算法基础

第九讲：算法复杂度理论

主　讲：顾乃杰　教授
单　位：计算机科学技术学院
学　期：2015—2016 (秋)

34.NP完全性

- 多项式时间
- NP完全性与可归约性
- NP完全性的证明
- NP完全问题

34.1多项式时间

- 有些问题很棘手:
  as they grow large, we are unable to solve them in reasonable time
- 合理的时间的构成？标准工作定义：*多项式时间*
  - On an input of size n the worst-case running time is $O(n^k)$ for some constant k
  - Polynomial time: $O(n^2)$, $O(n^3)$, $O(1)$, $O(n \lg n)$
  - Not in polynomial time: $O(2^n)$, $O(n^n)$, $O(n!)$

P和NP

- 最短与最长路径
- 欧拉回路与哈密顿回路
- 2-CNF (Conjunctive normal form, 合取范式) 可满足性和3-CNF可满足性

P和NP

- P类中包含的是在多项式时间内可以解决的问题。
- NP类中包含的是在多项式时间内“可验证”的问 题。是指给定某一解决方案的“证书”,就能在 问题输入规模的多项式时间内，验证该“证书” 是正确的。
- P中任何问题都属于NP,因为若某一问题属于P, 则可以在不给出证书的情况下，在多项式时间内解决它。
- 通俗地说，如果一个问题属于NP,且与NP中的 任何问题一样“难”的，则说它属于NPC类，即 NP完全的(NP Complete)。

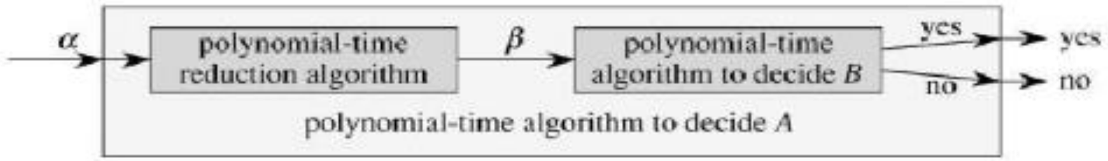
判定问题和最优化问题

- Optimization problem
- Decision problem

- NP完全性不直接适合于最优化问题，但适用于判定问题，因为这种问题的答案是简单的“是”或“否”。

34.3 NP完全性与可归约性

- Reduction
- 考虑一个判定问题(A),希望在多项式时间解决该问题。 称某一特定问题的输入为该问题的一个实例(instance)。 现在，假设另一个不同的判定问题(B),我们知道如何 在多项式时间内解决它。最后，假设有一个过程，它能将 A的任何实例a转换为B的、具有如下特征的某个实例b:



- (1)转换操作需要多项式时间;
- (2)两个实例的答案是相同的。即a的答案若是“yes”, 则b的答案亦是“yes”。
- 称这样一种过程为多项式时间的归约算法(Reduction Algorithm),并且，它提供了一种在多项式时间内解决问题A的方法。

