

NP完全性理论

—计算复杂性理论

1 图灵机

- **可解性**: 问题及其可解性可用函数和可计算性来代替
- **可计算性理论**: 研究计算的一般性质的数学理论, 它通过建立计算的数学模型 (例如抽象计算机) , 精确区分哪些是可计算的, 哪些是不可计算的。
- **可计算函数**: 能够在抽象计算机上编出程序计算其值的函数。这样就可以讨论哪些函数是可计算的, 哪些函数是不可计算的
- **丘奇-图灵论题**: 若一函数在某个合理的计算模型上可计算, 则它在图灵机上也是可计算的。
- **不可计算性**: 很多问题和函数是无法用具有有穷描述的过程完成计算

1 图灵机

艾伦·麦席森·图灵(1912.6.23-1954.6.7)，
英国计算机科学家、数学家、逻辑学家、
密码分析学家、理论生物学家，**计算机
科学之父、人工智能之父**，英国皇家学
会院士。



1935年当选为剑桥大学国王学院研究员；1936年提出被称为**图灵机**的逻辑
机通用模型（发表论文：**论可计算数及其在判定问题上的应用**）；1938年
获普林斯顿大学博士学位；1939年开始在英国军方工作，期间**破解德国密
码系统恩尼格玛密码机和金枪鱼密码机**，加速了盟军取得了二战的胜利；
1946年获大英帝国勋章；1945年-1948年在伦敦泰丁顿国家物理实验室负
责自动计算引擎（ACE）的研究工作；1948年任曼彻斯特大学高级讲师、
自动数字计算机（Madam）项目的负责人助理；1949年任曼彻斯特大学
计算机实验室副主任；1950年提出机器具备思维的可能性和“**图灵测试**”
的概念（发表论文：**计算机器与智能**）；1951年当选为英国皇家学会院士；
1954年服用含氰化物的苹果去世，享年41岁。

1 图灵机

1.1 图灵机

停机问题：能否写一个程序正确判定输入给它的任何一个程序是否会停机？

设程序 $\text{halts}(P, X)$ 总是正确地判定程序 P 在其输入 X 上是否停机：若停机，则返回 yes；否则死循环，返回 no。设另有一程序：

```
diagonal(Y){  
    a: if halts(Y,Y) then  
        goto a;  
    else halt;  
}
```

功能：若 halts 判定当程序 Y 用其自身 Y 作为输入时 Y 停机，则 $\text{diagonal}(Y)$ 死循环；否则它停机

$\text{diagonal}(\text{diagonal})$ 是否停机？**不可判定**

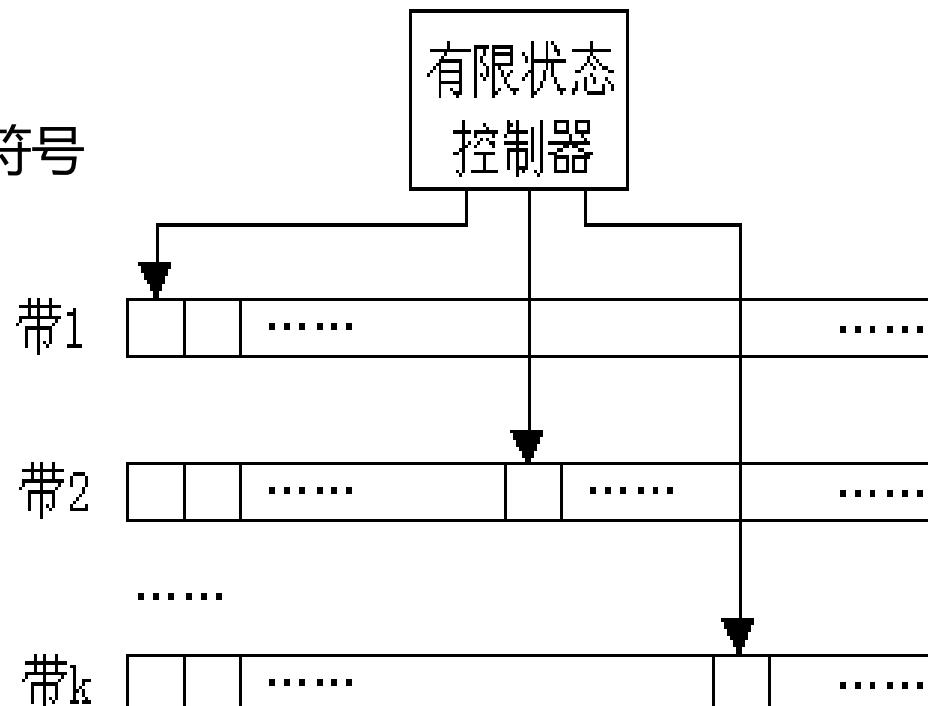
它停机当且仅当 $\text{halts}(\text{diagonal}, \text{diagonal})$ 返回否，也就是：
 diagonal 停机当且仅当它自己不停机，矛盾！

即： $\text{halts}(P, X)$ 并不存在，停机问题是不可解的！

1 图灵机

1.1 图灵机

- 无限长的纸带
 - 向左向右无限延伸
 - 每个格子存放字母表的符号
- 读写头
 - 指向纸带当前位置
 - 读取、写入、移动
- 有限状态控制器
 - 状态寄存器：
初始、中间、停机
 - 转移函数：
规则表、程序、指令集合



有单带、多带等变种，计算能力等价

1 图灵机

1.1 多带图灵机

根据有限状态控制器的当前状态及每个读写头读到的带符号，图灵机的一个计算步可实现下面3个操作之一或全部。

- (1)改变有限状态控制器中的状态。
- (2)清除当前读写头下的方格中原有带符号并写上新的带符号。
- (3)独立地将任何一个或所有读写头，向左移动一个方格(L)或向右移动一个方格(R)或停在当前单元不动(S)。

k 带图灵机可形式化地描述为一个7元组 $(Q, T, I, \delta, b, q_0, q_f)$ ，其中：

- (1) Q 是有限个状态的集合。 (2) T 是有限个带符号的集合。
- (3) I 是输入符号的集合， $I \subseteq T$ 。 (4) b 是惟一的空白符， $b \in T - I$ 。
- (5) q_0 是初始状态。 (6) q_f 是终止(或接受)状态。
- (7) δ 是移动函数。 它是从 $Q \times T^k$ 的某一子集映射到 $Q \times (T \times \{L, R, S\})^k$ 的函数。

1 图灵机

1.1 多带图灵机

图灵机既可作为语言接受器，也可作为计算函数的装置。

图灵机M的时间复杂性 $T(n)$ 是它处理所有长度为n的输入所需的最大计算步数。如果对某个长度为n的输入，图灵机不停机， $T(n)$ 对这个n值无定义。

图灵机的空间复杂性 $S(n)$ 是它处理所有长度为n的输入时，在k条带上所使用过的方格数的总和。如果某个读写头无限地向右移动而不停机， $S(n)$ 也无定义。

1 图灵机

1.2 确定性图灵机与非确定性图灵机

在图灵机计算模型中，移动函数 δ 是单值的，即对于 $Q \times T^k$ 中的每一个值，当它属于 δ 的定义域时， $Q \times (T \times \{L, R, S\})^k$ 中只有惟一的值与之对应，称这种图灵机为**确定性图灵机**，简记为**DTM**(Deterministic Turing Machine)。

非确定性图灵机 (NDTM)：一个 k 带的非确定性图灵机 M 是一个7元组： $(Q, T, I, \delta, b, q_0, q_f)$ 。与确定性图灵机不同的是非确定性图灵机允许移动函数 δ 具有不确定性，即对于 $Q \times T^k$ 中的每一个值 $(q; x_1, x_2, \dots, x_k)$ ，当它属于 δ 的定义域时， $Q \times (T \times \{L, R, S\})^k$ 中有惟一的一个子集 $\delta(q; x_1, x_2, \dots, x_k)$ 与之对应。可以在 $\delta(q; x_1, x_2, \dots, x_k)$ 中随意选定一个值作为它的函数值。

1 图灵机

1.2 非确定性图灵机

非确定型图灵机M在输入串上的计算过程可以表示为一棵树，不同的分支对应着每一步计算的不同的可能性。只要有任意一个分支进入接受状态，则称M接受；只要有任意一个分支进入拒绝状态，则称M拒绝；某些分支可能永远无法停机，但只要有一个分支可以进入接受或拒绝状态，我们就说M在输入上可停机。注意，我们规定 M必须是无矛盾的，即它不能有某个分支接受而同时另一个分支拒绝，这样有矛盾的非确定型图灵机是不合法的。

1 图灵机

1.2 非确定性图灵机

定理1：对于任意一个非确定型图灵机 M ，存在一个确定型图灵机 M' ，使得它们的语言相等，即 $L(M)=L(M')$ 。

证明：对于非确定型图灵机 M ，构造一个确定型图灵机 M' 如下：

1. 令 $k=1$ ；
2. 深度优先地模拟 M 的每个分支的计算，但每个分支最多只计算 k 步，如果某个计算分支在 k 步内可以停机，则 M' 也停机，并将该计算分支的计算结果输出。
3. 令 k 增加1，跳转到上一步继续执行。

显然，若 M 有某个分支可以停机，则此 M' 也一定会找到该分支并停机。因此 $L(M)=L(M')$ 。

定理2：对于一台时间复杂性为 $T(n)$ 的非确定性图灵机，可以用一台时间复杂性为 $O(C^{T(n)})$ 的确定型图灵机模拟，其中 C 为一常数。

2 问题变换与计算复杂性归约

通过问题变换的技巧，可以将2个不同问题的计算复杂性联系在一起。这样就可以将一个问题的计算复杂性归结为另一个问题的计算复杂性，从而实现问题的计算复杂性归约。

具体地说，假设有2个问题A和B，**将问题A变换为问题B**是指：

- (1) 将问题A的输入变换为问题B的适当输入。
- (2) 解出问题B。
- (3) 把问题B的输出变换为问题A的正确解。

若用 $O(\tau(n))$ 时间能完成上述变换的第(1)步和第(3)步，则称问题A是 $\tau(n)$ 时间可变换到问题B，且简记为 **$A \propto_{\tau(n)} B$** 。其中的n通常为问题A的规模(大小)。

