第9章 NP完全性

- P类与NP类
- 多项式时间变换与NP完全性
- 几个NP完全问题
- 用NP完全性理论分析问题
- NP难度

算法的时间复杂度

函数 f 和 g 是多项式相关的: 如果存在多项式 p 和 q 使得对任意的 $n\in N$, $f(n) \le p(g(n))$ 和 $g(n) \le q(f(n))$.

例如　$n\log n$ 与 n^2, n^2+2n+5 与 n^{10} 都是多项式相关的,

　　　$\log n$ 与 n, n^5 与 2^n 不是多项式相关的.

问题Π 的实例 I 的规模: I 的二进制编码的长度, 记作$|I|$.

定义　如果存在函数 f:N→N使得 对任意的规模为 n 的实例 I, 算法 A 对 I 的运算在 $f(n)$ 步内停止, 则称算法 A的时间复杂度为 $f(n)$.

多项式时间算法: 以多项式为时间复杂度.

易解的问题: 有多项式时间算法.

难解的问题: 不存在多项式时间算法.

几点说明

1. 当采用合理的编码时, 输入的规模都是多项式相关的.

“合理的”是指在编码中不故意使用许多冗余的字符.

例如, 设实例 I 是一个无向简单图 $G = <V,E>$,

V={ a, b, c, d }, E={ (a,b), (a,d), (b,c), (b,d), (c,d) }

用邻接矩阵表示, 编码 e_1=0101/1011/0101/1110/, 长度 20.

用关联矩阵表示, 编码 e_2=11000/10110/00101/01011/, 长度 24.

G 有n个顶点m条边,

用邻接矩阵时$|I|=n(n+1)$, 用关联矩阵时$|I|=n(m+1)$.

两者多项式相关.

2. 自然数应采用k ($k\geq 2$)进制编码, 不能采用一进制编码.

n 的二进制编码有$\lceil \log_2(n+1) \rceil$位, 一进制编码有$n$位, 两者不是多项式相关的.

北京大学

几点说明

3. 时间复杂度常表成计算对象的某些自然参数的函数，如图的顶点数或边数的函数. 实例的二进制编码的长度与这些自然参数通常是多项式相关的.

4. 运行时间通常是计算执行的 操作指令数，执行的指令数与实际运行时间是多项式相关的.

(1) 要求每一条指令的执行时间是固定的常数.

(2) 规定一个基本操作指令集，可由位逻辑运算与、或、非组成，任何可用这个基本操作指令集中常数条指令实现的操作都是合理的指令，由有限种合理的指令构成的操作指令集是合理的操作指令集.

在上述约定下, 算法是否是多项式时间的与采用的编码和操作指令集无关， 从而一个问题是易解的、还是难解的也与采用的编码和操作指令集无关.

易解的问题与难解的问题

易解的问题.

如排序、最小生成树、单源最短路径等

已证明的难解问题.

(1) 不可计算的, 即根本不存在求解的算法, 如希尔伯特第十问题——丢番图方程(有一个或几个变量的整系数方程)是否有整数解.

(2) 有算法 但至少需要指数或更多的时间或空间，如带幂运算的正则表达式的全体性, 即任给字母表*A*上的带幂运算的正则表达式*R*, 问:$\langle R\rangle=A^*$?这个问题至少需要指数空间.

既没有找到多项式时间算法、又没能证明是难解的问题.

如哈密顿回路问题、货郎问题、背包问题等

北京大学

判定问题

判定问题: 答案只有两个——是, 否.

判定问题 $\Pi = <D_\Pi, Y_\Pi>$, 其中D_Π是实例集合, $Y_\Pi \subseteq D_\Pi$是所有答案为“Yes”的实例.

哈密顿回路 (HC): 任给无向图G, 问G有哈密顿回路吗?

货郎问题 (TSP): 任给 n 个城市, 城市 i 与城市 j 之间的正整数距离 $d(i,j)$, $i \neq j$, $1 \leq i, j \leq n$, 以及正整数 D, 问有一条每一个城市恰好经过一次最后回到出发点且长度不超过 D 的巡回路线吗? 即, 存在 1, 2, …, n 的排列 σ 使得

$$\sum_{i=1}^{n-1} d(\sigma(i), \sigma(i+1)) + d(\sigma(n), \sigma(1)) \leq D ?$$

0-1背包的判定问题与优化问题

0-1背包: 任给 n 件物品和一个背包, 物品 i 的重量为 w_i, 价值为 v_i, $1 \le i \le n$, 以及背包的重量限制 B 和价值目标 K, 其中 w_i, v_i, B, K 均为正整数, 问能在背包中装入总价值不少于 K 且总重量不超过 B 的物品吗? 即, 存在子集 $T \subseteq \{1, 2, \dots, n\}$ 使得

$$\sum_{i \in T} w_i \le B \quad \text{且} \quad \sum_{i \in T} v_i \ge K ?$$

搜索问题、 组合优化问题与判定问题的对应.

