算法复杂性与NPC问题

问题与算法的描述

图灵机与确定性算法

NP类问题

NP完全问题

证明问题为NP完全的方法

NP困难问题

问题与算法的描述

- **问题**(problem)：有待回答、通常含有几个取值还未确定的自由变量的一般性提问(question)。　表示符号：Π
- **问题的构成**：
  1)对其关于参数的一般性描述；
  2)对该问题的答案所应满足的某些特性的说明。
- **问题的实例**：指定问题中所有参数的具体取值 。表示符号：I
- 旅行商问题：
  参数描述：n个城市$C_1,C_2,\ldots,C_n$，城市间距离$d(C_i,C_j)$
  答案描述：城市的排序：$C_{\pi(1)}, C_{\pi(2)}, \ldots, C_{\pi(n)}$， 最小化目标值:

$$\sum_{1\le i\le n-1} d(C_{\pi(i)}, C_{\pi(i+1)}) + d(C_{\pi(n)}, C_{\pi(1)})$$

- 旅行商问题实例: $n=4$, C_1 , C_2 , C_3 , C_4 , $d(C_1,C_2)=10$, $d(C_1,C_3)=5$, $d(C_1,C_4)=9$, $d(C_2,C_3)=6$, $d(C_2,C_4)=9$, $d(C_3,C_4)=3$
  答案：C_1, C_3, C_4,C_2, 长度: 27

判定问题及描述

- 判定问题：答案只有“是”与“非”两种可能的问题。 Π 实例集合D_Π，回答为“是”的集合 Y_Π
- 描述方法：分成两个部分

  例：用诸如集合、图、函数等各种描述分量来刻画判定问题的一般性例子;

  问：陈述基于一般性例子所提出的一个“是非”提问。
- 旅行商问题：

  例 待访问城市的有限集合$C=\{C_1,C_2,\ldots,C_n\}$、每对城市之间的距离 $d(C_i,C_j)\in Z^+$ 以及一个界 $B\in Z^+$ 。
- **问** 在C中存在一个总长不超过B的、通过所有城市的旅行路线吗？即是否存在的一个排序：$C_{\pi(1)}, C_{\pi(2)}, \ldots, C_{\pi(n)}$， 使得

$$\sum_{1\le i\le n-1} d(C_{\pi(i)}, C_{\pi(i+1)}) + d(C_{\pi(n)}, C_{\pi(1)}) \le B$$

- **算法**(algorithm)是指用来求解某一问题的、带有一般性的一步一步的过程。是计算流程的抽象形式，超越实现细节。

问题和语言

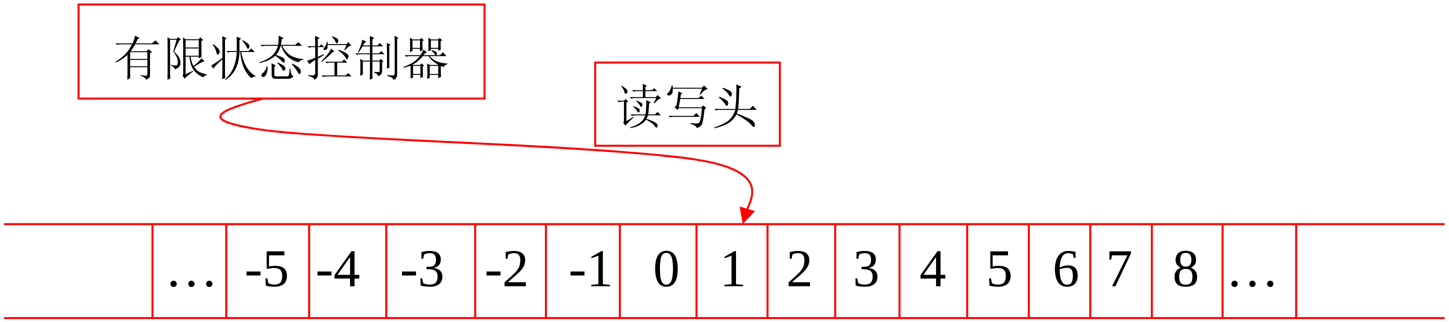
- 算法的性能的合理刻划：计算模型、问题实例的规模
- 计算模型：图灵机（TM）、随机存储机（RAM）等；
- 问题实例的规模：描述或表示它所需要的信息量。
- 编码策略：从某一字符集中选取所需字符构成有限长字符串。

合理的编码策略具有可解码性、简洁性，常利用结构化字符串，通过递归、复合等方式给出。对于问题实例 I，在一个合理的编码策略 e 下，描述该实例的字符串中的字符个数称为该问题的输入长度，也作为该问题实例的输入规模。

- 旅行商问题的一个实例：字符集{C,[,],/,0,1,2,3,4,5,6,7,8,9}，编码字符串：C[1]C[2]C[3]C[4]//10/5/9//6/9//3 ，输入长度为 32。
- 长度多项式相关：Length[I]≤p(|x|) 且 |x|≤q(Length[I])
- 语言：字符集Σ中字符组成的一些有限长字符串的集合L ⊂Σ*
- $L[\Pi, e]=\{x \in \Sigma^* | x$为某个例子$I \in Y_\Pi$在 e 下的编码$\}$

图灵机的构成

- **图灵机：** 一个具有存储载体的、按照具体指令可完成向左或向右移动、放置标记、抹去标记以及在计算终止时停机等四种基本操作的虚拟机器，可以看作是描述算法的语言。
- **确定性单带图灵机(DTM)的组成：** 一个有限状态控制器、一个读写头、一条双向的具有无限多个格的线性带。
- **DTM程序规定如下信息：**
  1. 字符取用集Γ，其包含输入字符集∑及空白符$b\in \Gamma \backslash \Sigma$。
  2. 一个有限状态集Q，包含初始状态q_0和两个停机状态q_Y，q_N
  3. 一个转移函数δ：$(Q\backslash\{q_Y, q_N\})\times\Gamma \rightarrow Q \times\Gamma \times\{l, r\}$