当 $\tau(n)$ 为n的多项式时，称问题A可在多项式时间内变换为问题B。特别地，当 $\tau(n)$ 为n的线性函数时，称问题A可线性地变换为问题B。

2 问题变换与计算复杂性归约

问题的变换与问题的计算复杂性归约的关系：

命题1(计算时间下界归约)：若已知问题A的计算时间下界为 $T(n)$ ，且问题A是 $\tau(n)$ 可变换到问题B，即 $A \propto_{\tau(n)} B$ ，则 $T(n) - O(\tau(n))$ 为问题B的一个计算时间下界。

命题2(计算时间上界归约)：若已知问题B的计算时间上界为 $T(n)$ ，且问题A是 $\tau(n)$ 可变换到问题B，即 $A \propto_{\tau(n)} B$ ，则 $T(n) + O(\tau(n))$ 是问题A的一个计算时间上界。

在命题1和命题2中，当 $\tau(n) = o(T(n))$ 时，问题A的下界归约为问题B的下界，问题B的上界归约为问题A的上界。

3 P类与NP类语言

P类和NP类语言的定义：

$P = \{L | L\text{是一个能在多项式时间内被一台DTM所接受的语言}\}$

$NP = \{L | L\text{是一个能在多项式时间内被一台NDTM所接受的语言}\}$

由于一台确定性图灵机可看作是非确定性图灵机的特例，所以可在多项式时间内被确定性图灵机接受的语言也可在多项式时间内被非确定性图灵机接受。故 $P \subseteq NP$ 。

3 P类与NP类语言

NP类语言举例——无向图的团问题。

该问题的输入是一个有n个顶点的无向图 $G=(V, E)$ 和一个整数k。要求判定图G是否包含一个k顶点的**完全子图(团)**，即判定是否存在 $V' \subseteq V$, $|V'|=k$, 且对于所有的 $u, v \in V'$, 有 $(u, v) \in E$ 。

若用邻接矩阵表示图G, 用二进制串表示整数k, 则团问题的一个实例可以用长度为 $n^2 + \log k + 1$ 的二进位串表示。因此, 团问题可表示为**语言**:

CLIQUE= $\{w\#v | w, v \in \{0, 1\}^*,$ 以w为邻接矩阵的图G有一个k顶点的团, 其中v是k的二进制表示。}

3 P类与NP类语言

接受该语言CLIQUE的**非确定性算法**：用非确定性选择指令选出包含 k 个顶点的候选顶点子集 V ，然后确定性地检查该子集是否是团问题的一个解。算法分为3个阶段：

算法的第一阶段将输入串 $w\#v$ 分解，并计算出 $n=\sqrt{|w|}$ ，以及用 v 表示的整数 k 。若输入不具有形式 $w\#v$ 或 $|w|$ 不是一个平方数就拒绝该输入。显而易见，第一阶段可 $O(n^2)$ 在时间内完成。

在算法的第二阶段中，非确定性地选择 V 的一个 k 元子集 $V' \subseteq V$ 。

算法的第三阶段是确定性地检查 V' 的团性质。若 V' 是一个团则接受输入，否则拒绝输入。这显然可以在 $O(n^4)$ 时间内完成。因此，整个算法的时间复杂性为 $O(n^4)$ 。

非确定性算法在多项式时间内接受语言CLIQUE，故 $\text{CLIQUE} \in \text{NP}$ 。

4 多项式时间验证

多项式时间可验证语言类VP可定义为：

$VP = \{L | L \in \Sigma^*, \Sigma \text{ 为一有限字符集, 存在一个多项式 } p \text{ 和一个多项式时间验证算法 } A(X, Y) \text{ 使得对任意 } X \in \Sigma^*, X \in L \text{ 当且仅当存在 } Y \in \Sigma^*, |Y| \leq p(|X|) \text{ 且 } A(X, Y) = 1\}$ 。

定理3： $VP=NP$

$VP \subseteq NP$ ：对任意 $L \in VP$ ，设 p 是一个多项式且 A 是一个多项式时间验证算法，则下面的非确定性算法接受语言 L ：

- (1) 对于输入 X ，非确定性地产生一个字符串 $Y \in \Sigma^*$ 。
- (2) 当 $A(X, Y) = 1$ 时，接受 X 。

该算法的步骤(1)与图问题的第二阶段的非确定性选择算法一样，至多在 $O(|X|)$ 的时间内完成。步骤(2)的计算时间是 $|X|$ 和 $|Y|$ 的多项式，而 $|Y| \leq p(|X|)$ ，因此，它也是 $|X|$ 的多项式。整个算法可在多项式时间内完成。因此， $L \in NP$ 。由此可见， $VP \subseteq NP$ 。

4 多项式时间验证

定理3： $VP=NP$

NP \subseteq VP：设 $L \in NP$, $L \in \Sigma^*$, 且非确定性图灵机M在多项式时间p内接受语言L。设M在任何情况下只有不超过d个的下一动作选择，则对于输入串X，M的任一动作序列可用 $\{0,1,\dots,d-1\}$ 的长度不超过 $p(|X|)$ 的字符串来编码。不失一般性，设 $|\Sigma| \geq d$ 。验证算法 $A(X,Y)$ 用于验证“Y是M上关于输入X的一条接受计算路径的编码”。即当Y是这样一个编码时， $A(X,Y)=1$ 。 $A(X,Y)=1$ 显然可在多项式时间内确定性地进行验证，且 $L = \{X | \text{存在 } Y \text{ 使得 } |Y| \leq p(|X|) \text{ 且 } A(X,Y)=1\}$ 。因此， $L \in VP$ 。由此可知， $NP \subseteq VP$ 。综合两者，可得 $VP=NP$ 。

例如(哈密顿回路问题)：一个无向图G含有哈密顿回路吗？

无向图G的哈密顿回路是通过G的每个顶点恰好一次的简单回路。可用语言HAM-CYCLE 定义该问题如下：

$$\text{HAM-CYCLE} = \{G | G \text{ 含有哈密顿回路}\}$$

5 NP完全问题

设 $L_1 \subseteq \Sigma_1^*$, $L_2 \subseteq \Sigma_2^*$ 是2个语言。所谓语言 L_1 能在**多项式时间内变换**为语言 L_2 (简记为 $L_1 \leq_p L_2$) 是指存在映射 $f: \Sigma_1^* \rightarrow \Sigma_2^*$, 且 f 满足:

- (1) 有一个计算 f 的多项式时间确定性图灵机;
- (2) 对于所有 $x \in \Sigma_1^*$, $x \in L_1$, 当且仅当 $f(x) \in L_2$ 。

定义: 语言 L 是**NP完全**的当且仅当

- (1) $L \in NP$;
- (2) 对于所有 $L' \in NP$ 有 $L' \leq_p L$ 。

如果有一个语言 L 满足上述性质(2), 但不一定满足性质(1), 则称该语言是**NP难 (NP-hard)** 的。所有NP完全语言构成的语言类称为**NP完全语言类**, 记为**NPC**。

5 NP完全问题

定理4：设 L 是NP完全的，则

- (1) $L \in P$ 当且仅当 $P = NP$ ；
- (2) 若 $L \propto_p L_1$, 且 $L_1 \in NP$, 则 L_1 是NP完全的。

证明: (1) 若 $P = NP$, 则显然 $L \in P$ 。反之, 设 $L \in P$, 而 $L_1 \in NP$ 。则 L 可在多项式时间 p_1 内被确定图灵机 M 所接受。又由 L 的NP完全性知 $L_1 \propto_p L$, 即存在映射 f , 使 $L = f(L_1)$ 。设 N 是在多项式时间 p_2 内计算 f 的确定图灵机。用图灵机 M 和 N 构造识别语言 L_1 的算法 A 如下:

- i) 对于输入 x , 用 N 在 $p_2(|x|)$ 时间内计算出 $f(x)$ 。
 - ii) 在时间 $|f(x)|$ 内将读写头移到 $f(x)$ 的第一个符号处。
 - iii) 用 M 在时间 $p_1(|x|)$ 内判定 $f(x) \in L$ 。若 $f(x) \in L$, 则接受 x , 否则拒绝 x 。
- 上述算法显然可以接受语言 L_1 , 其计算时间为 $p_2(|x|) + |f(x)| + p_1(|x|)$ 。由于图灵机一次只能在一个方格中写入一个符号, 故 $|f(x)| \leq |x| + p_2(|x|)$ 。因此, 存在多项式 r 使得 $p_2(|x|) + |f(x)| + p_1(|x|) \leq r(x)$ 。因此, $L_1 \in P$ 。由 L_1 的任意性, 知 $P = NP$

5 NP完全问题

定理4：设 L 是NP完全的，则

- (1) $L \in P$ 当且仅当 $P = NP$;
- (2) 若 $L \propto_p L_1$, 且 $L_1 \in NP$, 则 L_1 是NP完全的。

证明: (2) 只要证明对任意的 $L' \in NP$, 有 $L' \propto_p L_1$ 。由于 L 是 NP 完全的, 故存在多项式时间变换 f 使 $L = f(L')$ 。又由于 $L \propto_p L_1$, 故存在一多项式时间变换 g 使 $L_1 = g(L)$ 。因此, 若取 f 和 g 的复合函数 $h = g(f)$, 则 $L_1 = h(L')$ 。易知 h 为一多项式。因此, $L' \propto_p L_1$ 。由 L' 的任意性, 即知 $L_1 \in NPC$ 。

5 NP完全问题

定理4：设 L 是NP完全的，则

(1) $L \in P$ 当且仅当 $P = NP$ ；

(2) 若 $L \propto_p L_1$, 且 $L_1 \in NP$, 则 L_1 是NP完全的。

- ✓ 由(1)可知：如果任一NPC问题可在多项式时间内求解，则所有NP中的问题都可在多项式时间内求解。反之，若 $P \neq NP$ ，则所有NPC问题都不可能在多项式时间内求解。
- ✓ (2)给出了证明一个问题是否是NPC或NP-hard问题的方法

5 NP完全问题

✓ NPC和NP-hard关系

- NP-hard问题至少跟NPC问题一样难。
 - NPC问题是NP-hard的，但反之不一定
- 例：停机问题是NP-hard而非NPC的。
- ∴该问题不可判定，即无任何算法(无论何复杂度)
求解该问题
- ∴该问题 \notin NP。但是
可满足问题SAT \propto_p 停机问题

5 NP完全问题

✓ **NP=? P**

∴确定型图灵机是非确定型图灵机的特例，∴**P⊆NP**

是否有**NP⊆P**? 即是否**NP=P**?