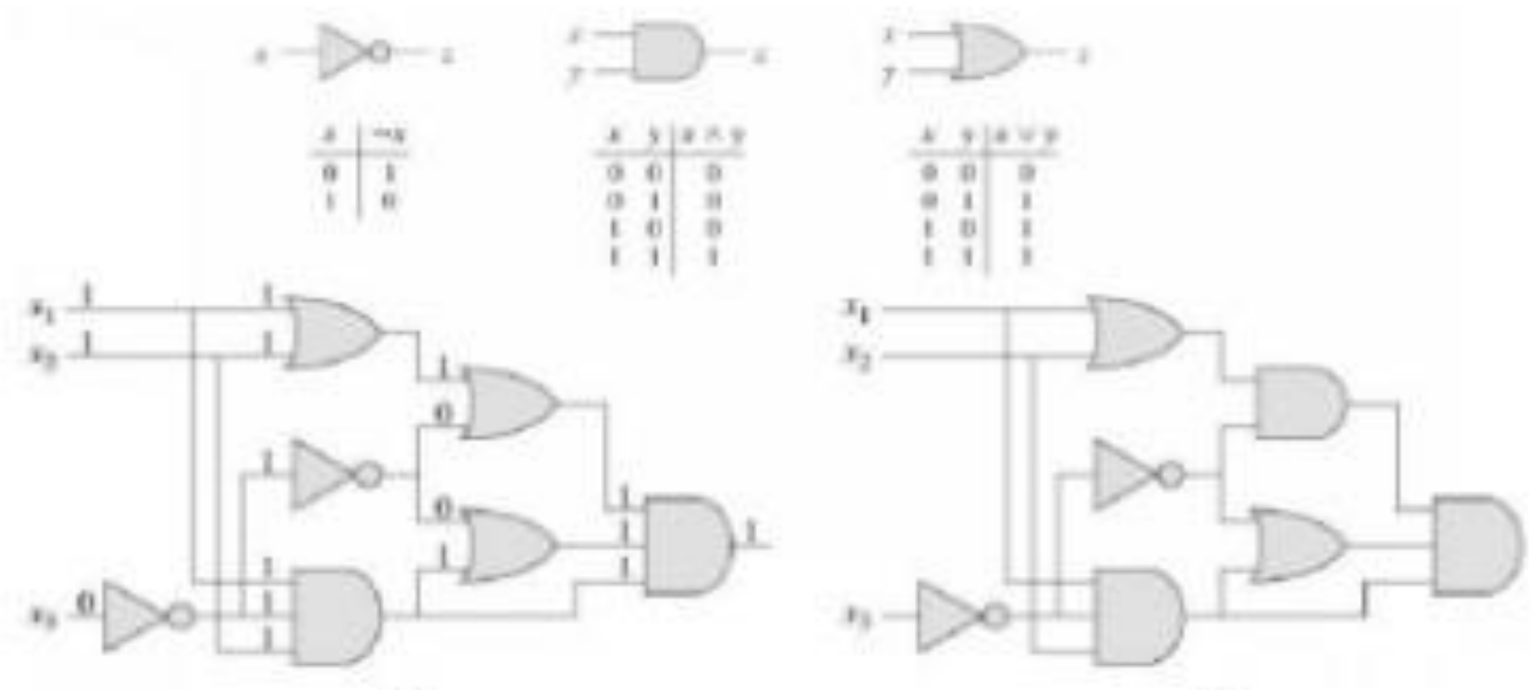
电路可满足性

- 电路可满足性问题：给定一个由“与”, “或”,“非”门组成的布尔组合电路， 它是可满足电路吗？

- 回答：Yes / No

电路可满足性

- 电路可满足性问题(Circuit satisfiability problem)。已知 一个布尔组合电路，它由And、Or、Not门组成，我们希 望知道这个电路是否存在一组布尔输入，是的它的输出为1.



(a) (b)

34.4 NP完全性的证明

- 考虑一个判定问题A,若我们已知其不存在 多项式时间算法。进一步假设有一个多项 式时间规约，它将一个A的实例转换成B的 实例，则B一定不存在多项式时间算法。

- 对NP完全性的证明亦是如此。

证明C-SAT是NP完全(1)

- 引理34.5:电路可满足性问题属于NP类。
- 证明：Circuit-SAT可在多项式验证，因此 属于NP类。

证明C-SAT是NP完全(2)

- 证明Circuit-SAT是NP完全问题的第二部分，就是 要证明该语言是NP难度的，即必须证明NP中的 每一种语言，都可以在多项式时间归约为CircuitSAT。教材上基于计算机硬件的工作机理，给出 了一个简要的证明过程。
- 引理34.6:C-SAT问题是NP难度的。
- 证明：C-SAT至少与NP中的任何语言具有同 样的难度，并且它又是属于NP的，因此它是NP 完全的。
- 定理34.7:C-SAT问题是NP完全的。

NP完全性的证明

- 引理34.8:
- If L is a language such that $L' \le_P L$ for some $L' \in$ NPC, then L is NP-hard. Moreover, if $L \in$ NP, then $L \in$ NPC
- In other words, by reducing a known NP-complete language L' to L, we implicitly reduce every language in NP to L. Thus, Lemma 34.8 gives us a method for proving that a language L is NP-complete:
- Prove $L \in$ NP.
- Select a known NP-complete language L'.

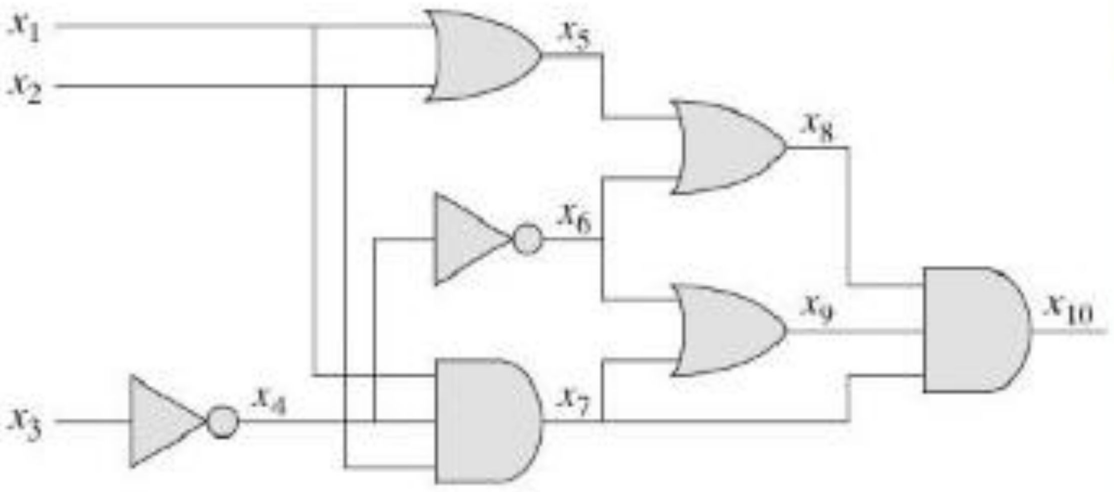
NP完全性的证明

- Describe an algorithm that computes a function f mapping every instance $x \in \{0, 1\}^*$ of L' to an instance $f(x)$ of L.
- Prove that the function f satisfies $x \in L'$ if and only if $f(x) \in L$ for all $x \in \{0, 1\}^*$.
- Prove that the algorithm computing f runs in polynomial time.
- (Steps 2-5 show that L is NP-hard.) Thismethodology of reducing from a single known NP-complete language is far simpler than the more complicated process of showing directly how to reduce from every language in NP.