确定性算法

- **图灵机程序的运行**：将输入字符串x一格一个字符地存放在带格1到带格|x|中，所有其它的带格均存放空白字符 b。读写头位于带格1，状态控制器处于初始状态，即开始按转移函数指令一步一步地运行：若当前状态q是停止状态q_Y或q_N，则停止，并分别回答“是”或“非”。若当前状态q既不是q_Y也不是q_N，则按照转移函数

$$\delta(q\ ,\ s) = (q',\ s',\ \Delta)$$

读写头首先擦掉当前带格的字符s，并写上字符s',然后依照Δ=*l*或 *r* 将读写头向左或右移动一格，最后将当前状态更新为q' 。

- 图灵机程序M所识别的语言：

$$L_M=\{x\in\Sigma^*|M\text{作用于}x\text{时，停机于状态}q_Y\}$$

- **算法**：当一个DTM程序对定义于其输入字符集上的所有可能字符串均（在有限步内）停机时，称其为一个算法。
- DTM程序M求解问题Π：M是算法，且 $L_M=L(\Pi, e)$。

复杂性函数定义

- 一个DTM程序M对于输入 x 的计算时间定义为该程序从开始直至进入停机状态为止所运行的步数。
- DTM程序M的时间复杂性函数定义：T_M: $Z^+ \to Z^+$
  $T_M(n)=\max\{m|$存在$x\in\Sigma^*$，$|x|=n$，使得M对x的计算时间为 $m\}$
- 多项式时间DTM程序：存在多项式 $p(x)$，使得 $T_M(n) \le p(n)$
- P-语言类： $P=\{L|$存在多项式时间程序M，使得$L=L_M\}$
- 多项式问题 Π : 存在多项式时间DTM程序M，其在编码策略e下求解Π，即 $L(\Pi,e)\in P$。称Π为P-类问题。
- P-类语言只能描述具有多项式时间确定算法的问题。