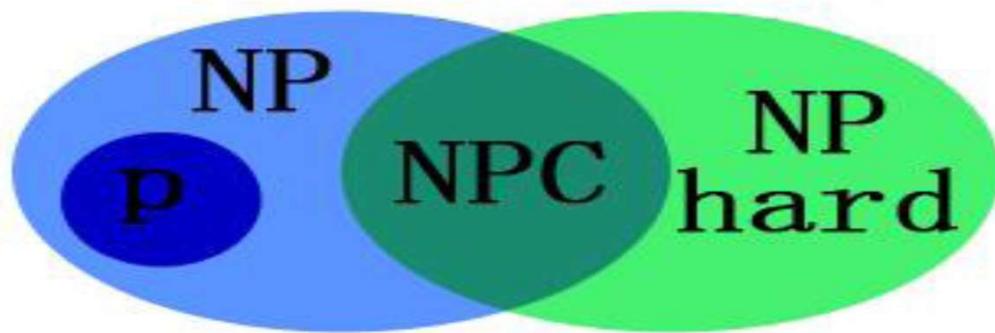
美国麻省的**Clay**数学研究所于**2000年5月24日**在巴黎法兰西学院宣布：对七个“**千年数学难题**”中的每一个均悬赏**100**万美元，而问题**NP=? P**位列其首：

- 1.P问题对NP问题 2.霍奇猜想
- 3.庞加莱猜想(2002.11-2003.7，俄罗斯数学家佩雷尔曼在3篇论文预印本中证明了几何化猜想，2006被授予菲尔兹奖)
- 4.黎曼假设 5.杨—米尔斯存在性和质量缺口
- 6.纳维叶—斯托克斯方程的存在性与光滑性
- 7.贝赫和斯维讷通—戴尔猜想

5 NP完全问题

✓ P、NP、NPC和NP-hard之关系

NPC是NP中最难的问题，但是NP-hard至少与NPC一样难



✓ 如何证明问题q是NP-hard或是NPC的？

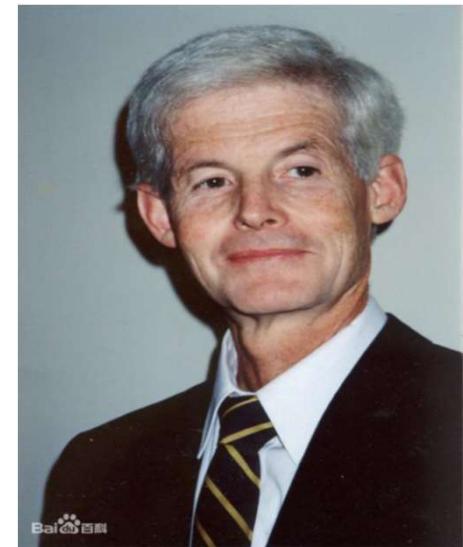
要证 q 是NP-hard的，只要找到1个已知的NPC或NPH问题 p ，然后将 p 多项式归约到 q 即可。若还能验证 $q \in NP$ ，则 q 是NPC的。

6 Cook定理

定理5 (Cook定理): 布尔表达式的可满足性问题 **SAT** 是 **NP** 完全的，即 **SAT** \in **NPC**。

- ✓ **Cook** 的贡献：第一个**NPC** 问题

史提芬·库克(Stephen Arthur Cook, 1939—) **NP** 完全性理论的奠基人，他在 1971 年论文 “**The Complexity of Theorem Proving Procedures**” 中给出了第一个 **NP** 完备的证明，即 **Cook** 定理，且证明了：**SAT** \in **P** 当且仅当 **P** = **NP**。



- ✓ 1961 年获得美国密西根大学理学学士学位；1962 年获得哈佛大学理学硕士学位；1966 年获得哈佛大学博士；1966 年至 1970 年担任加州大学伯克利分校助理教授；1970 年至 1975 年担任多伦多大学副教授；1975 年晋升为多伦多大学教授；1982 年获得图灵奖。₂₅

6 Cook定理

布尔表达式的可满足性问题(SAT)

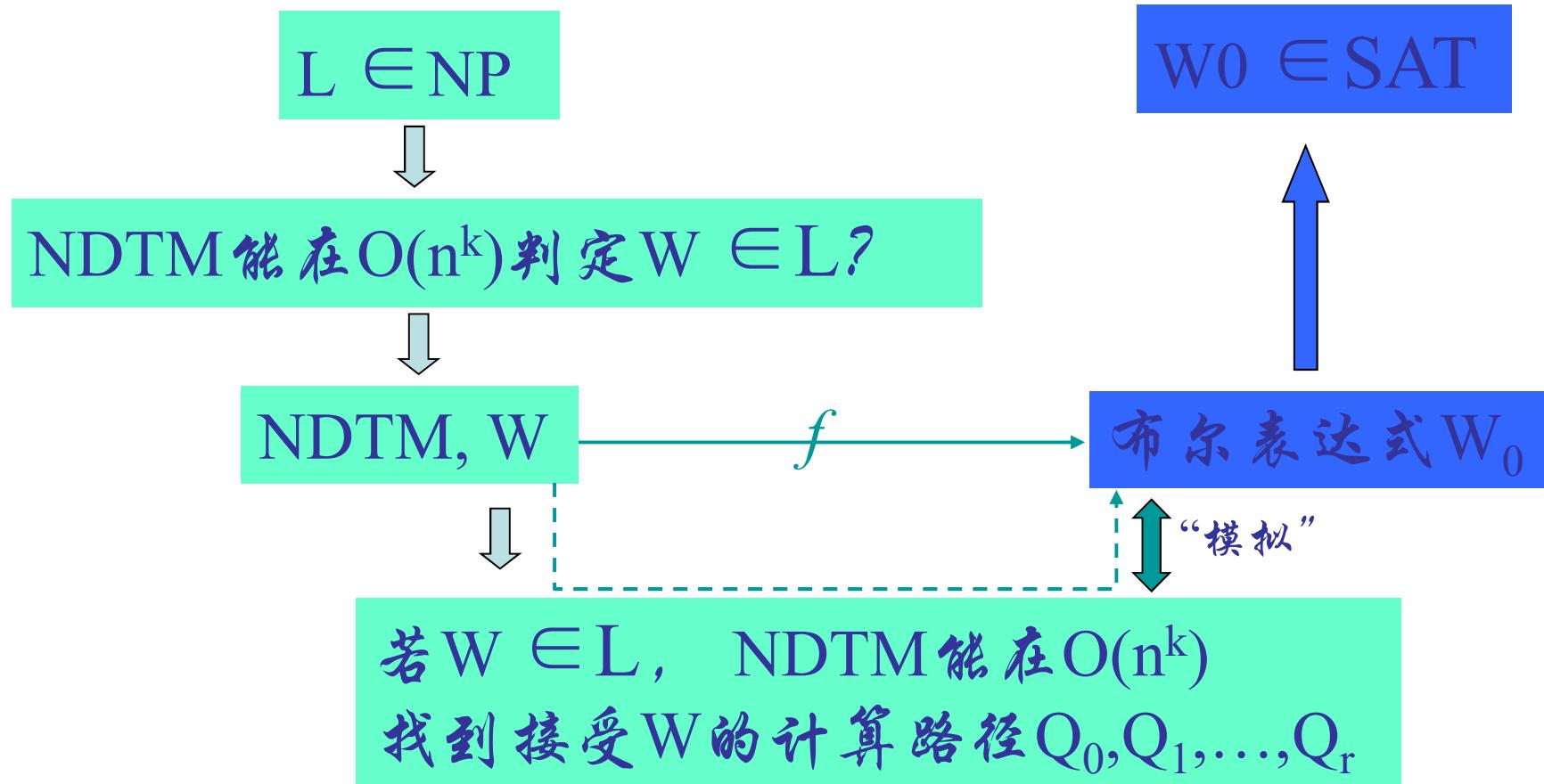
- ✓ SAT问题是给定 n 个布尔变量 x_1, x_2, \dots, x_n 的 m 个布尔表达式 A_1, A_2, \dots, A_m , 确定是否存在某种对各布尔变量的0,1赋值, 使得布尔表达式 A_1, A_2, \dots, A_m 为1 (为真) ?
- ✓ 例

$$A = (x_1 \wedge (x_2 \vee \bar{x}_3)) \wedge (\bar{x}_1 \vee (\bar{x}_2 \wedge \bar{x}_3))$$

$$A = (x_1 \vee (x_2 \wedge x_3)) \wedge (x_1 \vee (\bar{x}_2 \wedge \bar{x}_3)) \wedge (x_2 \vee x_3) \wedge (\bar{x}_2 \vee \bar{x}_3)$$

6 Cook定理

□ 证明思路



6 Cook定理

□ 具体方法

- ✓ $L \in NP$, 设 M 是一台能在多项式时间内识别 L 的非确定性图灵机, 而 W 是对 M 的一个输入
- ✓ 由 M 和 W 能在多项式时间内构造一个布尔表达式 W_0
- ✓ 模拟由 M 接受 W 的所有瞬象序列
- ✓ 使得 W_0 是可满足的当且仅当 M 接受 W