公式可满足性

- 电路可满足性问题的任何实例可以在多项式时间内，归约为公式可满 足性问题的一个实例。利用归纳法可以将任意布尔组合电路表示为一 个布尔公式。例如对下图的布尔电路进行归约。
- CIRCUIT-SAT $\leq$P SAT
- 为什么仅当公式φ可满足时，电路C才是可满足的呢？因为若φ是可满 足的，则φ中每个子句都取得1值，这种赋值同样使得电路中的每个门 的输出取得1值，因此C也是可满足的。

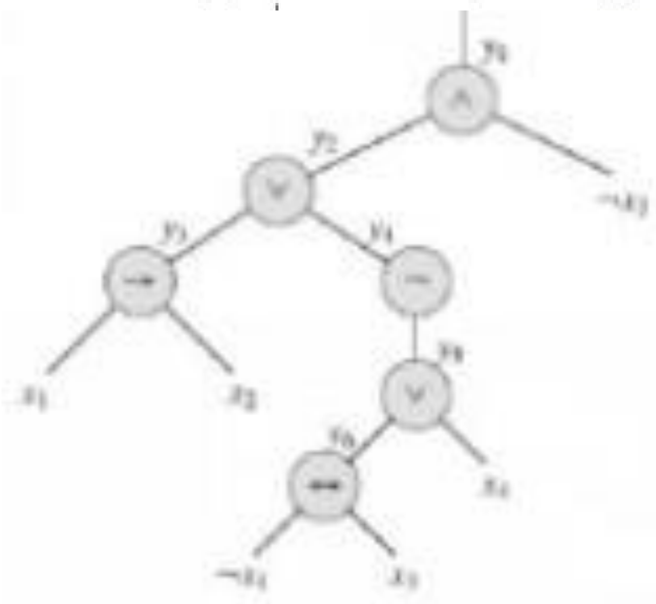
公式可满足性



$$
\begin{aligned}
\varphi = x10 &\wedge (x4 \leftrightarrow \neg x3)\\
&\wedge (x5 \leftrightarrow (x1 \vee x2))\\
&\wedge (x6 \leftrightarrow \neg x4)\\
&\wedge (x7 \leftrightarrow (x1 \wedge x2 \wedge x4))\\
&\wedge (x8 \leftrightarrow (x5 \vee x6))\\
&\wedge (x9 \leftrightarrow (x6 \vee x7))\\
&\wedge (x10 \leftrightarrow (x7 \wedge x8 \wedge x9)).
\end{aligned}
$$

3-CNF可满足性

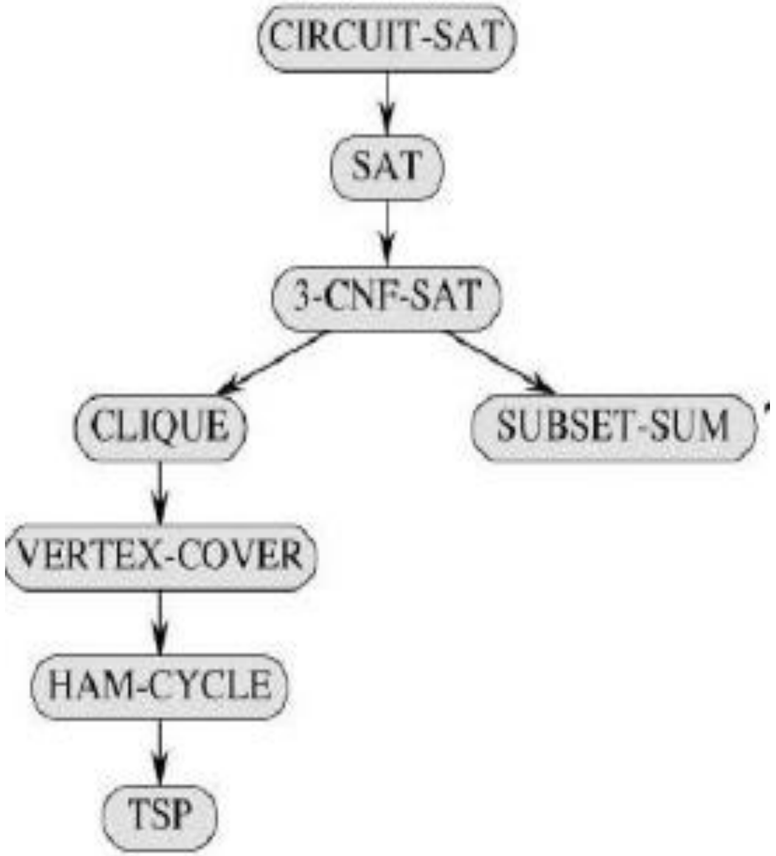
- (1)为公式构造一棵二叉语法分析树。
- (2)把树看做电路，构造公式。
- (3)利用真值表将公式的子句变换为析取式。
- 例如：　$\varphi = ((x1 \rightarrow x2) \vee \neg((\neg x1 \leftrightarrow x3) \vee x4)) \wedge \neg x2.$



$$
\begin{aligned}
\varphi' = y1 &\wedge (y1 \leftrightarrow (y2 \wedge \neg x2)) \\
&\wedge (y2 \leftrightarrow (y3 \vee y4)) \\
&\wedge (y3 \leftrightarrow (x1 \rightarrow x2)) \\
&\wedge (y4 \leftrightarrow \neg y5) \\
&\wedge (y5 \leftrightarrow (y6 \vee x4)) \\
&\wedge (y6 \leftrightarrow (\neg x1 \rightarrow x3))
\end{aligned}
$$

34.5NP完全问题

- 团(Clique)
- 点覆盖(Vetex Cover)
- 哈密顿回路
- 旅行商问题(Traveling-salesman problem)
- 子集和问题(Subset-sum problem)



团问题

- 无向图 G = (V, E) 中的团Clique是一个节点集 V‘ V 其中每一对节点间都有边相连。即团是图 G 的一个完全子图。 团问题是寻找图中的最大团。 其判定问题是：在给定图中是否存在一个规模为k 的团。
- 一个朴素的算法就是列出V 的所有 k-子集然后检 验它们是否构成团。其时间复杂度为? 若 k 为常 数，则它是多项式时间，但 k 可能接近 |V| /2,这 样一来算法的运行时间就是超多项式时间，因此， 人们猜想团问题不大可能存在有效算法。

