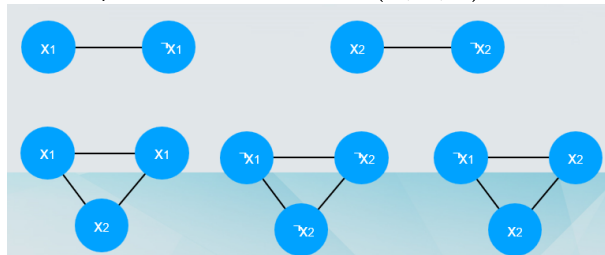
- 如果  $\varphi$  是可满足的  $\Rightarrow G$  具有大小至多为  $k$  的顶点覆盖
- 如果  $\varphi$  是不可满足的  $\Rightarrow G$  不具有大小至多为  $k$  的顶点覆盖

$$\text{令 } \varphi = (x_1 \vee x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_2)$$

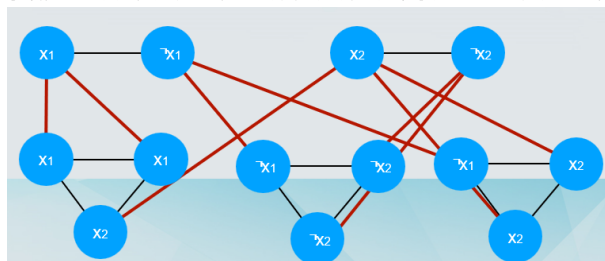
对于每个在  $\varphi$  中的变量  $x$ ，我们可以在图  $G$  中创建两个节点  $x$  和  $\neg x$ ，并用一条边  $e = (x, \neg x)$  将他们连接起来



对于在  $\varphi$  中的每个 clause  $l = (l_1, l_2, l_3)$ ，我们可以在  $G$  中创建三个节点  $l_1, l_2, l_3$ ，并将它们两两相连



我们在顶部和底部的所有具有相同标签的节点之间添加一条边



## 其中一个方向

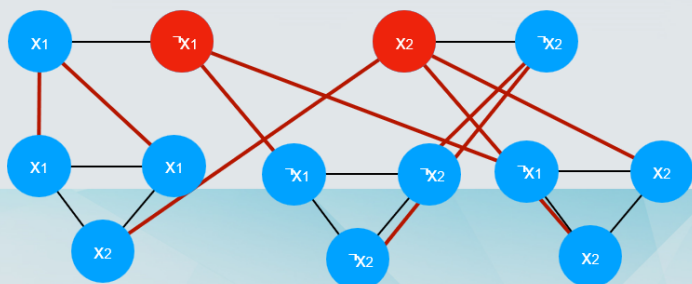
如果  $\varphi$  是可满足的  $\Rightarrow G$  具有大小至多为  $k$  的顶点覆盖

设  $(y_1, \dots, y_d) \in \{0, 1\}^d$  为满足公式  $\varphi$  的一个赋值

- $y_i$  代表了  $\varphi$  的第  $i$  个变量的真值赋值
- **对于顶部的节点**：如果  $y_i = 1$ ，将节点  $x_i$  包含在顶点覆盖  $C$  中；否则包含节点  $\neg x_i$
- **对于底部的节点**：在每个三角形中，选择在顶部已经选中的节点  $x_i$  并且不将其包含在顶点覆盖中。将另外两个节点包含在顶点覆盖中
- **顶部和底部之间的每条边**：都与  $C$  中的某个节点相连；如果它在底部没有与任何节点相连，那是因为它在顶部被选择时跳过了

- For the nodes on the top: If  $y_i = 1$ , include node  $x_i$  in the vertex cover  $C$ , otherwise, include node  $\neg x_i$ .

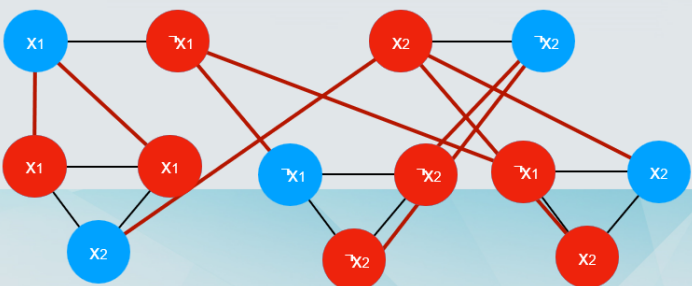
- Assume  $y_1 = 0, y_2 = 1$ .



Running example:  $\varphi = (x_1 \vee x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_2)$

- For the nodes on the bottom: In each triangle, choose a node  $x_i$  that has been picked on the top and do not include it in the vertex cover. Include the other two nodes.

- Assume  $y_1 = 0, y_2 = 1$ .



Running example:  $\varphi = (x_1 \vee x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_2)$

顶点覆盖的大小为  $k = d + 2m$ ,  $d$  为变量数量,  $m$  为 clause 数量

- 每个变量在顶部被选择
- 对于每个 clause, 我们在底部选择两个节点

## 其他方向

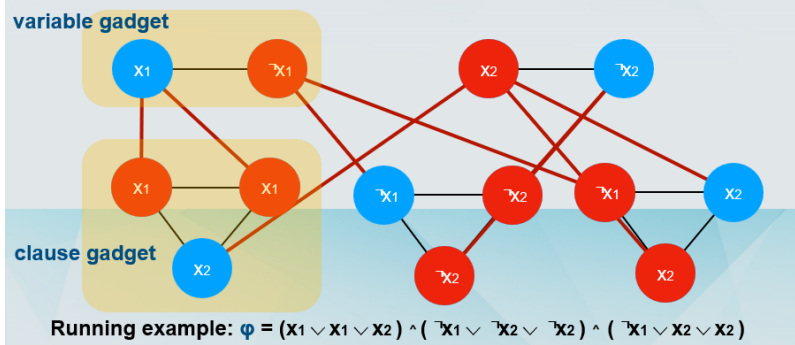
- 如果  $\varphi$  是不可满足的  $\Rightarrow G$  不存在大小至多为  $k$  的顶点覆盖
- 如果  $G$  存在大小至多为  $k$  的顶点覆盖  $\Rightarrow \varphi$  是可满足的

如果  $G$  存在大小至多为  $k$  的顶点覆盖  $\Rightarrow \varphi$  是可满足的

设  $C$  为图  $G$  中大小为  $k = d + 2m$  的顶点覆盖

- 因为他是一个顶点覆盖, 所以在每个 clause gadget (小三角形——中, 必须选择三个节点中的至少两个节点
  - 这意味着  $C$  中至少有  $2m$  个节点在底部
  - 这进一步意味着  $C$  中最多有  $d$  个节点在顶部
- 为了满足顶部的边, 在每个 variable gadget 中, 至少有一个节点必须包含在顶点覆盖  $C$  中
  - 为了满足顶部和底部的边, 在每个 variable gadget 中, 至少有一个节点必须包含在  $C$  中
  - 因此, 在每个 variable gadget 中, 必须恰好包含一个节点在  $C$  中

- To satisfy the edges at the top, in each “variable gadget”, at least one node must be included in  $C$ .



- 考虑与顶点覆盖  $C$  顶部节点对应的真值赋值
  - 要么  $x_1$  为 1；要么  $\neg x_i$  为 1
  - 这是因为在每个 variable gadget 中，必须只包含一个在  $C$  中的节点
- 由于所有的交叉边都被覆盖， $C$  中必须有一个在顶部的端点
  - 这意味着 clause 的一个变量被设置为 1
  - 因此这个 clause 是满足的