6 Cook定理

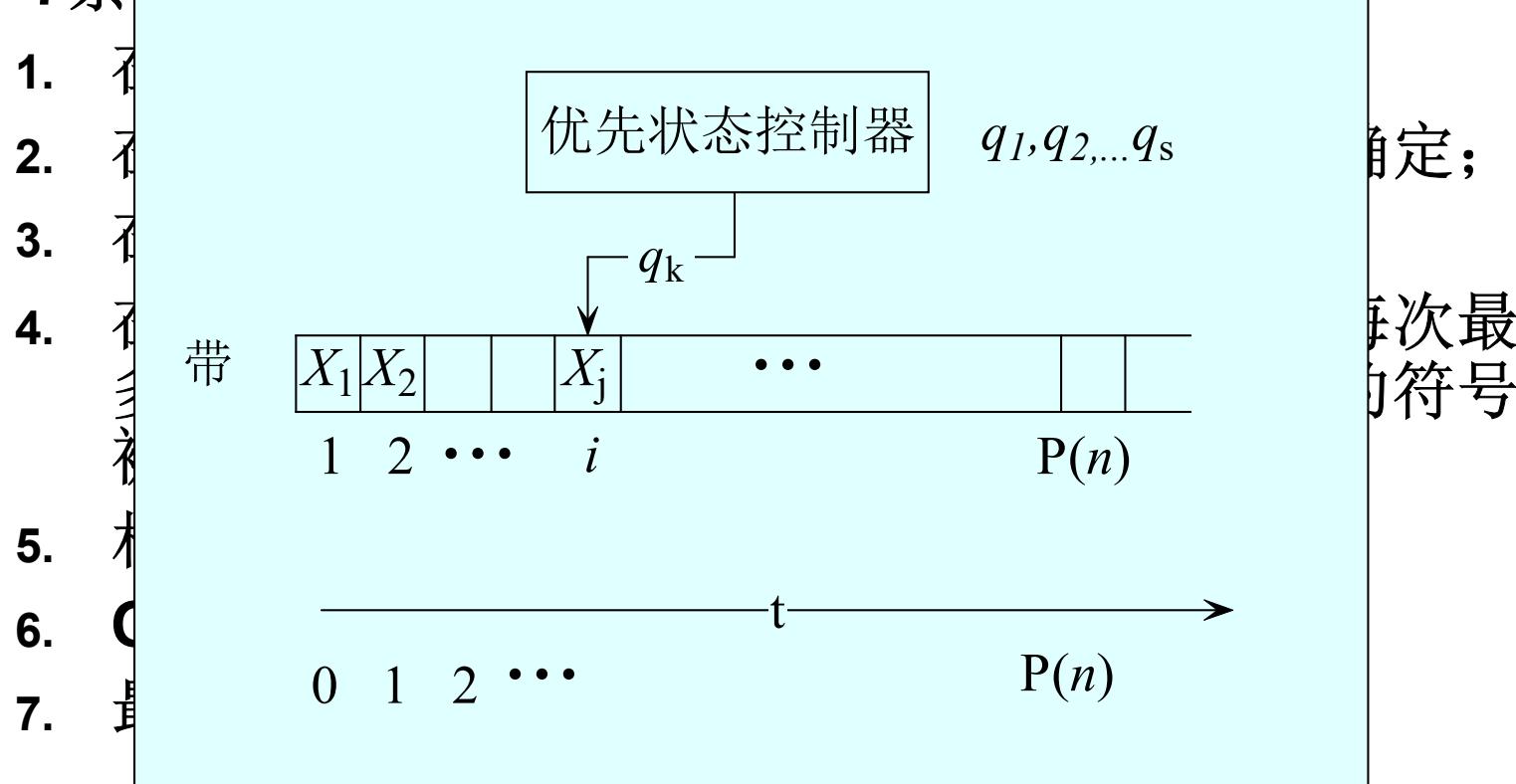
□ 已知与假设

- 不难证明， $L \in NP$ 能由一台单带的NDTM在多项式时间内识别。
- 设 M 为一台单带NDTM
 - s 个状态 q_0, q_1, \dots, q_{s-1}
 - m 个带符号 X_0, X_1, \dots, X_m
 - 多项式 $P(n)$ 是 M 的识别 L 时间复杂性。
- 设 W 是 M 的一个长度为 n 的输入，若 M 接受 W ，只需不超过 $P(n)$ 次移动。
- 对于 W ，存在 M 的一个瞬像序列 Q_0, Q_1, \dots, Q_r ，其中 Q_0 为初始瞬像， Q_r 为接受瞬像。可以令 $r = P(n)$ ，若不足可用空动作补齐

6 Cook定理

□ 与瞬像序列 Q_0, Q_1, \dots, Q_r 等价的断言

- 7条断言



6 Cook定理

□ 引进命题

- $C < i, j, t > = 1$ 当且仅当在时刻 t , M 输入带的第 i 个方格中的带符号为 X_j ; 其中, $1 \leq i \leq P(n)$, $1 \leq j \leq m$, $0 \leq t \leq P(n)$ 。
- $S < k, t > = 1$ 当且仅当在时刻 t , M 的状态为 q_k ; 其中 $1 \leq k \leq m$, $0 \leq t \leq P(n)$,
- $H < i, t > = 1$ 当且仅当在时刻 t , 读写头扫描第 i 个方格; 其中, $1 \leq i \leq P(n)$, $0 \leq t \leq P(n)$ 。
- 总共最多有 $O(P^2(n))$ 个命题。 // $O(P^2(n) \log n)$

6 Cook定理

□ 引进谓词

- $U(x_1, x_2, \dots, x_r) = 1$ 当且仅当各变量 x_1, x_2, \dots, x_r 中只有一个变量取值 1 时。

$$U(x_1, x_2, \dots, x_r) = (x_1 + x_2 + \dots + x_r) \prod_{i \neq j} (\bar{x}_i + \bar{x}_j)$$

- $U(x_1, x_2, \dots, x_r)$ 的长度是 $O(r^2)$ 。 // $O(r^2 \log n)$

6 Cook定理

□ 构造与断言1对应的布尔表达式A

- 在每一瞬像中读写头恰只扫描一个方格。设 A_t 表示在时刻t时M的读写头恰好扫描一个方格

$$A_t = U(H < 1, t >, H < 2, t >, \dots, H < P(n), t >)$$

$$0 \leq t \leq P(n)$$

$$A = A_0 A_1 \cdots A_{P(n)}$$

$$O(P^3(n))$$

6 Cook定理

□ 构造与断言2对应的布尔表达式B

- 在每一瞬像中，每个方格中的带符号是唯一确定的
- 设 B_{it} 表示在时刻t，第i个方格中只含有一个带符号

$$B_{it} = U(C < i, 1, t >, C < i, 2, t >, \dots, C < i, m, t >)$$

$$0 \leq t \leq P(n)$$

$$B = \prod_{0 \leq i, t \leq P(n)} B_{it}$$

$$O(P^2(n)) \quad // B_{it} \text{长度与n无关}$$

6 Cook定理

□ 构造与断言3对应的布尔表达式C

- 在每一瞬像中恰有一个状态

$$C = \prod_{0 \leq t \leq P(n)} U(S < 0, t >, S < 1, t >, \dots, S < s-1, t >)$$

$O(P(n))$ //U长度与n无关

6 Cook定理

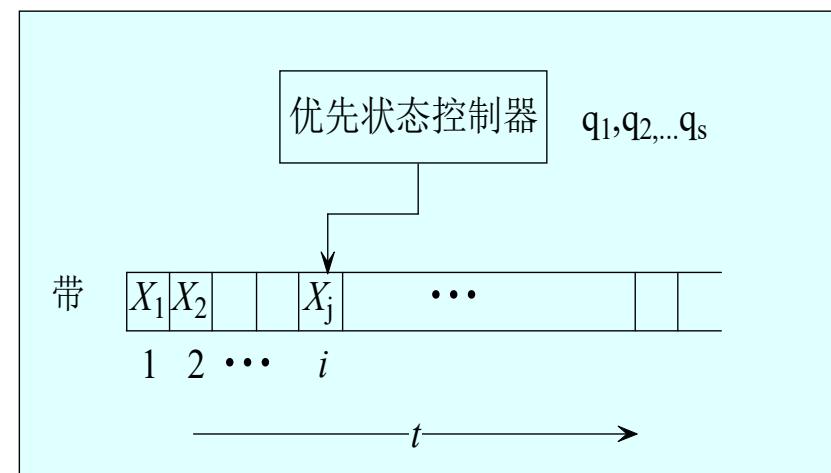
□ 构造与断言4对应的布尔表达式D

- 在该计算路径中，从一个瞬像到下一个瞬像每次最多有一个方格的符号被修改

$$D = \prod_{i,j,t} (C < i, j, t > \equiv C < i, j, t+1 > + H < i, t >)$$

// $H < i, t >$ 仅在一个方格*i*处取值1，此时 $C < i, j, t > \neq C < i, j, t+1 >$

$$O(P^2(n))$$



6 Cook定理

□ 构造与断言5对应的布尔表达式E

- 相继的瞬像是根据移动函数来改变状态的

$$E_{ijkt} = \neg C < i, j, t > + \neg H < i, t > + \neg S < k, t >$$

$$+ \sum_l (C < i, j_l, t+1 > S < k_l, t+1 > H < i_l, t+1 >)$$

//l遍取当M处于状态 q_k 且扫描 X_j 时所有可能的移动，即取遍使得 $(q_{kl}, X_{jl}, d_{il}) \in \delta(q_k, X_j)$ 的一切值

$$E = \prod_{i,j,k,t} E_{ijkt} \quad O(P^2(n)) \quad // E_{ijkt} \text{长度与n无关}$$

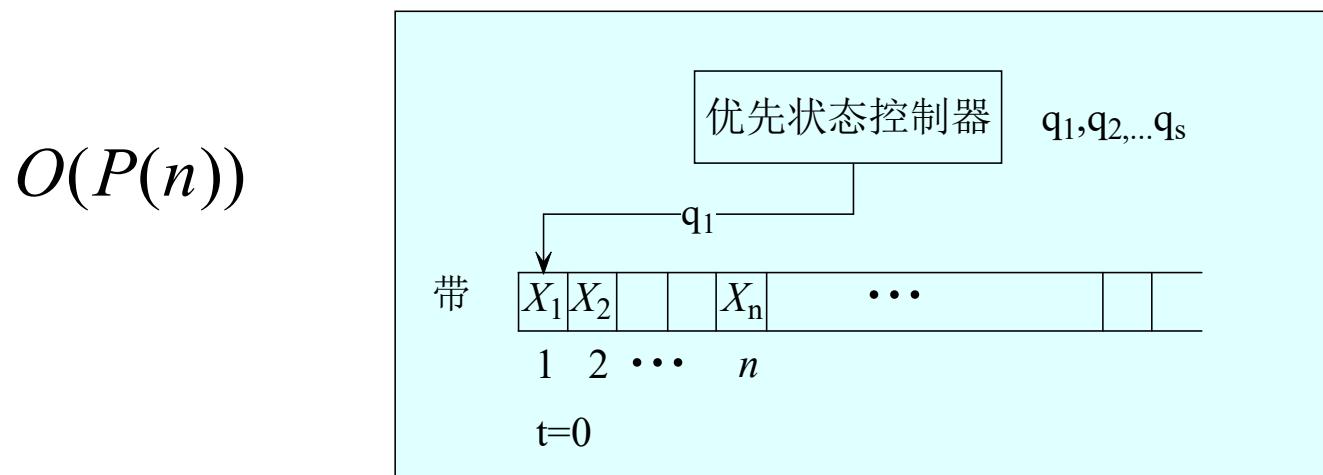
6 Cook定理

□ 构造与断言6对应的布尔表达式F

- Q_0 是M在输入W时的初始瞬像

$$F = S <1,0> H <1,0> \prod_{1 \leq i \leq n} C <i,j_i,0> \prod_{n \leq i \leq P(n)} C <i,1,0>$$