团问题

定理34.11:团问题是NP完全的。

证明:

(1) 团问题显然是多项式可验证的。

(2)从 3-CNF-SAT开始归约。设$\varphi = C1 \wedge C2 \wedge \ldots \wedge Ck$ 是一个具有 k 个子句的3-CNF布尔公式。我们根据公式φ构造一个图G。 图 G = (V, E) 构造如下，对φ中每个子句Cr=(l1Vl2Vl3),将3个顶点v1,v2,v3放入V中。若一些两个条件同时满足，则用一条边连接两个顶点vi和vj。

vi和vj处于不同的三元组中，即 $r \neq s$;

团问题

- 它们的相应文字是一致的，即li不是lj的非。

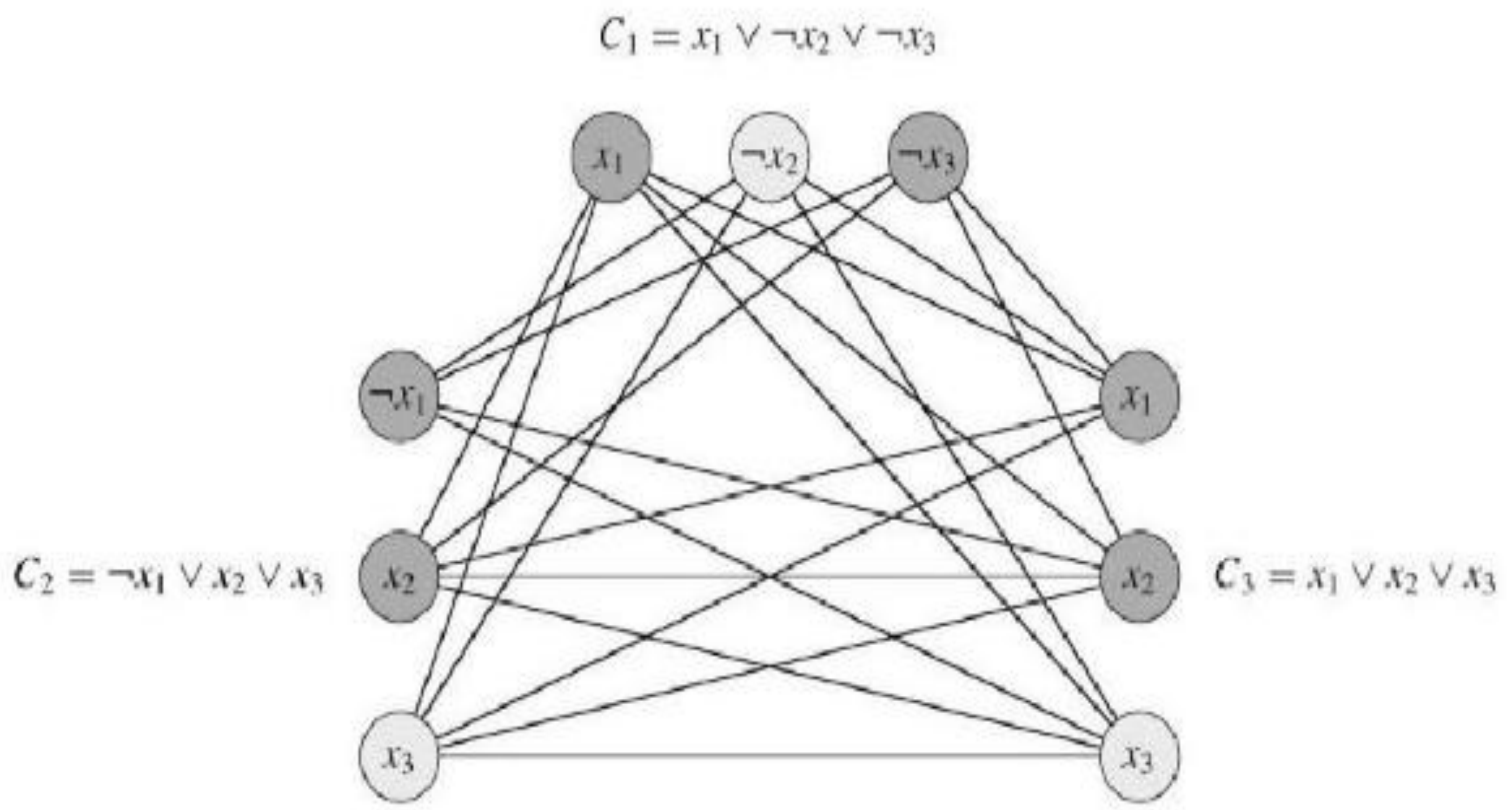
This graph can easily be computed from φ in polynomial time. As an example of this construction, if we have

$\varphi=(x1 \vee \neg x2 \vee \neg x3) \wedge (\neg x1 \vee x2 \vee x3) \wedge (x1 \vee x2 \vee x3)$,

then G is the graph shown in Figure 34.14.