如果搜索问题、 组合优化问题有多项式时间算法, 则对应的判定问题也有多项式时间算法; 通常反之亦真.

组合优化问题与判定问题

组合优化问题 Π^* 由3部分组成:

(1) 实例集D_{Π^*}

(2) $\forall I \in D_{\Pi^*}$, 有一个有穷非空集 $S(I)$, 其元素称作 I 的可行解

(3) $\forall s \in S(I)$, 有一个正整数 $c(s)$, 称作 s 的值

如果 $s^* \in S(I)$, 对所有的 $s \in S(I)$, 当Π^*是最小 (大) 化问题时,

$$c(s^*) \le c(s) \qquad (c(s^*) \ge c(s))$$

则称 s^*是 I 的最优解, $c(s^*)$是 I 的最优值, 记作$\mathrm{OPT}(I)$.

Π^*对应的判定问题 $\Pi = \langle D_\Pi, Y_\Pi \rangle$定义如下:

$$D_\Pi = \{ (I, K) \mid I \in D_{\Pi^*}, K \in \mathrm{Z}^* \}$$, 其中Z*是非负整数集合.

当Π^*是最小化问题时, $Y_\Pi = \{ (I, K) \mid \mathrm{OPT}(I) \le K \}$;

当Π^*是最大化问题时, $Y_\Pi = \{ (I, K) \mid \mathrm{OPT}(I) \ge K \}$.

北京大學

P类与 NP类

定义 所有多项式时间可解的判定问题组成的问题类称作 **P类**.

定义 设判定问题 $\Pi=\langle D,Y\rangle$, 如果存在两个输入变量的多项式时间算法 A 和多项式 p，对每一个实例 $I\in D$, $I\in Y$ 当且仅当存在 t, $|t|\le p(|I|)$, 且 A 对输入 I 和 t 输出"Yes", 则称Π是多项式时间可验证的, A是Π的多项式时间验证算法, 而当 $I\in Y$ 时, 称 t 是 $I\in Y$ 的证据.

由所有多项式时间可验证的判定问题组成的问题类称作 **NP类(nondeterministic polymial)**.

非确定型多项式时间算法

非确定型多项式时间算法

(1) 对给定的实例 I, 首先“猜想”一个 t, $|t| \le p(|I|)$

(2) 然后检查 t 是否是证明 $I \in Y$ 的证据

(3) 猜想和检查可以在多项式时间内完成

(4) 当且仅当 $I \in Y$ 时能够正确地猜想到一个证据 t

*注：非确定型多项式时间算法不是真正的算法

定理 P⊆NP

问题：P=NP?

多项式时间变换

定义 设判定问题$\Pi_1=<D_1,Y_1>$, $\Pi_2=<D_2,Y_2>$. 如果函数 f : $D_1\rightarrow D_2$ 满足条件:

(1) f 是多项式时间可计算的;

(2) 对所有的 $I\in D_1$, $I\in Y_1 \Leftrightarrow f(I)\in Y_2$;

则称 f 是 Π_1 到 Π_2 的多项式时间变换. 如果存在 Π_1 到 Π_2 的多项式时间变换, 则称 Π_1 可多项式时间变换到 Π_2, 记作 $\Pi_1\le_p \Pi_2$.

例 HC$\le_p$TSP.

证 对 HC 的每一个实例 I: 无向图 $G=<V, E>$, TSP 对应的实例 $f(I)$ 为: 城市集 V, 任意两个不同的城市 u 和 v 之间的距离

$$d(u,v)=\begin{cases}1, & 若(u,v)\in E,\\ 2, & 否则,\end{cases}$$

以及界限 $D = |V|$.

最大生成树$\le_p$最小生成树

最小生成树: 任给连通的无向赋权图 $G=<V,E,W>$ 以及正整数 B, 其中权 $W:E\to Z^+$, 问有权不超过 B 的生成树吗?

最大生成树: 任给连通的无向赋权图 $G=<V,E,W>$ 以及正整数 D, 其中权 $W:E\to Z^+$, 问 G 有权不小于 D 的生成树吗?

例 最大生成树$\le_p$最小生成树.

证 任给最大生成树的实例 I:连通的无向赋权图 $G=<V, E,W>$ 和正整数D, 最小生成树的对应实例 $f(I)$: 图$G'=<V, E,W'>$和正整数 $B=(n-1)M-D$, 其中

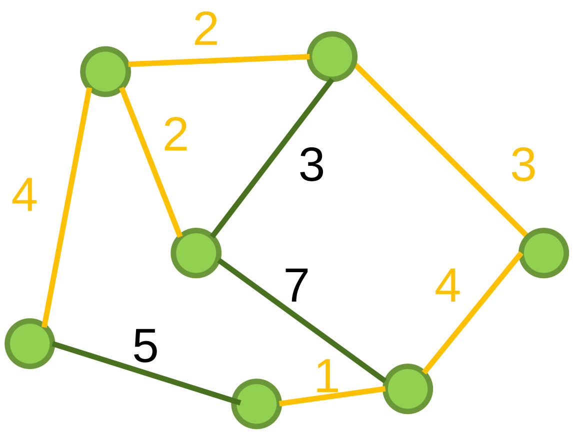
$$n=|V|,\quad M=\max\{\, W(e) \mid e\in E \,\}+1,\quad W'(e)=M-W(e)$$