// t=0时， M处于初始 q_1 状态， 读写头在最左边， 带上前n个方格为W的符号串， 其余方格都是空白符



6 Cook定理

□ 构造与断言7对应的布尔表达式G

- 最后一个瞬像中的状态是接受状态

$$G = S < s - 1, P(n) >$$

// q_{s-1} 为接受状态

6 Cook定理

□ 构造 W_0

$$W_0 = ABCDEFG$$

- 给定可接受的瞬像序列 Q_0, Q_1, \dots, Q_r ， 显然可找到变量 $C < i, j, t >$ ， $S < k, t >$ 和 $H < i, t >$ 的某个0,1赋值， 使 W_0 取值为1。
- 若有一个使 W_0 满足的赋值，则可根据其变量的赋值相应地找到可接受计算路径 Q_0, Q_1, \dots, Q_r 。
- W_0 是可满足的当且仅当 M 接受 W 。
- 因为每一个因子最多需要个符号 $O(P^3(n))$ ， 而每个符号的长度至多为 $O(\log n)$ ， 因此的长度不超过 $O(P^3(n)\log n)$ 。

7典型的NP完全问题

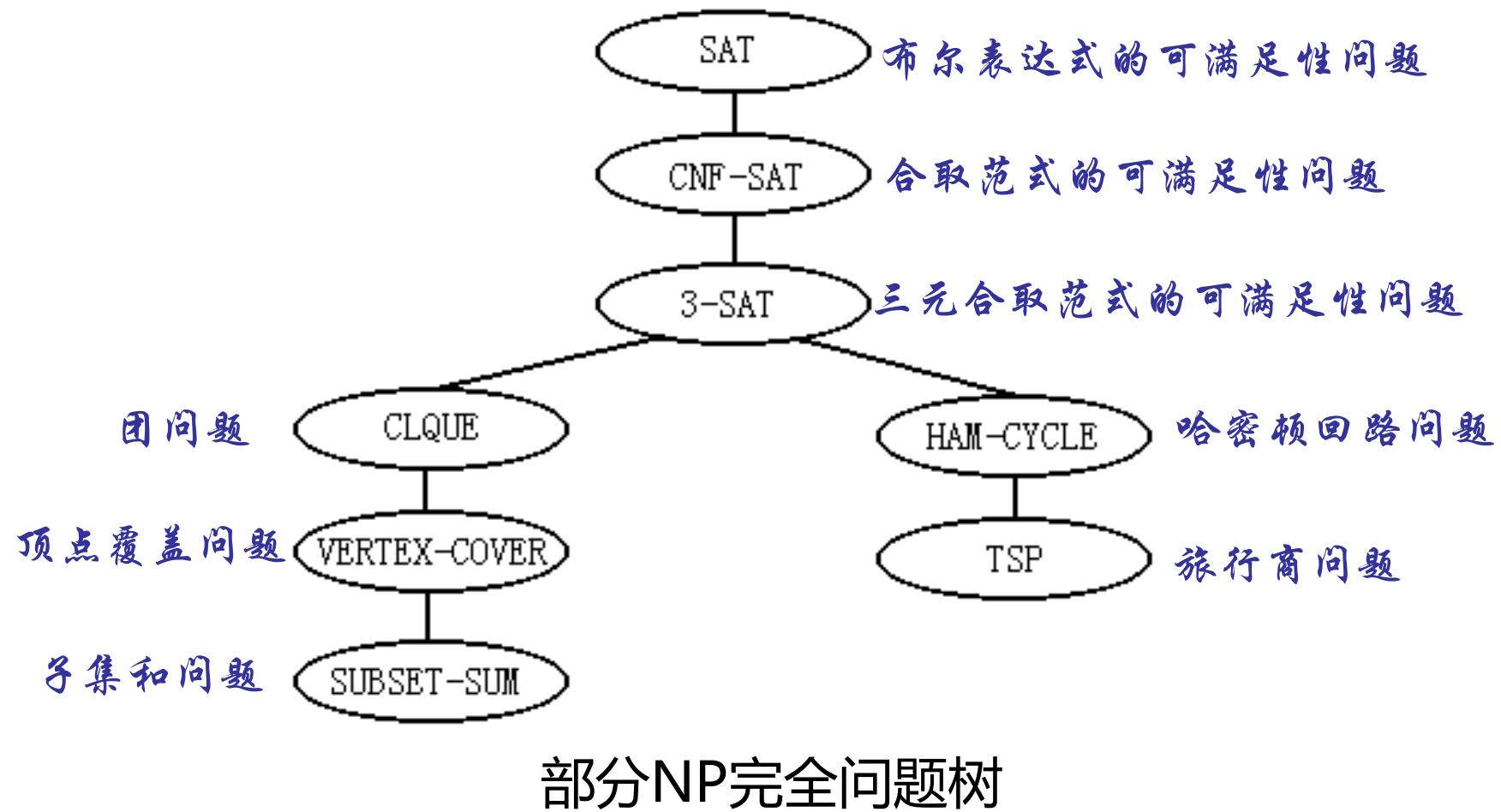
✓ Karp的贡献

理查德·卡普（Richard Karp，1935-）1972年论文“**Reducibility among Combinatorial Problems**”发展和加强了由库克提出的“NP完全性”理论。尤其是库克仅证明了命题演算的可满足问题是NP完全的，而卡普则证明了从组合优化中引出的大多数经典问题(背包问题、覆盖问题、匹配问题、分区问题、路径问题、调度问题等)都是NP完全问题。只要证明其中任一个问题属于P类的，就可解决计算复杂性理论中最大的一个难题，即P=?NP。



Karp于1955、1956和1959年分别获哈佛大学文学学士、理学硕士和应用数学博士学位，现任UCBerkeley计算机科学讲座教授，美国科学院、美国工程院、美国艺术与科学院、欧洲科学院院士。因其在计算机科学领域的基础贡献曾获图灵奖(1985)、冯诺依曼奖、美国国家科学勋章、哈佛大学百年奖章等奖项。

7 典型的NP完全问题



合取范式的可满足性问题 (CNF-SAT)

问题描述：给定一个合取范式 α ，判定它是否可满足。

如果一个布尔表达式是一些因子和之积，则称之为合取范式，简称CNF(Conjunctive Normal Form)。这里的因子是变量 x 或 \bar{x} 。例如： $(x_1 + x_2)(x_2 + x_3)(\bar{x}_1 + \bar{x}_2 + x_3)$ 就是一个合取范式，而 $x_1x_2 + x_3$ 就不是合取范式。

要证明CNF-SAT \in NPC，只要证明在Cook定理中定义的布尔表达式A, ..., G或者已是合取范式，或者有的虽然不是合取范式，但可以用布尔代数中的变换方法将它们化成合取范式，而且合取范式的长度与原表达式的长度只差一个常数因子。

3元合取范式的可满足性问题 (3-SAT)

问题描述：给定一个3元合取范式 α ，判定它是否可满足。

证明思路：

1、3元合取范式是合取范式的特殊情况，所以，也属于NP问题。

2、SAT可用多项式时间归约为3-SAT，即 $SAT \underset{p}{\propto} 3-SAT$ ：

✓ 给定SAT的一个实例 $f=c_1 \wedge c_2 \wedge \dots \wedge c_m$ ，它有m个析取子句

c_i ， $1 \leq i \leq m$ 和n个命题变元 x_j ， $1 \leq j \leq n$ 。

✓ 把f的每个析取子句 c_i ，变换为等价的子句集合，使每个子句都由三个文字组成。

✓ 则等价的子句集合是一个新的合取范式F，并且 $F \Leftrightarrow f$ 。 44

3元合取范式的可满足性问题 (3-SAT)

问题描述：给定一个3元合取范式 α ，判定它是否可满足。

证明思路：

考虑四种情况：

- (1) c_i 刚好有三个文字： 则 c_i 不变；
- (2) c_i 只有两个文字： $c_i=x_k \vee x_l$, $1 \leq k, l \leq n$, $k \neq l$ 。

对 c_i 作如下的恒等变换：

$$x_k \vee x_l \Leftrightarrow x_k \vee x_l \vee (x_s \wedge \neg x_s) \Leftrightarrow (x_k \vee x_l \vee x_s) \wedge (x_k \vee x_l \vee \neg x_s)$$

- (3) c_i 只有一个文字： $c_i=x_i$ 时， 同理可得：

$$x_i \Leftrightarrow (x_i \vee x_k \vee x_l) \wedge (x_i \vee x_k \vee \neg x_l) \wedge (x_i \vee \neg x_k \vee x_l) \wedge (x_i \vee \neg x_k \vee \neg x_l) \quad 45$$

3元合取范式的可满足性问题 (3-SAT)