- $\varphi=(x1 \vee \neg x2 \vee \neg x3) \wedge (\neg x1 \vee x2 \vee x3) \wedge (x1 \vee x2 \vee x3)$

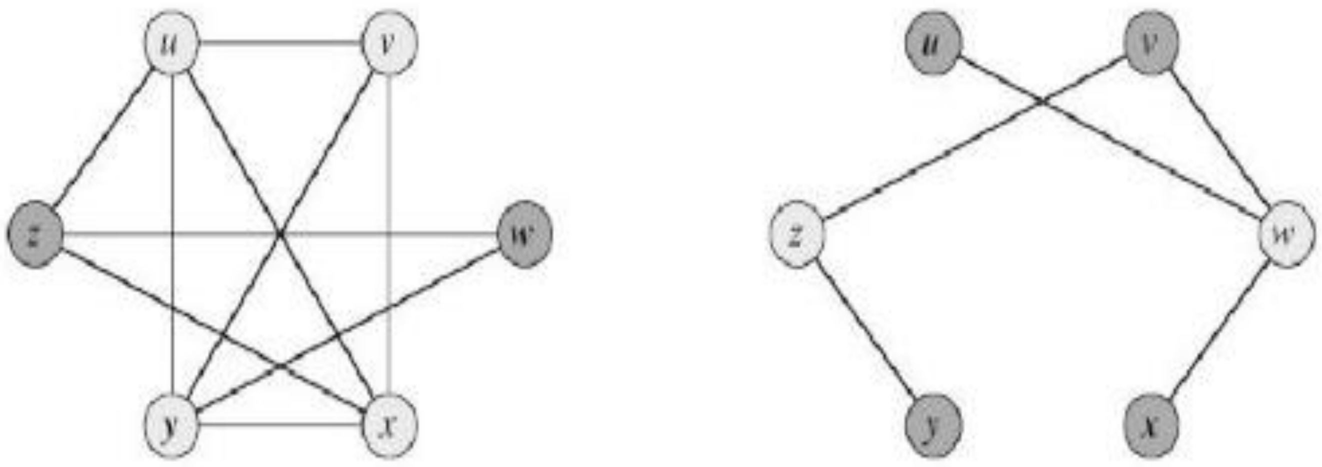
团问题



$C_1 = x_1 \vee \neg x_2 \vee \neg x_3$
x_1
$\neg x_2$
$\neg x_3$
$\neg x_1$
x_2
x_3
$C_2 = \neg x_1 \vee x_2 \vee x_3$
x_1
x_2
x_3
$C_3 = x_1 \vee x_2 \vee x_3$

顶点覆盖问题

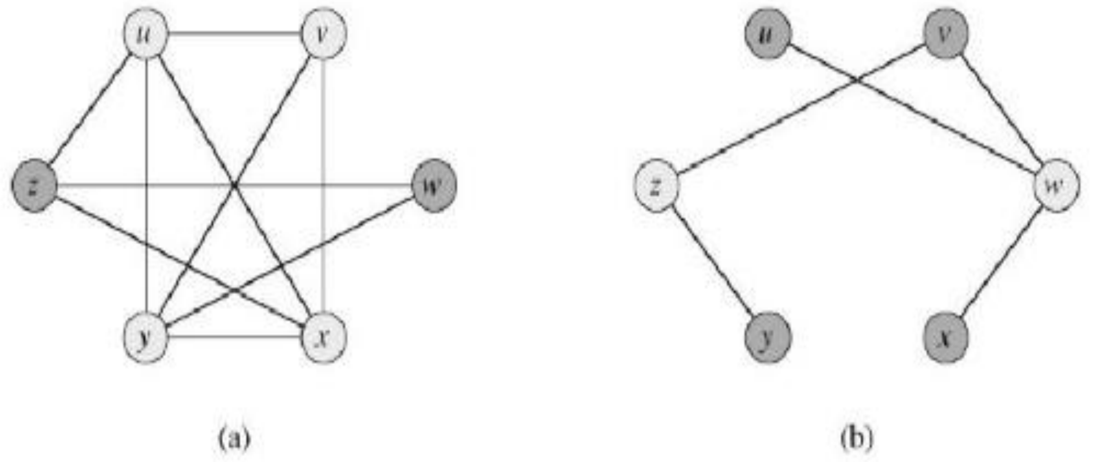
- 无向图 G = (V, E) 的点覆盖是一个节点集 V’ ⊆V ,满足若 (u, v) ∈ E,则 u ∈ V’ 或 v ∈ V’。 例如 图 (b) has a vertex cover {w, z} of size 2



(a) (b)

顶点覆盖问题

- 定理34.12:点覆盖问题是NP完全的。
- 证明：定义补图如下：图 G 的补图就是G’是包含不在G中的那些边的图。下图展示了图 G 和它的补图G’。



顶点覆盖问题

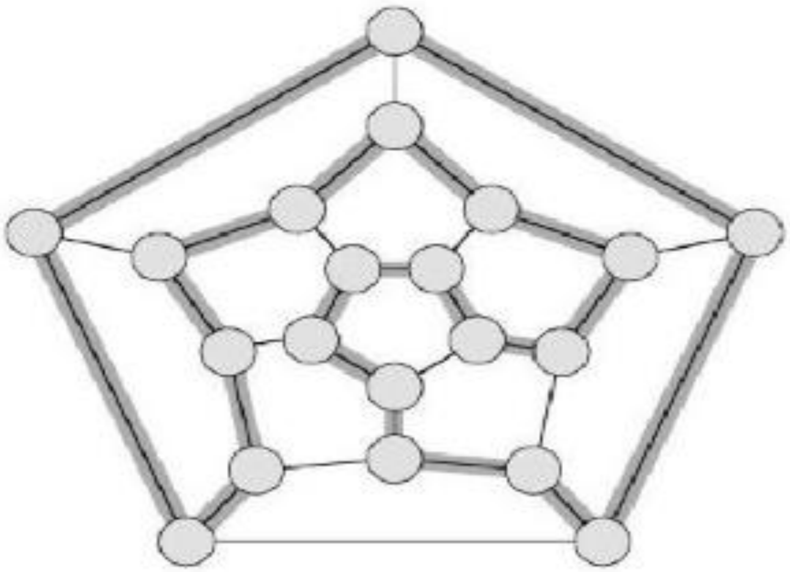
- 归约：图 G 有一个大小为 k 的团，当且仅当其补图G’有一个大小为 |V| - k 的点覆盖。
- 证明：假设G 包含一个团 V’ ⊆ V ,且 |V’| = k。我们可以断言 V - V’ 是 G’ 的一个点覆盖。设 (u, v) 是 E’ 中的任意边。则(u, v) ∉E, 这意味着 u 或 v 至少有一个不属于团 V’ ,因为 V’ 中的每一对节点间都有 一条E中的边相连，等价地， u 或 v 中至少有一个属于V - V’ 此即边 (u, v) 被 V - V’所覆盖。由于 (u, v) 是从E’中任选的边，因此E’被 V - V’ 所覆盖。