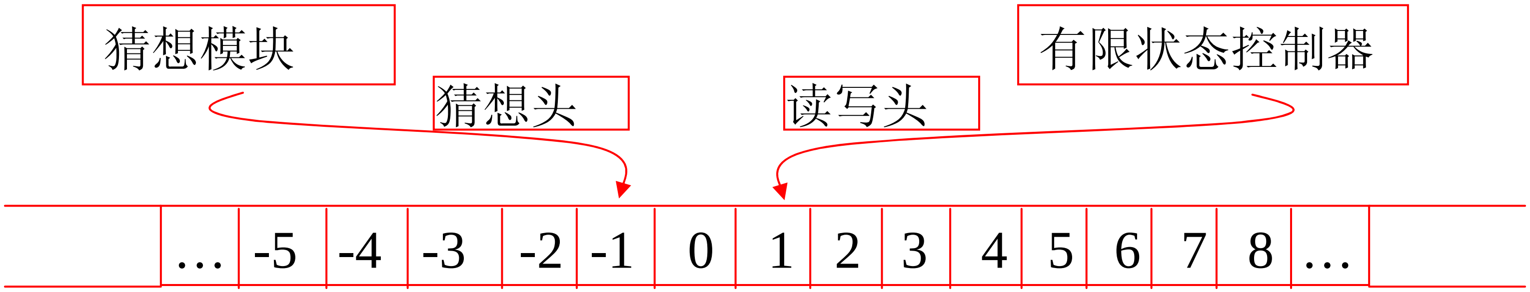
非确定性单带图灵机

- 非确定性图灵机包括多值模型、猜想模块模型
- 多值模型规定如下信息：

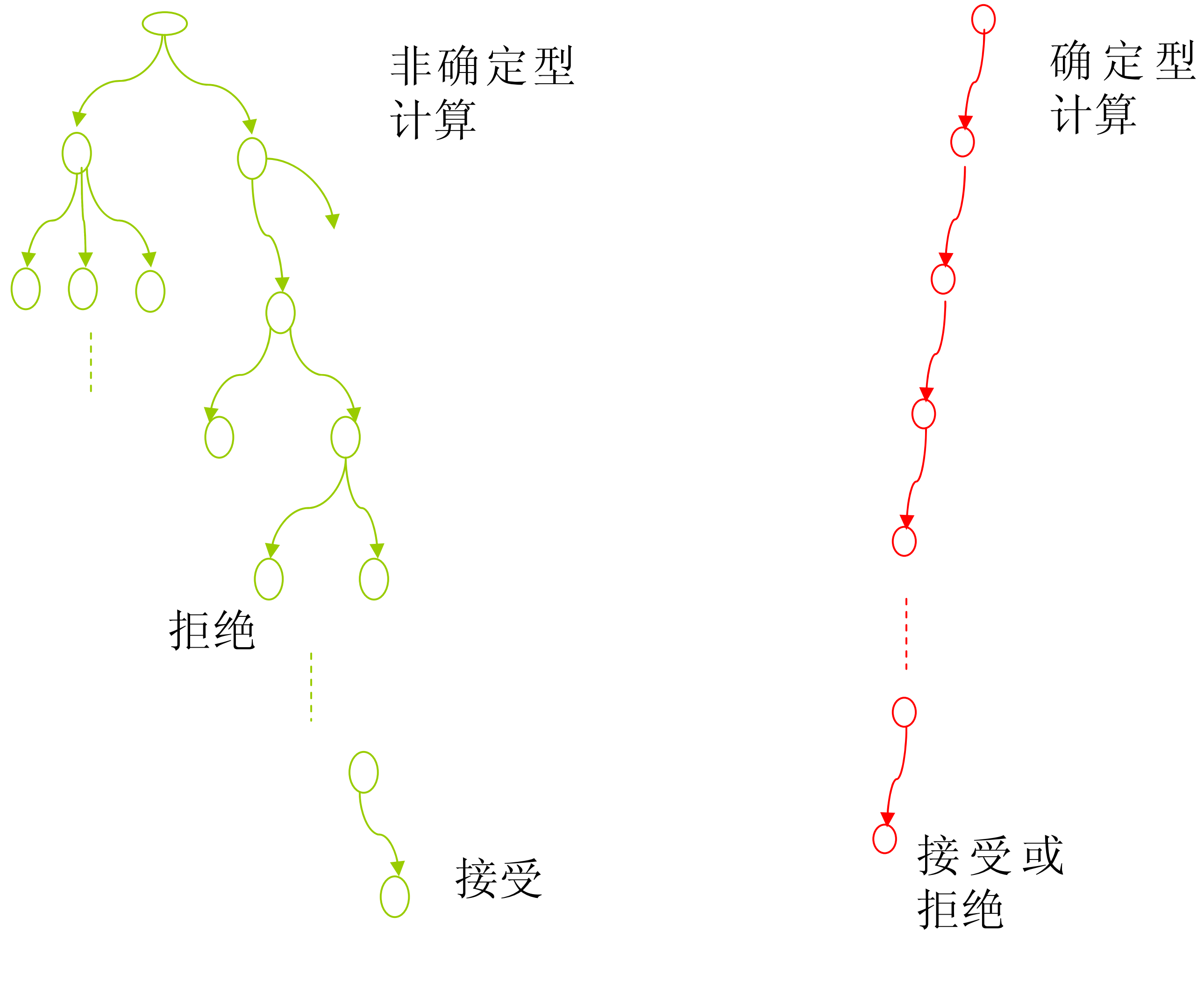
1. 字符取用集Γ，其包含输入字符集Σ及空白符$b\in \Gamma\backslash\Sigma$。
2. 一个有限状态集Q，包括初始状态q_0和两个停机状态q_Y，q_N
3. 一个多值转移函数∂：$(Q\backslash\{q_Y, q_N\})\times\Gamma \rightarrow 2^{Q\times\Gamma\times\{l, r\}}$

$$\partial(q, s) = S \subset Q \times\Gamma\times\{l, r\}$$

- 非确定性图灵机可以被想象为在同一时刻能够独立、并行地完成多种运算（表现在转移函数的多值性）。
- 猜想模块模型：加一个猜想模块



多值性模型与确定性模型示意



猜想模块模型

- 猜想模块有一个只进行写操作的猜想头；
- 计算进程分为两个阶段：猜想阶段和验证阶段
- 首先，输入字符串x被写在从1到|x|的带格中，其余为空白符。猜想头位于带格-1处，有限状态控制器的读写头处于带格1的位置，有限状态控制器处于不起作用状态。
- 其次，猜想模块处于起作用状态，一次一步地指示猜想头：要么在被扫描的带格中写下Γ中的一个字符,并向左移动一格；要么停止
- 猜想模块停止，有限状态控制器即开始起作用，它首先处于初始状态q_0，然后就按照DTM完全相同的规则进行。这是检验阶段，由猜想头写出的字符串也被考虑。停机q_Y，称为可接受的计算。
- 对于一个给定的输入字符串x，NDTM程序M将会做无限多个可能的计算。如果这些计算中至少有一个为可接受的计算，则称NDTM程序M接受字符串 x。

L_M={x | 程序M接受x}－M所识别的语言

NDTM的时间复杂性函数

- 程序NDTM接受x的时间：在M对于x的所有可接受计算中，程序从一开始直到停机状态为止在猜想和检验阶段所进行的步数的最小值 。
- NDTM程序M时间复杂性函数定义：T_M：$Z^+ \to Z^+$

  $T_M(n)$=max{m | 存在x∈L_M，|x|=n，

  使得M接受x所需的计算时间为 m}
- 多项式时间NDTM程序M：存在多项式p(x)，使得$T_M(n) \le p(n)$。
- NP语言类：NP={L|存在多项式时间NDTM程序M，使L=L_M}
- NP类问题Π：存在合理编码策略e，使得L[Π,e]∈NP。
- 任何现有的计算模型都可以通过加上一个类似于NDTM中的带有只写头的猜想模块来扩充，而且所有如此得到的计算模型在多项式时间内可相互转换的意义下将是等价的。没有必要特别提及NDTM模型，我们将简单地说“**多项式时间不确定算法**”，并将NP类语言与所有可用多项式时间不确定算法求解的判定问题等同看待。

无向图的团问题

- 问题描述：

例：给定一个有n个顶点的无向图G=(V,E)及整数 k

问：G是否包含一个具有k个顶点的完全子图（团）？

- 问题实例的字符串表示：G－邻接矩阵，长度：$m=n^2+\log k+1$

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

- 非确定性算法设计

第一阶段：将输入字符串w#v进行分解，计算$n=|w|^{1/2}$以及用v表示的整数k。若输入不具有形式w#v或|w|不是平方数，则拒绝。

第二阶段：非确定性选择V的一个k元子集V' ,并用其集合特征向量表示A[1..n]。

第三阶段：确定性地检查V'的团性质。若V'是一个团则接受，否则拒绝。

团问题选择子集的非确定性算法

- **integer** j,m;
- j:=0;
- **for** i **to** n **do**
-     m:=**choice**(0,1);
-     **case:**
-       m=0: A[i]:=0;
-       m=1: A[i]:=1; j:=j+1;
-     **end{case}**
- **end{for}**
- **if** j≠k **then** failure; **end{if}**