证明思路：

(4) c_i 有三个以上文字：

① 假定 $c_i = x_1 \vee x_2 \dots \vee x_k$, $3 < k \leq n$, 可把 c_i 转换为由 $k-2$ 个子句组成的三元合取范式：

$$C_i \Leftrightarrow (x_1 \vee x_2 \vee y_1) \wedge (x_3 \vee \neg y_1 \vee y_2) \wedge \dots \wedge (x_{k-2} \vee \neg y_{k-4} \vee y_{k-3}) \wedge (x_{k-1} \vee x_k \vee \neg y_{k-3})$$

其中， y_1, \dots, y_{k-3} 是新增加的命题变元，其值可给定。

② c_i 可满足，当且仅当 C_i 可满足：若 c_i 可满足，即可使 $c_i = x_1 \vee x_2 \dots \vee x_k$ 为真，在 c_i 中，至少有一个文字 x_l 取值为真。对所有的 s , $1 \leq s \leq l-2$, 令 y_s 为真；对所有的 t , $l-2 < t \leq k-3$, 令 y_k 为假。则 C_i 为真。

团问题CLIQUE

问题描述：给定一个无向图 $G=(V, E)$ 和一个正整数 m ，判定图 G 是否包含一个 m 团，即是否存在， $V' \subseteq V$, $|V'|=m$, 且对任意 $u, w \in V'$ 有 $(u, w) \in E$ 。

证明思路：

已经知道 $\text{CLIQUE} \in \text{NP}$ 。通过 $\text{SAT} \xrightarrow{p} \text{CLIQUE}$ 来证明 CLIQUE 是NP难的，从而证明团问题是NP完全的。

团问题CLIQUE

证明SAT \leq_p CLIQUE

令 $f=c_1 \wedge c_2 \wedge \dots \wedge c_m$ 是SAT的一个实例，它具有m个析取子句 c_i ， $1 \leq i \leq m$ ，和n个布尔变量 x_1, x_2, \dots, x_n 。

1) 用多项式时间把f变换为图 $G=(V, E)$

构造 $G=(V, E)$ ，使得 V 中的一个顶点，对应于 f 中一个出现的文字，边集 E 由下面的关系给出：

$$E = \{(x_i, x_j) | x_i \text{ 和 } x_j \text{ 不在同一个子句中，且 } x_i \neq \neg x_j\}$$

例：SAT的实例是：

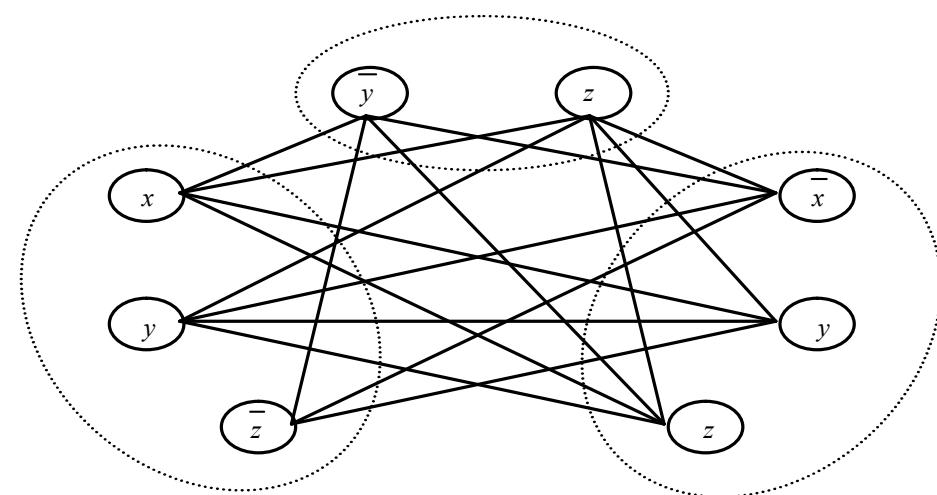
$$f = (x \vee y \vee \neg z) \wedge (\neg y \vee z) \wedge (\neg x \vee y \vee z)$$

显然，可用多项式时间为SAT的实例 f 构造所对应的图。

团问题CLIQUE

2) f可满足的，当且仅当G中有一个大小为m的集团。

必要性：若f可满足的，f的m个析取子句的真值均为真，每一个析取子句 c_i 中，至少有一个文字 x_i 取值为真。从每一个析取子句中，取一个真值为真的文字，共可取出m个真值为真的文字 x_1, x_2, \dots, x_m ，并且满足 $x_i \neq \neg x_j$ ，它们对应于图G中m个顶点。根据图G的构造，对 $1 \leq i, j \leq m$, $i \neq j$, 边 $(x_i, x_j) \in E$ 。G中这m个顶点，构成了一个大小为m的完全子图，它即为G中的一个大小为m的团。



团问题CLIQUE

2) **f可满足的，当且仅当G中有一个大小为m的团。**

充分性：

若G中存在一个大小为m的团，则必有一个大小为m的完全子图
假设这个子图的m个顶点，对应于文字 x_1, x_2, \dots, x_m ，
对 $1 \leq i, j \leq m, i \neq j$ ，有边 $(x_i, x_j) \in E$ 。

根据图G的构造， x_i 与 x_j 不同属一个子句，并且满足 $x_i \neq \neg x_j$ 。

则 x_1, x_2, \dots, x_m 分属于f的m个子句中，

并且，不会同时出现同一布尔变元的正负文字。

因此，只要使 x_1, x_2, \dots, x_m 分别取真值为真，则f的真值为真。

因此，f是可满足的。

综上所述， $SAT \leq_p CLIQUE$ 。

所以，团问题CLIQUE是NP完全的。

顶点覆盖问题 (VERTEX-COVER)

问题描述：给定一个无向图 $G=(V, E)$ 和一个正整数 k , 判定是否存在 $V' \subseteq V$, $|V'|=k$, 使得对于任意 $(u, v) \in E$ 有 $u \in V'$ 或 $v \in V'$ 。如果存在这样的 V' , 就称 V' 为图 G 的一个大小为 k 顶点覆盖。

证明思路：

首先, VERTEX-COVER $\in NP$ 。因为对于给定的图 G 和正整数 k 以及一个“证书” V' , 验证 $|V'|=k$, 然后对每条边 $(u, v) \in E$, 检查是否有 $u \in V'$ 或 $v \in V'$, 显然可在多项式时间内完成。

其次, 通过 $CLIQUE \in_p VERTEX-COVER$ 来证明顶点覆盖问题是NP难的。

顶点覆盖问题

证明 $\text{CLIQUE} \leq_p \text{VERTEX COVER}$

1) 无向图的补图

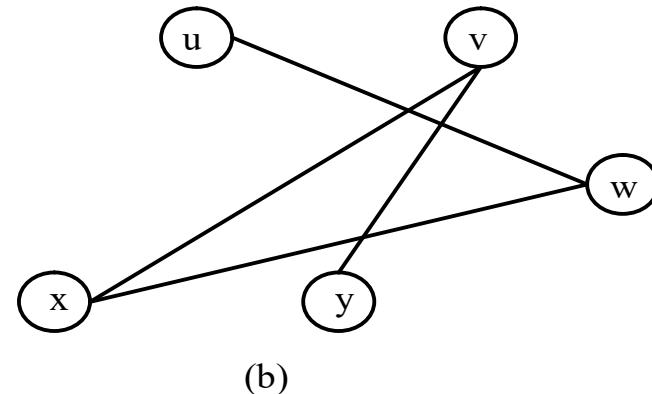
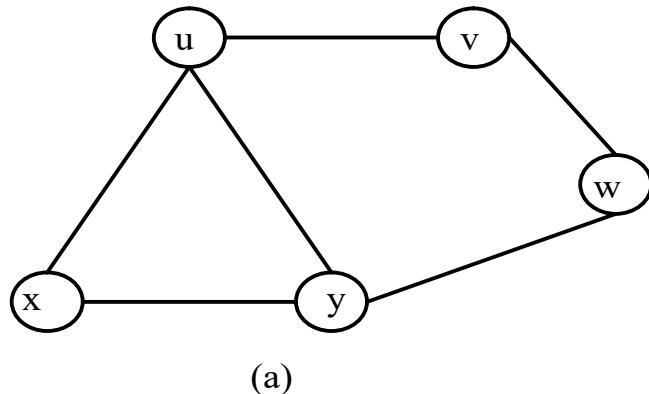
若无向图 $G=(V, E)$, 则 G 的补图 $\sim G=(V, \sim E)$, $\sim E=\{(u,v)|(u,v)\notin E\}$

例: 图 (b) 是图 (a) 的补图。

图 (a) 中, $|V|=5$, 有一个大小为3的团 $\{u,x,y\}$,

在 (b) 中, 则有一个大小为2的顶点覆盖 $\{v,w\}$ 。

2) 显然, 可在多项式时间里构造图 G 的补图 $\sim G$



顶点覆盖问题

3) $G=(V, E)$ 有一个大小为 $|V|-k$ 的团，当且仅当它的补图 $\sim G$ 有一个大小为 k 的顶点覆盖。

必要性： G 中有一个大小为 $|V|-k$ 的团，则它具有一个大小为 $|V|-k$ 个顶点的完全子图，令这 $|V|-k$ 个顶点集合为 V' 。

令 (u, v) 是 $\sim E$ 中的任意一条边，则 $(u, v) \notin E$ 。

所以 (u, v) 中必有一个顶点不属于 V' ，即 (u, v) 中必有一个顶点属于 $V - V'$ 。

即边 (u, v) 被 $V - V'$ 覆盖。

因为 (u, v) 是 $\sim E$ 中的任意一条边，

因此， $\sim E$ 中的边都被 $V - V'$ 覆盖，

所以， $V - V'$ 是 $\sim G$ 的一个大小为 $|V - V'| = k$ 的顶点覆盖。

顶点覆盖问题

3) $G=(V, E)$ 有一个大小为 $|V|-k$ 的团，当且仅当它的补图 $\sim G$ 有一个大小为 k 的顶点覆盖。

充分性：

$\sim G$ 有一个大小为 k 的顶点覆盖，

令这个顶点覆盖为 V' , (u, v) 是 $\sim E$ 中的任意一条边，

则 u 和 v 中至少有一个顶点属于 V' 。

因此，对任意的顶点 u 及 v ，若 $u \in V - V'$ ，并且 $v \in V - V'$ ，

则必然有 $(u, v) \in E$ 。

即 $V - V'$ 是 G 中一个大小为 $|V| - k$ 的团。

综上所述， $\text{CLIQUE} \leq_p \text{VERTEX COVER}$ 。

所以，顶点覆盖问题 VERTEX COVER 是NP完全的。

子集和问题 (SUBSET-SUM)