顶点覆盖问题

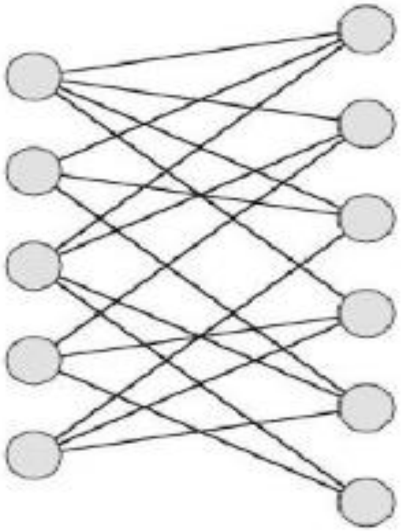
- Conversely, suppose that G’ has a vertex cover V‘⊆ V, where |V’| = |V| - k. Then, for all u, v ∈ V, if (u, v) ∈ ē, then u ∈ V‘ or v ∈ V’ or both. The contrapositive of this implication is that for all u, v ∈ V, if u ∉ V‘ and v ∉ V', then (u, v) E. In other words, V -V' is a clique, and it has size |V |-|V'| = k.

哈密顿回路问题

- 哈密顿回路(hamiltonian-cycle)



(a)



(b)

哈密顿回路问题

- 定理34.13:哈密顿回路问题是NP完全问题。

1.证明*HC*是NP问题。

HC的证书为一顶点序列$\langle v_0, v_1, \ldots, v_n, v_0\rangle$，我们需要

(1)检查是否包含图中所有的顶点，且顶点个数是否为V+1，其中起始点出现两次；

(2)对于顶点序列中(v_i, v_{i+1}) $(0\le i<n)$和(v_n, v_0)，检查是否是图中的边。

显然是多项式时间验证HC。

哈密顿回路问题

2. 规约如下：

因为我们是从顶点覆盖问题规约到汉密尔顿回路问题，而顶点覆盖的实例为：

G=(V, E) 和整数k,我们需要将这些转化为汉密尔顿回路的实例G'=(V',E')。

(1)对于G中的每条边，在G'中都构造一个检验分量（附件图，如图1所示），用来检验G中这条边是否被覆盖到。

对于这个附件图有一些约束：

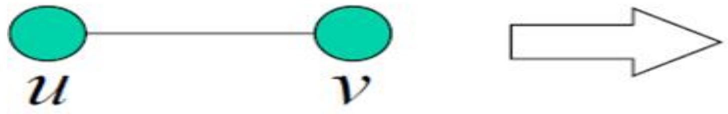
- 能够与外界连接的点只有[u, v, 1], [u, v, 6], [v, u, 1], [v, u, 6]。
- 如果从[u, v, 1]进去，则必须从[u, v, 6]出来；从[v, u, 1]进去，则必须从[v, u, 6]出来。
- 通过此附件图时只有两种路线，如图2所示。
  - 当u在顶点覆盖，而v不在顶点覆盖中，则依照b) 图所示通过。
  - 当u和v都在顶点覆盖中，则依照c) 图所示通过。
  - 当v在顶点覆盖，而u不在顶点覆盖中，则依照d) 图所示通过。

(2)对于G中的每个点u，将与u相邻的点进行任意排列，并记作: $u^{(1)}, u^{(2)}, \ldots, u^{(\deg[u])}$；且$1 \le i < \deg[u]$时，$u^{(i)}$与$u^{(i+1)}$连一条线，如图3所示。