- 第一阶段可在$O(n^2)$内完成。第二阶段可在O(n)内完成；第三阶段可在$O(k^2)$内完成。总的时间复杂度为
  $O(n^2+k^2)$

定理：对于每一个NP问题Π，都存在多项式 $p(x)$ 使得问题Π可以用一个时间复杂度为 $O(2^{p(n)})$ 的确定性算法求解

- **证明**：设A为求解Π的一个多项式时间不确定性算法，其时间复杂性由一个多项式 $q(n)$ 来界定。不失一般性，设q可在多项式时间内被估值。因此，对于长度为 n 的每个被接受的输入，必然存在字符集Γ上长度至多为 $q(n)$ 的某个猜想字符串，使算法A的检验阶段在不多于 $q(n)$ 步内回答“是”。若$|\Gamma|=k$，则所需要考虑的可能猜测的数目最多为 $k^{q(n)}$。对于一个长度为 n 的给定输入，通过应用算法A的确定性检验阶段到相应的 $k^{q(n)}$ 多个可能猜测字符串中的每一个，直到停止或运行 $q(n)$ 步，我们可以肯定地发现A对于该输入是否有一个可接受计算。如果A在该时间界内遇到一个导致可接受计算的猜测串，则该模拟过程回答“是”；否则回答“非”。这显然形成了一个求解Π的确定性算法，而且该算法的时间复杂度将以 $q(n)k^{q(n)}$ 为上界。其等价于 $O(2^{p(n)})$ 。

NP－完全问题(1)

- 在非确定性图灵机上时间复杂性为q(n)的判定问题与在确定性图灵机上时间复杂性为 $q(n)k^{q(n)}$ 的问题相当。

$$P \subset NP，\text{问：} P = NP？$$

- 研究NP类中问题之间的关系，从NP中找出一些具有特定性质的、与P中问题有显著不同的问题。形成了NP－完全理论
- **多项式变换**：语言$L_1 \subset \Sigma_1{}^*$到另一个语言$L_2 \subset \Sigma_2{}^*$的多项式变换是指映射$f: \Sigma_1{}^* \to \Sigma_2{}^*$，满足下面两个条件：

  1. 存在计算 f 的一个多项式时间DTM程序；

  2. 对于所有的$x \in \Sigma_1{}^*$有：$x \in L_1$当且仅当$f(x) \in L_2$。

  记号：$L_1 \propto L_2$。$\Pi_1 \propto \Pi_2$ 如果 $L(\Pi_1, e_1) \propto L(\Pi_2, e_2)$

- NP－完全问题：称一个语言L(判定问题Π)是NP－完全的，如果$L \in NP$($\Pi \in NP$)，而且，对于任意$L' \in NP$($\Pi' \in NP$)都有

$$L' \propto L \ \ (\Pi' \propto \Pi)$$

NP－完全问题(2)

- 几个性质：

1. 若L' $\propto$ L, L$\in$P，则L' $\in$ P;

2. 若L' $\propto$ L，L $\propto$ L", 则 L' $\propto$ L"

3. 若判定问题Π是NP－完全的，P $\neq$ NP，则Π$\in$NP\P;

4. 若L' , L $\in$NP， L' $\propto$ L 则 L' $\in$ NPC $\Rightarrow$ L$\in$NPC。

- 几个典型的NP－完全问题

1. 可满足性问题

例：给定布尔变量之集以及上子句的一个集合C。

问：是否存在的一个真值分配，使得C是可满足的。

2. 图的顶点覆盖问题

例：给定一个图G(V,E)和一个正整数K≤|V| 。

问：是否存在G的一个顶点数不超过K的覆盖？

NP－完全问题(3)

3. Hamilton回路问题

例：给定一个图G(V,E)。

问：G含有一个Hamilton回路吗？

4. 划分问题

例：已知一个有限集合A，其每个元素a都赋予一个权值$s(a)\in Z^+$

问：是否存在A的子集A'使得$\sum_{a\in A'} s(a) = \sum_{a\in A\setminus A'} s(a)$。

5.三元可满足性问题 3SAT

例：给定布尔变量的一个有限集合U及定义于其上的子句集 $C=\{c_1,c_2,\ldots,c_m\}$, 其中$|c_i|=3$, $i=1,2,\ldots,m$。

问：是否存在U之上的一个真赋值，使得C中所有的子句均被满足？

6. 三元精确覆盖问题

例：给定有限集X，$|X|=3q$，以及X的三元子集族C。

问：C含X的一个精确覆盖吗？即存在$C'\subset C$，使得$\cup_{c\in C'} c=X$，而且X中的每个元素恰属于C'的一个子集。

证明新问题是NPC问题的方法(1)

- 证明 $\Pi \in NP$；
- 选取一个已知的NP完全问题 Π' ；
- 构造一个从 Π' 到 Π 的变换 f ；
- 证明 f 为一个多项式变换。
- Cook定理： 可满足性问题是NPC－问题。
- **例1** 证明三元可满足性问题是NPC－问题。

选择满足性问题作为参照物。设布尔变量集 $U=\{u_1,u_2,\cdots,u_n\}$ 及子句集 $C=\{c_1,c_2,\cdots,c_m\}$ 构成SAT的一个一般性例子。我们构造新的布尔变量集 U' 及定义其上的三元子句集 C' ，使得 C' 可满足当且仅当 C 可满足。