问题描述：给定整数集合 S 和一个整数 t ，判定是否存在 S 的一个子集 $S' \subseteq S$ ，使得 S' 中整数的和为 t 。例如，若 $S=\{1, 4, 16, 64, 256, 1040, 1041, 1093, 1284, 1344\}$ 且 $t=3754$ ，则子集 $S'=\{1, 16, 64, 256, 1040, 1093, 1284\}$ 是一个解。

证明思路：

首先，对于子集和问题的一个实例 $\langle S, t \rangle$ ，给定一个“证书” S' ，要验证 $t = \sum_{i \in S'} i$ 是否成立，显然可在多项式时间内完成。因此， $\text{SUBSET-SUM} \in \text{NP}$ ；

其次，证明 $\text{VERTEX-COVER} \leq_p \text{SUBSET-SUM}$ 。

哈密顿回路问题 (HAM-CYCLE)

问题描述：给定无向图 $G=(V, E)$ ，判定其是否含有一哈密顿回路。

证明思路：

首先，已知哈密顿回路问题是一个NP类问题。

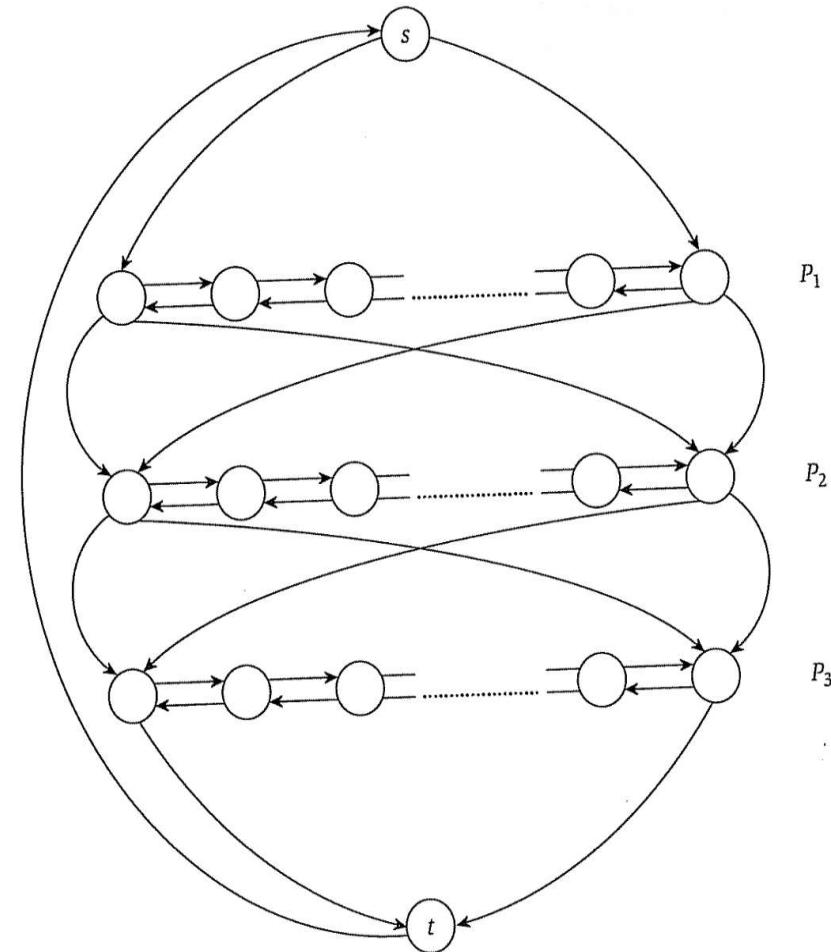
其次，通过证明 $3\text{-SAT} \leq_p \text{HAM-CYCLE}$ ，

得出： $\text{HAM-CYCLE} \in \text{NPC}$ 。

哈密顿回路问题

证明 $3\text{-SAT} \propto_p \text{HAM-CYCLE}$

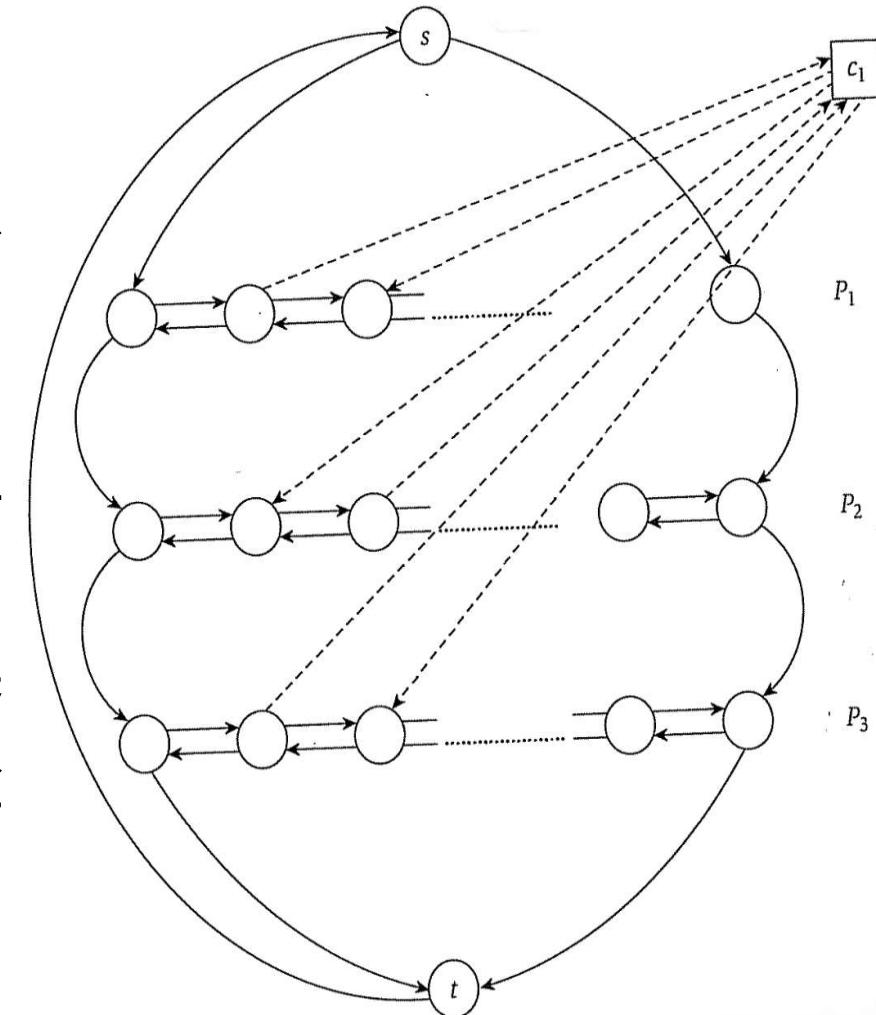
首先，构造一个基础有向图，包含 2^n 条互不相交的哈密顿回路，每条回路对应变量的一种可能的真值赋值。如右图，每层P对应3-SAT问题中的一个布尔变量，共n层节点，每层 $3(k+1)$ 个节点，其中k为3-SAT问题中子句的数量。再引入两个节点s和t。从s通过n层节点到t每一层都可以选择从左到右或从右到左，所以一共有 2^n 种选择可以不重复遍历所有节点。



哈密顿回路问题

证明 $3\text{-SAT} \propto_p \text{HAM-CYCLE}$

接着，引入 k 个不同的节点 c_1 到 c_k 。对任何一个节点 c ，都可以将其与图中任意3层连接起来，如右图所示。对于 c_j 连接 P_i 层的点必须是两个相邻的点 $v_{i,3j}$ 和 $v_{i,3j+1}$ 一共有两种不同的连接方式： $(v_{i,3j}, c_j)$, $(c_j, v_{i,3j+1})$ 和 $(v_{i,3j+1}, c_j)$, $(c_j, v_{i,3j})$ 。不管是任何一种连接方式，都可以保证在 P 层中从左到右或从右到左至少有一种方式可以遍历 P_i 中的所有节点及 c_j 。



哈密顿回路问题

我们可以把 c_j 看作3-SAT问题中的一个子句，其所连接的不同的三层中的点可以视作三个不同的布尔变量，其中**两种不同的连接方式则代表了是否对所选变量取反**。对每一层，从左到右或从右到左两种走法则可以视为变量取1或0.当子句 c_j 值为1， c_j 这个节点至少可以保证与三层中的其中一层连通（该层中的节点与 c_j 构成的子图中存在哈密顿回路）。那么，对于所有的子句 c 和节点 c 来说，我们可以把一个由 k 个子句组成的3-SAT问题转化为哈密顿回路问题。

证明3-SAT存在符合条件的赋值则图中存在哈密顿回路

假设存在满足3-SAT的情况，则可根据3-SAT问题中的子句数量和内容构造如上页所示的图。如果布尔变量 x_i 被赋值为1，那么我们就在 P_i 中从左到右遍历，反之则方向相反。对于每一个子句 c_j ，因为其结果为1，则一定与三层中的至少一层中的节点的连通方向与遍历方向相同，即存在哈密顿回路。