(3) 创建k个选择器顶点$s_1, s_2, \ldots, s_k$，这些选择器顶点用来被当做G中的各个顶点覆盖的点。

(4) 对于G中的每个点u, 则用$[u, u^{(1)}, 1]$与每个选择器顶点连线，$[u, u^{(\deg[u])}, 6]$与每个选择器顶点连线。

哈密顿回路问题



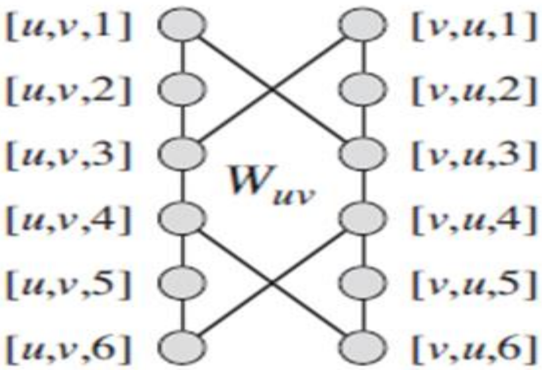


图1

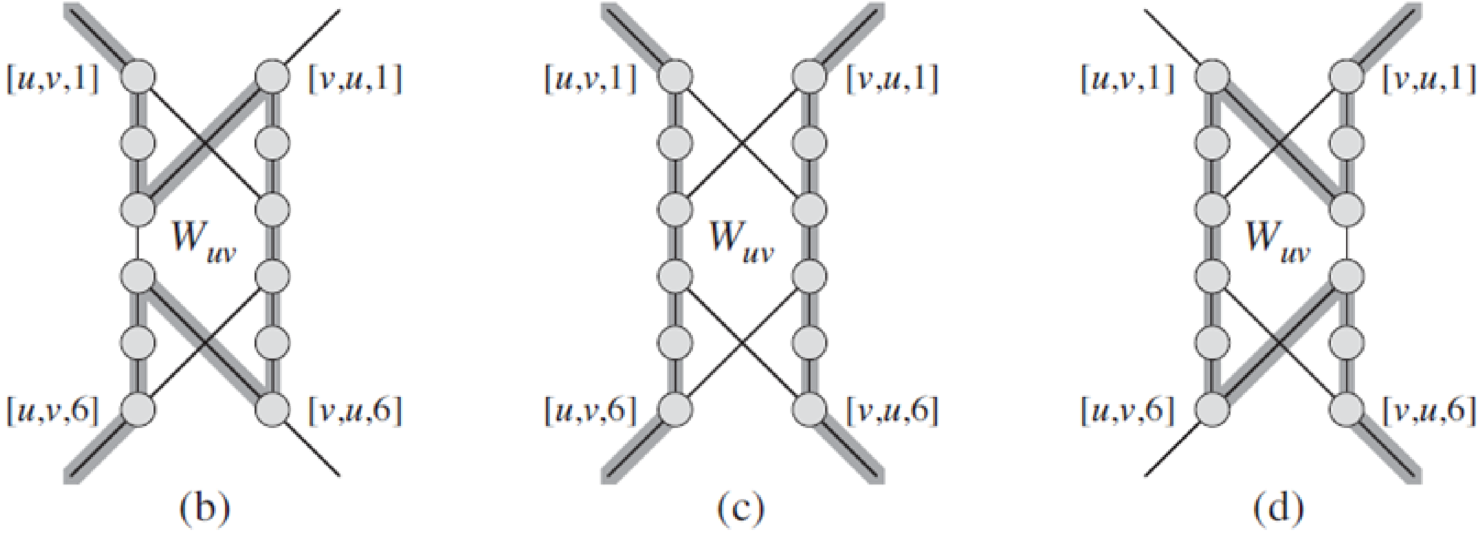
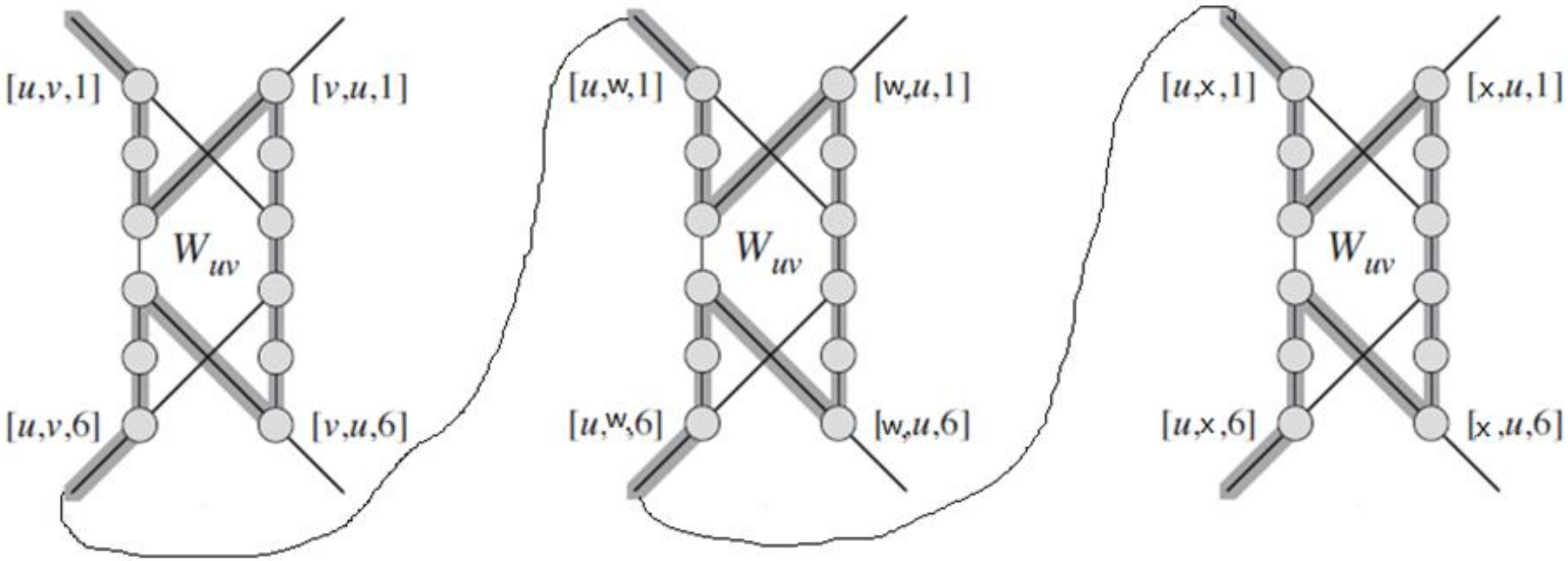