对于每个子句 c_j ，设 $c_j=\{z_1\vee z_2\vee\cdots\vee z_k\}$，其中，$z_i\in U\cup\bar{U}$ $\bar{U}$ 表示 U 中变量的非的全体。分三种情况讨论：

1）$k=1$ 情形，此时令

证明新问题是NPC问题的方法(2)

$$U_j'=\{y_j^1,y_j^2\},\quad C_j'=\{z_1\vee y_j^1\vee y_j^2, z_1\vee y_j^1\vee \overline{y}_j^2, z_1\vee \overline{y}_j^1\vee y_j^2, z_1\vee \overline{y}_j^1\vee \overline{y}_j^2\}$$

情形2： $k=2$ ，令

$$U_j'=\{y_j^1\},\quad C_j'=\{z_1\vee z_2\vee y_j^1, z_1\vee z_2\vee \overline{y}_j^1\}$$

情形3: $k=3$，令 $U_j'=\{\ \}$, $C_j'=\{c_j\}$ ；

情形4： $k>3$ ，令 $U_j'=\{y_j^i \mid 1\le i\le k-3\}$ ；

$$C_j'=\{z_1\vee z_2\vee y_j^1\}\cup\{\overline{y}_j^i\vee z_{i+2}\vee y_j^{i+1}\mid 1\le i\le k-4\}\cup\{\overline{y}_j^{k-3}\vee z_{k-1}\vee z_k\}$$

最后，令 $U'=U\cup\bigcup_{i=1}^{m}U_i'$, $C'=\bigcup_{i=1}^{m}C_i'$ ，往证：C 可满足当且仅当C' 可满足

设 $t:U\to\{T,F\}$ 是使C 满足的一个真赋值。以下说明t 可扩充成满足 C' 的一个真赋值：$t':U'\to\{T,F\}$ 。

因为 $U'\backslash U$中的变量可分解成不同的集合 $U_j',1\le j\le m$ ，而每个U_j'的变量仅出现在属于C_j' 的子句中，我们仅需要说明如何将t扩充到各个U_j即可，且证明在上述四种情形的每一种情形下，C_j'中的所有子句均被满足。

证明新问题是NPC问题的方法(3)

若 U_j'属于情形 $k=1$或情形$k=2$，则C_j' 中的子句已被t 所满足，从而可任意地扩展它到 U_j'，例如，对所有的$y \in U_j'$ ，令 $t'(y)=T$

若 U_j是由情形 $k=3$所确定的，那么U_j' 为空集，而t 的赋值已经使 $C_j'=\{c_j\}$中的单个子句取真值。

若 U_j是由情形 $k>3$所确定的，因为t为C 的一种可满足性真赋值，必然存在一个最小的整数 l，使得变量 z_l在 t之下被赋予真值，且 $l \neq 0$。若l 为1或2，则可对 $i:1 \leq i \leq k-3$ 令 $t'(y_j^i)=F$ ；若 l为 $k-1$ 或 k ，则对于$i:1 \leq i \leq k-3$ ，令$t'(y_j^i)=T$ ；其余情况，令 $t'(y_j^i)=T, 1 \leq i \leq l-2$ ，$t'(y_j^i)=F, l-1 \leq i \leq k-3$ 。

容易证明，这些选择将保证C_j'中所有的子句均被满足，进而C'中的所有子句也均被赋值t'所满足。

反之，若t'为C'的一个可满足性真赋值，则不难证明t'对于U 中变量的限制必形成对C 的一个可满足性真赋值。

证明新问题是NPC问题的方法(4)

至此，我们证明了C'可满足当且仅当C可满足。

要证明上述变换可在多项式时间内完成，只需注意到C'中三元子句的数目被$m \cdot n$的一个多项式所界定。也就是说，3SAT例子的大小由SAT例子的大小的一个多项式函数所界定。由此，根据上述构造方法，不难证明它是一个多项式变换。

例2 证明图的顶点覆盖问题VC是NPC-问题

首先，图的顶点覆盖问题是NP问题。因为求解它的不确定性算法只需猜测顶点的一个子集，然后在多项式时间内就可以检验该子集是否包含每条边的至少一个端点，并具有适当的大小，即该顶点子集中顶点的个数不超过限定的值。