哈密顿回路问题

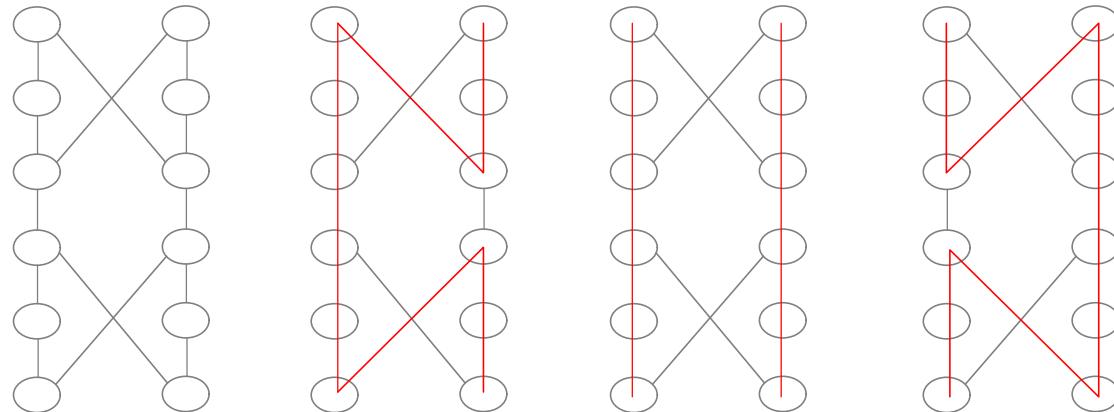
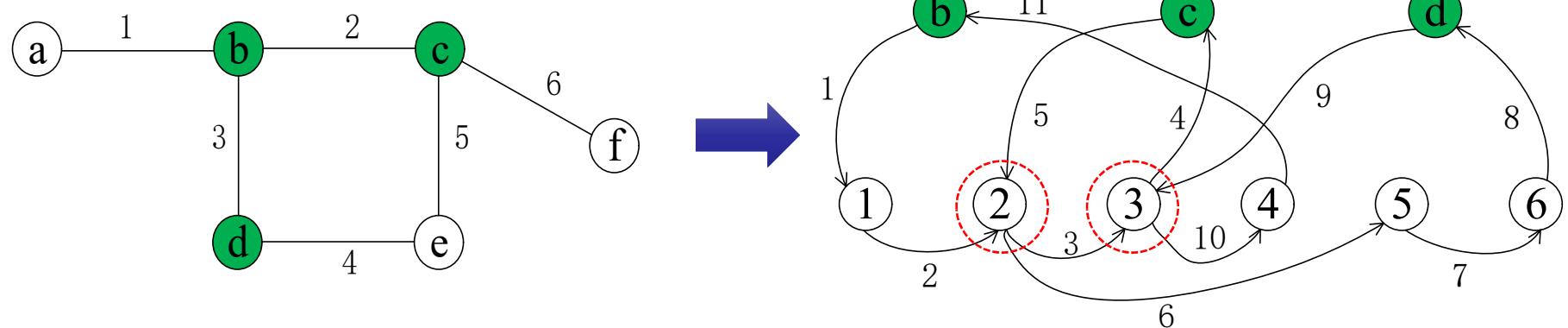
证明 图中存在哈密顿回路则3-SAT问题有解

首先，我们证明G中的哈密顿回路不能通过c来实现层间的跨越，因为如果有这样的跨越存在。我们通过 c_j 从 P_i 层中的 $v_{i,3j}$ 跨越到了 P_{i+x} 层，其中x为任意常数满足 $i+x$ 小于k，对于节点 $v_{i,3j+1}$ 我们将无法遍历，因为如果一旦遍历该节点(只能从 $v_{i,3j+2}$ 到 $v_{i,3j+1}$)，我们将不可避免再一次经过 $v_{i,3j}$ 或 $v_{i,3j+2}$ 。同理，对于从 $v_{i,3j+1}$ ，我们将永远无法遍历 $v_{i,3j}$ 。

既然我们已经证明了G中不存在通过c实现的层间跨越，那么我们跨越视作c连通了两个G中同一层中相邻的点，因此点c的存在不会影响每一层遍历的方向，即3-SAT问题中变量的取值0或1。那么，对于存在哈密顿回路的图G，每一层都会存在一种相应的遍历方向，对应3-SAT问题中的变量取值得其有解。

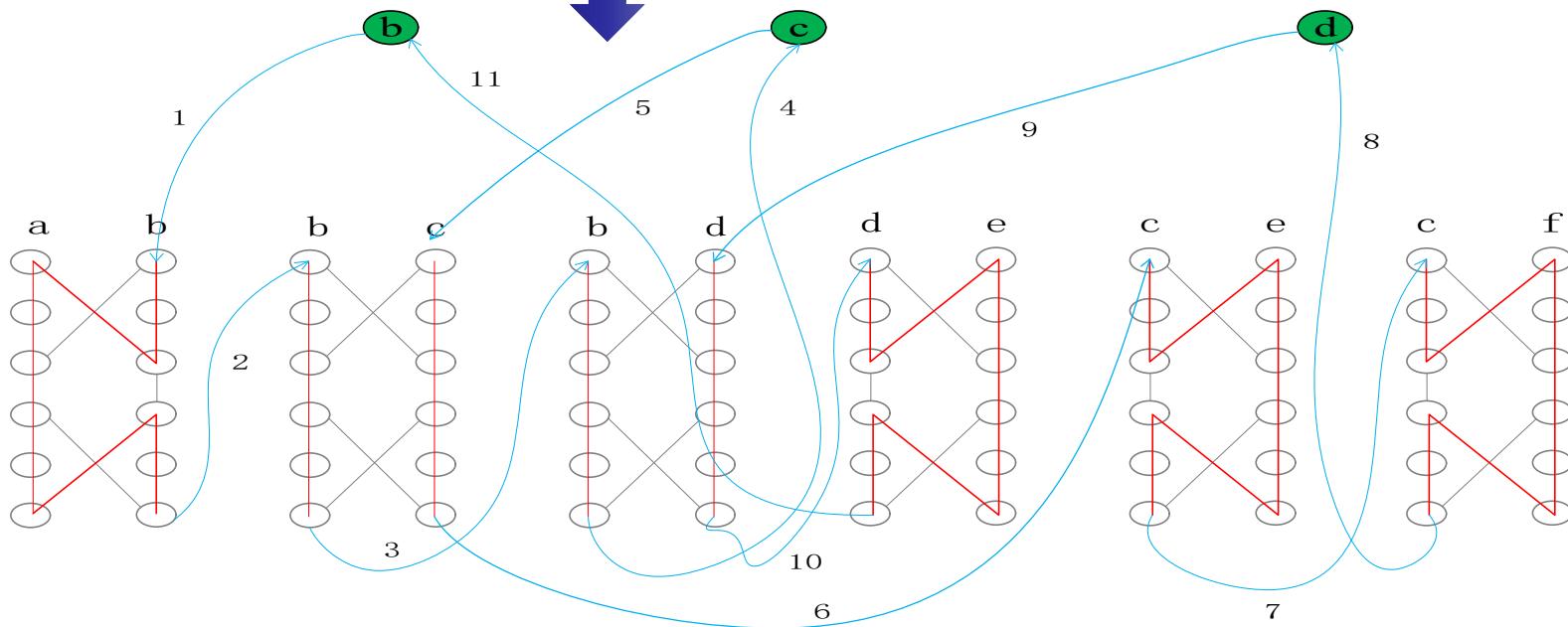
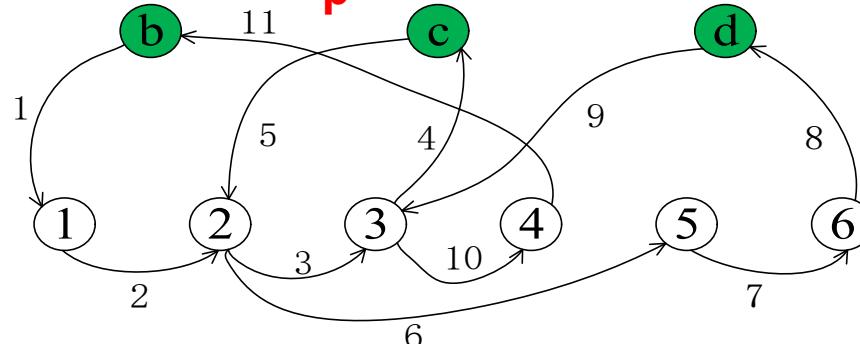
哈密顿回路问题

证明 $\text{VERTEX COVER} \propto_p \text{HAM-CYCLE}$



哈密顿回路问题

证明 $\text{VERTEX COVER} \propto_p \text{HAM-CYCLE}$



旅行售货员问题TSP

问题描述：给定一个无向完全图 $G=(V, E)$ 及定义在 $V \times V$ 上的一个费用函数 c 和一个整数 k ，判定 G 是否存在经过 V 中各顶点恰好一次的回路，使得该回路的费用不超过 k 。

首先，给定TSP的一个实例 (G, c, k) ，和一个由 n 个顶点组成的顶点序列。验证算法要验证这 n 个顶点组成的序列是图 G 的一条回路，且经过每个顶点一次。另外，将每条边的费用加起来，并验证所得的和不超过 k 。这个过程显然可在多项式时间内完成，即 $TSP \in NP$ 。

其次，旅行售货员问题与哈密顿回路问题有着密切的联系。哈密顿回路问题可在多项式时间内变换为旅行售货员问题。即 $HAM-CYCLE \underset{p}{\in} NPC$ 。从而，旅行售货员问题是NP难的。

因此， $TSP \in NPC$ 。

旅行售货员问题TSP

证明 HAM-CYCLE \in pTSP

哈密顿回路问题可在多项式时间内转换成旅行售货员问题。设图 $G=(V, E)$ 是哈密顿回路的一个实例，据此构造TSP的一个实例如下。设 $E'=\{(i, j)|i, j \in V\}$ ，构造一个完全图 $G'=(V, E')$ ，且定义费用函数 c 为： $(i, j) \in E$ 时， $c(i, j)=0$ ； $(i, j) \notin E$ 时， $c(i, j)=1$ 。

可证明 G 有一个哈密顿回路当且仅当 G' 有一个费用为0的旅行售货员回路。事实上，若 G 有一个哈密顿回路 H ，显然 H 也是 G' 的一个旅行售货员回路。由于 H 的每一条边均属于 E ，故每边的费用均为0。因此 H 是 G' 的一个费用为0的旅行售货员回路。反之，若 G' 有一个费用为0的旅行售货员回路 H ，由费用函数 c 的定义知， H 的每边费用均为0，从而 H 的每条边均属于 E 。故 H 为 G 的一条哈密顿回路。

因此，HAM-CYCLE \in pTSP。即旅行售货员问题是NP难的⁶⁴