图2

哈密顿回路问题



例子：假设u与v,w,x相邻　　　　**图3**

哈密顿回路问题

3.证明是多项式的规约。

因为在G’中有k+12E个顶点，G’中有14E+∑（$\deg[u]-1$）(各覆盖检验分量之间的边)+2kV(与选择器顶点的连接边)=14E+2E-V+2kV,显然是多项式规约。

4.假设G中有一个大小为k的顶点覆盖S’=<$s_1', s_2', \ldots, s_k'$>，则这些点一定能够覆盖G中所有边，设汉密尔顿回路为C，则我们将：

$s_1=s_1'$，$s_2=s_2', \ldots, s_k=s_k'$，则我们将[$s_1', s_1^{(1)}$,1]与选择器顶点$s_1$的边加入C，且将[$s_1', s_1^{(1)}$,1]到[$s_1', s_1^{(\deg[u])}$,6]的路线（$s_1'$覆盖了相关联的边）加入到C（经过附件图的路线根据两点是否在顶点覆盖中而不同）。

最后将[$s_1', s_1^{(\deg[u])}$,6]到s_2的边加入到C中，以此类推，即[$s_2', s_2^{(1)}$,1]与选择器顶点s_2的边加入C，...，直到[$s_k', s_k^{(\deg[u])}$,6]与s_1的边加到C中，

因为S’是一个顶点覆盖，因此C会经过所有的附件图，因此C是一个汉密尔顿回路。

5.假设G’中有一个汉密尔顿回路C，则将C中每个选择器顶点连向的[u, $u^{(1)}$, 1]的点u加入到S中，此时集合S就是G的顶点覆盖，下面来证明。

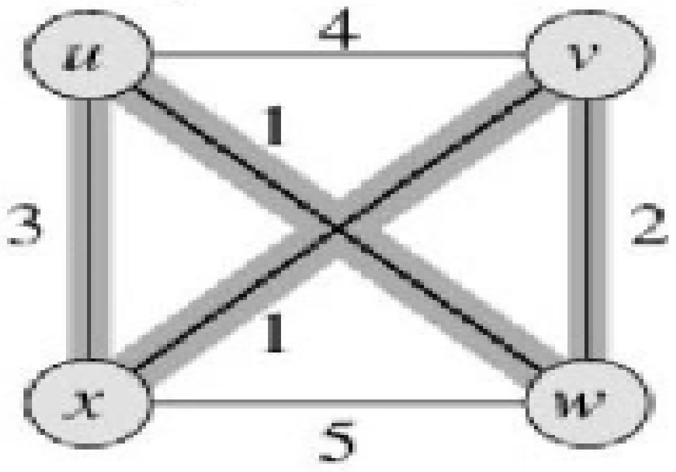
k个选择器顶点会将C的路径分为k段，对于每段路径都是从某个选择器顶点到另一个选择器顶点且不经过其他选择器顶点的路径，

而这条路径经过的附件图在G中对应的边就是G中的某个顶点v相关联的边，因此k段路径共对应G中k个不同的顶点，

因为汉密尔顿回路经过了所有附件图的点，因此表明这k个顶点覆盖了G中所有的边，即G中有大小为k的顶点覆盖。

旅行商问题

- traveling-salesmanproblem与hamiltonian-cycle问题密切相关。一个旅行商要访问n个城市，用有n节点的完全图表示城市间的道路，旅行商希望进行一次巡回旅行，即一个哈密顿回路。从城市i到城市j的费用为 c(i, j) ,旅 行商希望整个旅行费用最低。下图中的一个最小 费用线路为〈u, w, v, x, u〉其费用为 7。



旅行商问题

- TSP判定问题的 形式化描述如下：　TSP = { 〈G, c, k〉 : G = (V, E) 是一个完全图，c 是 V × V → Z上的一个函数，　k ∈ Z,且G 包含一个费用最多为 k 的旅行商的旅行回路 }。

旅行商问题

- 定理34.14:旅行商问题是NP完全的。
- 证明：HAM-CYCLE ≤P TSP
- 设 G = (V, E) 是 HAM-CYCLE的一个实例。我们可构造一个TSP实例 如下，构造一个完全图 G‘ = (V, E’),其中 E’= {(i, j) : i, j ∈ V and i ≠ j },并定义费用函数 c 如下：

$$c(i,j)=\begin{cases}0 & 若(i,j)\in E\\ 1 & 若(i,j)\notin E\end{cases}$$

- (由于 G 是没有自回路的无向图，因此 c(v, v) = 1 ,对所有 v ∈ V.)于 是 (G’, c, 0)就是TSP的一个实例，这可以在多项式时间确定。

旅行商问题

- 现在来说明 G 有一个哈密顿回路，当且仅当 G’有一个费用至多为0的 回路。假设 G 有一个哈密顿回路h。 h 中的每条边都属于 E ,因此在 G’ 中的费用为 0 。因此 h 在G’中是费用为 0的回路；反之，假设图G’ 有一个费用至多为0的回路h’。由于 E’ 中的边的费用只能是 0 或 1, 因此回路h’ 的费用就是 0 ,且回路上每条边的费用必为 0。因此， h’ 仅包含 E 中的边，所以可得 h‘ 是 G.的一个哈密顿回路。

子集和问题

- Subset sum problem:给定一个整数集A(俗称为背包)和整数b,要求找出A的一 个子集，使得其中元素之和等于b。

子集和问题

- 定理34.15:子集和问题是NP完全的。

（证明参考教材P627页）

The End！