选三元可满足性问题做参照物。设

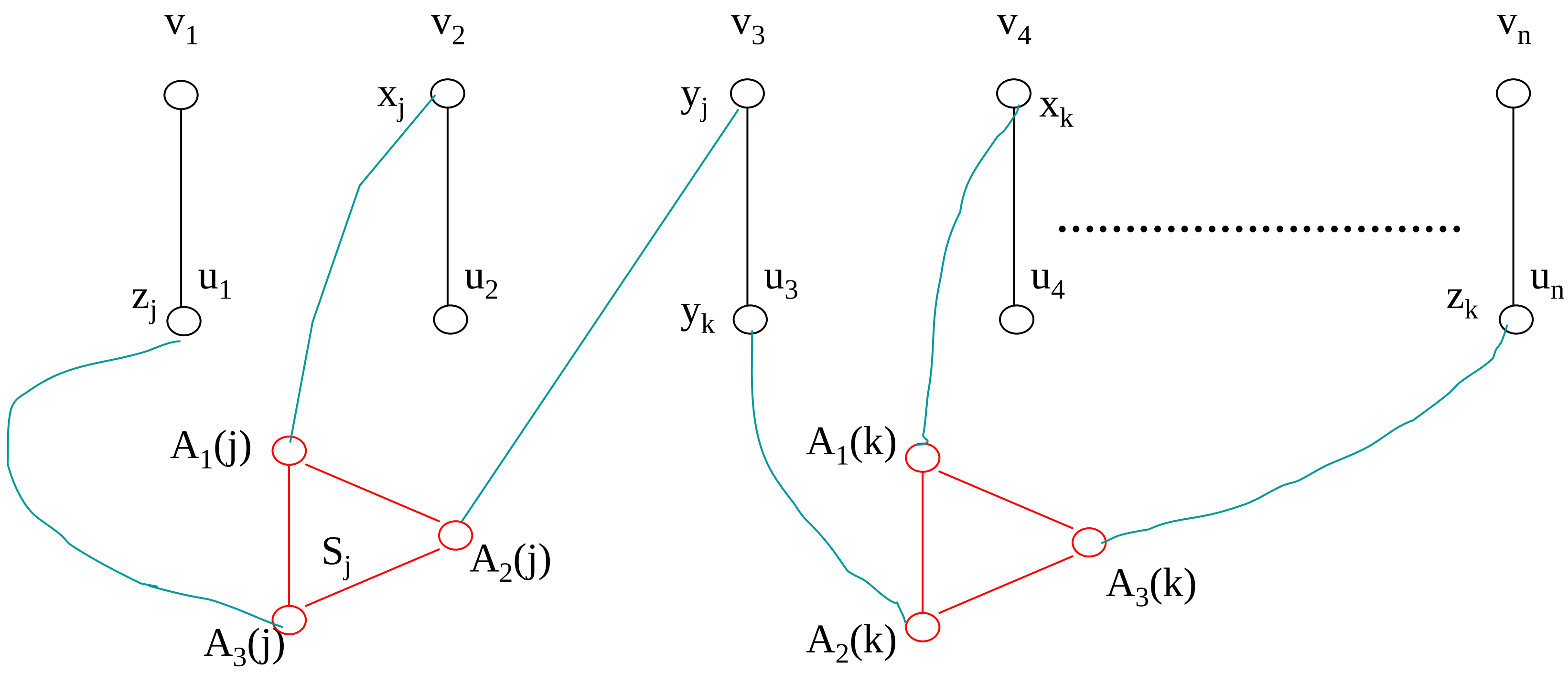
$$\Pi': U=\{u_1,u_2,\cdots,u_n\}, \quad C=\{c_1,c_2,\cdots,c_m\}$$

为3SAT的一个一般性例子。我们要构造一个图 $G=(V,E)$ 和一个正整数 $K \le |V|$，使得G有一个顶点数不超过K的覆盖，当且仅当

证明新问题是NPC问题的方法(5)

C 是可满足的。

对每个变量$u_i \in U$,定义$T_i=(V_i,E_i)$ ， 其中$V_i=\{u_i,\bar{u}_i\},E_i=\{(u_i,\bar{u}_i)\}$； 对于每个子句 $c_j \in C$， 设 $c_j=x_j \vee y_j \vee z_j$ ， 构造一个三角形$S_j=(V'_j,E'_j)$，其中 $V'_j=\{A_1(j),A_2(j),A_3(j)\},E'_j=\{(A_1(j),A_2(j)),(A_1(j),A_3(j)),(A_2(j),A_3(j))\}$ 并且构造新的边集 $E''_j=\{(A_1(j),x_j),(A_2(j),y_j),(A_3(j),z_j)\}$ ， 如下图。



证明新问题是NPC问题的方法(6)

其中 $v_i = \overline{u}_i, i = 1,2,\cdots,n$。令

$$V = \left(\bigcup_{i=1}^{n} V_i\right) \cup \left(\bigcup_{j=1}^{m} V'_j\right), \quad E = \left(\bigcup_{i=1}^{n} E_i\right) \cup \left(\bigcup_{j=1}^{m} E'_j\right) \cup \left(\bigcup_{j=1}^{m} E''_j\right), \quad G = (V, E)$$

$K = n + 2m$，由于$|V| = 2n + 3m, |E| = n + 3m + 3m = n + 6m$，以及$K$均为有限值，上述构造图的过程一定在多项式时间内完成。以下证明：C是可满足的当且仅当G有一个顶点数不超过K的覆盖。

设$V' \subseteq V$是G的一个顶点覆盖，$|V'| \le K$。由图G的构造，V'必然包含每个T_i中至少一个顶点，和每个S_j中至少两个顶点，这已经给出了至少$K = n + 2m$个顶点，故V'必然含有每个T_i中恰好一个顶点和每个S_j中恰好两个顶点。据此给出U的一种真赋值 $t : U \to \{T, F\}$ 如下：

当 $u_i \in V'$ 时，$t(u_i) = T$；而 $\overline{u}_i \in V'$时，$t(u_i) = F$。

为了说明该赋值满足每个子句 $c_j \in C$，考虑 E''_j中的三条边。这些边中恰有两条可被$V'_j \cap V'$中的顶点覆盖，剩下的一条边必由属于V'的某个V_i中的一个顶点覆盖。但这意味着子句c_j中的相应变量，或

证明新问题是NPC问题的方法(7)

者是 u_i 或者是 $\bar{u}_i$，在赋值 t 之下取值为真，从而 c_j 可由 t 满足。反之，如果 $t:U\to\{T,F\}$ 为C的一个可满足性真赋值，我们构造G的一个覆盖如下：对于 V_i 中的顶点，如果 $t(u_i)=T$，选取 u_i，如果 $t(u_i)=F$ 则选取 $\bar{u}_i$；对于 $V'_j\,(1\le j\le m)$ 中的顶点，由于 $c_j=x_j\vee y_j\vee z_j$ 中至少有一个取真值 T，不妨设 $t(x_j)=T$，则在前面诸 V_i 中顶点的选取中，x_j 必然被选取，此时我们在 V'_j 中只需选取顶点 $A_2(j)$、$A_3(j)$ 即可。

上述方法选出的顶点集显然是G的覆盖，而且顶点数恰好为 $K=n+2m$。

例3 证明：Steiner树问题是NPC-问题

例：给定图 $G=(V,E)$，对其每条边 $e\in E$ 都有相应的权 $w(e)\in Z^+$，另外有G的顶点子集 $R\subseteq V$，某个界 $B\in Z^+$。

问：是否存在G的一棵子树 $T=(V_1,E_1)$，使 $R\subseteq V_1\subseteq V, E_1\subseteq E$，而且 $\sum_{e\in E_1}w(e)\le B$？

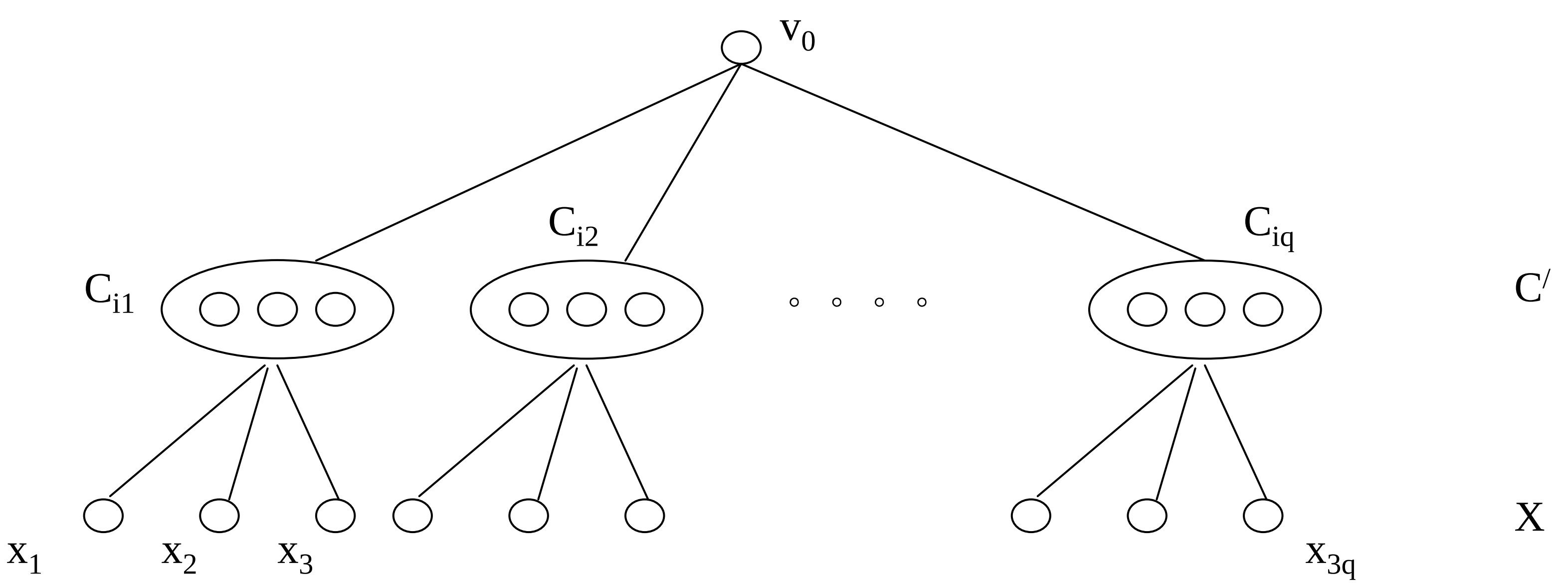
证明新问题是NPC问题的方法(8)

为证明Steiner树问题是NPC的,选X3C问题作为参照物：$X=\{x_1,x_2,\cdots,x_{3q}\}$ 三元子集族 $C=\{c_1,c_2,\cdots,c_n\}$， 构造Steiner树问题的相应例子如下：

取 $G=(V,E)$ ， 其中

$$V=\{v_0\}\cup C\cup X,\ E=\{(v_0,c_i)\,|\,1\le i\le n\}\cup\{(c_i,x_j)\,|\,x_j\in c_i,1\le i\le n,1\le j\le 3q\}$$ ；

令所有边的权值为1，$R=\{v_0\}\cup X$ ，$B=4q$。 显然这一构造过程可在多项式时间内完成。



证明新问题是NPC问题的方法(9)

如果$C'\subseteq C$为X的一个精确覆盖，则取$T=(V_1,E_1)$，其中

$V_1=\{v_0\}\cup\{c_i \mid c_i\in C'\}\cup X$，$E_1=\{(v_0,c_i)\mid c_i\in C'\}\cup\{(c_i,x_j)\mid c_i\in C', x_j\in c_i, 1\le j\le 3q\}$

由于每个x_j恰好出现在C'的某个三元子集里，故T是连通的无圈图，因而是一棵树。而且$|E_1|=4q$。注意到$R\subseteq V_1$，且$\sum_{e\in E_1} w(e)=|E_1|\le B$，由此知Steiner树问题的回答为“是”。

反之，若$T=(V',E')$为$G=(V,E)$的一棵Steiner树，$R\subseteq V'$，则每个$x_j(1\le j\le 3q)$都是T的顶点。不妨设每个x_j都是T的叶顶点，因为，否则的话，顶点x_j的度数大于1，及存在边$(x_j,c_{i_1}),(x_j,c_{i_2})\in E'\subseteq E$删去其中一条边，如$(x_j,c_{i_1})$。由于$(v_0,c_{i_1})$，$(v_0,c_{i_2})$不可能都属于$E'$，否则$T$包含有圈，将不属于$E'$的那条边添入$E'$得到另一棵树$T_1$，它是总权值未变的Steiner树。但这棵树包含X中的顶点作为叶顶点的个数比T的多，这样做下去，必然找到一棵Steiner树，在其上，所有X中的顶点都是叶顶点。再由连通性，每个x_j恰好和某一个c_i在T中邻接，令$C'=\{c_i \mid (c_i,x_j)\in T, 1\le j\le 3q\}$，则$C'$显然是$X$的一个精确覆盖

证明新问题是NPC问题的方法(10)

例4 证明 0/1背包判定问题是NPC问题

例：给定一个有限集合 X，对每一个$x \in X$，对应一个值$w(x) \in Z^+$和相应的值 $p(x) \in Z^+$。另外，还有一个容量约束$M \in Z^+$和一个价值目标 $K \in Z^+$。

问：是否存在 X 的一个子集 $X' \subseteq X$ 使得$\sum_{x \in X'} w(u) \leq M$，而且$\sum_{x \in X'} p(x) \geq K$

- 显然，0/1背包问题是NP问题。现在，我们考虑特殊的 0/1 背包问题：对所有的$x \in X$有$w(x) = p(x)$, 且取 $M = K = \frac{1}{2}\sum_{x \in X} w(x)$。

显然，这个0/1背包判定问题回答为"是"当且仅当集合X 的划分问题回答为"是"。可见，划分问题恰是0/1 背包问题的特例。从而由划分问题是NPC问题推得0/1背包判定问题是NPC问题。

NP难问题(1)

- **图灵归约：** 设Π_1和Π_2是两个判定问题，我们说Π_1在多项式时间内可图灵归约为Π_2，记做 $\Pi_1 \propto_T \Pi_2$，如果存在求解Π_1的一个算法 A_1，它多次调用求解Π_2的算法A_2作为其子程序，而且假定，若每次调用该子程序A_2均需用单位时间，则 A_1为一个多项式时间算法。称A_1为从Π_1到Π_2的多项式归约（也称图灵归约）。
- 图灵归约的定义也可不限于判定问题，因而可以适用于最优化问题等更广的一类问题。如果判定问题Π_1可以归约到Π_2，则Π_2至少和Π_1一样难。
- **NP难问题：** 对于问题Π，如果存在一个NP完全问题Π'，使得 $\Pi' \propto_T \Pi$，则称Π是NP困难的。
- NP难问题不要求是NP类问题。
- 第 k重子集问题

  例：已知整数 $c_1, c_2, \cdots, c_n, k, L$;

  问：存在 k个不同的子集 $S_1, S_2, \cdots, S_k \subseteq \{1, 2, \cdots, n\}$，使得对于 $i = 1, \cdots, k$ 有 $\sum_{j \in S_i} c_j \ge L$？

NP难问题(2)

利用划分问题(NP完全问题)证明第k最重子集问题是NP难问题

事实上，设 $S=\{c_1,c_2,\cdots,c_n\}$ 是划分问题的一般例子，假设我们已经有个算法 $A[S,k,L]$ 可以用来解第 k 最重子集问题，则可设计出求解划分问题的算法如下：

首先，若 $\sum_{i=1}^{n} c_i$ 为奇数，则立即推出回答问题否。现在，假定 $\sum_{i=1}^{n} c_i$ 是偶数，并令 $L=\frac{1}{2}\sum_{i=1}^{n} c_i$ ，用算法 $A[S,k,L]$ 作为子程序，按下列二分搜索技术确定重至少是L 的不同子集的数目$K*$：

1) 令 $K_{\min}=0, K_{\max}=2^n$；

2) 若 $K_{\max}-K_{\min}=1$，置 $K*=K_{\min}$，且停机；

3) 令 $K=(K_{\min}+K_{\max})/2$，并调用算法 $A[S,K,L]$。

4) 若回答为“是”，则令 $K_{\min}=K$ 转2)；否则，令 $K_{\max}=K$ ，转2)

以上过程通过恰好n 次调用$A[S,K,L]$,即可找到 $K*$ 。至此，只需再调用一次该算法即可给出所考虑划分问题例子的答案，即调用 $A[S,K*,L+1]$ 。若该调用的答案为“是”，则S中所有重至少为L的子集也必满足其重至少为L+1，因此，S没有重为L的子集，故对划分问题的回答为“否”。相反，若该调用的回答为“否”，则意味着存在

NP难问题(3)

某一个S的子集，使得其重为L，对划分问题的回答为“是”。

由上述迭代过程容易看出，若假设每次调用算法A只需要单位时间，则以上即给出了求解划分问题的一个多项式时间算法。也就是说，我们证明了划分问题可多项式时间归约到第k个最重子集问题。

- 旅行商问题是NP难问题

旅行商问题不是NP的。因为对其解的任一猜想，要检验它是否是最优的，需要同所有其它的环游比较，这样的环游会有指数多个，因而不可能在多项式时间内完成。

用图的Hamilton回路问题(NPC问题)证明旅行商问题是NP难的

已知无向图G=(V,E),|V|=n,构造其对应的旅行商问题如下：

$$d_{ij}=\begin{cases}1, & \text{if } (v_i,v_j)\in E\\ 2, & \text{otherwise}\end{cases}$$

这一变换可以在多项式时间内完成，而且，图G有Hamilton回路的充要条件是上述构建的旅行商问题有解，且其解对应的路程长度为n。故我们证明了对称旅行商问题是NP困难的