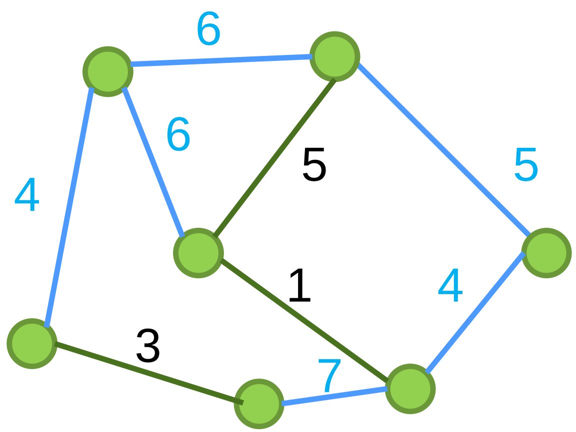
如果存在G的生成树T, 使得 $\sum_{e\in T} W(e)\ge D$, 则

$$\sum_{e\in T} W'(e)=(n-1)M-\sum_{e\in T} W(e)\le (n-1)M-D=B.$$

反之亦然. f 多项式时间可计算.

变换实例





***D*=12**

最大生成树*T* 的实例 *G*

***B*=6×8–12=36**

最小生成树*T* 的实例*G*'

$$M = 8,\quad W(T) = 16 \ge 12,\quad W'(T) = 32 \le 36$$

$$\sum_{e \in T} W'(e) = (n-1)M - \sum_{e \in T} W(e)$$

$\le_p$的性质

定理　$\le_p$具有传递性. 即, 设 $\Pi_1\le_p\Pi_2$, $\Pi_2\le_p\Pi_3$, 则$\Pi_1\le_p\Pi_3$.

证　设 Π_i=<D_i,Y_i>, i=1,2,3, f 和 g分别是 Π_1 到 Π_2 和 Π_2 到 Π_3 的多项式时间变换. 对每一个$I\in D_1$, 令$h(I)=g(f(I))$.

计算 f 和 g 的时间上界分别为多项式 p 和 q, 不妨设 p 和 q 是单调递增的. 计算 h 的步数不超过 $p(|I|) + q(|f(I)|)$. 输出作为合理的指令, 一步只能输出长度不超过固定值 k 的字符串, 因而 $|f(I)| \le k\,p(|I|)$. 于是,

$$p(|I|) + q(|f(I)|) \le p(|I|) + q(kp(|I|)),$$

得证h是多项式时间可计算的.

对每一个$I\in D_1$,

$$I\in Y_1 \Leftrightarrow f(I)\in Y_2 \Leftrightarrow h(I) = g(f(I))\in Y_3,$$

得证 h是 Π_1 到 Π_3 的多项式时间变换.

$\leq_p$的性质

定理　设 $\Pi_1 \leq_p \Pi_2$, 则 $\Pi_2 \in$P 蕴涵 $\Pi_1 \in$P.

证　设 $\Pi_1=<D_1,Y_1>$, $\Pi_2=<D_2,Y_2>$, f 是 Π_1 到 Π_2 的多项式时间变换, A 是计算 f 的多项式时间算法. 又设 B 是 Π_2 的多项式时间算法. 如下构造 Π_1 的算法 C:

(1) 对每一个 $I\in D_1$, 应用 A 得到 $f(I)$,

(2) 对 $f(I)$ 应用 B,

(3) C 输出“Yes”当且仅当 B 输出“Yes”.

推论　设 $\Pi_1 \leq_p \Pi_2$, 则 Π_1 是难解的蕴涵 Π_2 是难解的.

由最小生成树$\in$P, 得知最大生成树$\in$P.

如果 TSP$\in$P, 则 HC$\in$P. 反过来, 如果 HC 是难解的, 则 TSP 也是难解的.

NP完全性

定义 如果对所有的 $\Pi' \in \mathrm{NP}$, $\Pi' \le_p \Pi$, 则称 Π 是NP难的. 如果 Π 是 NP 难的且 $\Pi \in \mathrm{NP}$, 则称 Π 是 NP完全的.

定理 如果存在NP难的问题 $\Pi \in \mathrm{P}$, 则 P=NP.

推论 假设P≠NP, 那么, 如果 Π 是NP难的, 则 $\Pi \notin \mathrm{P}$.

定理 如果存在NP难的问题 Π' 使得 $\Pi' \le_p \Pi$, 则 Π 是NP难的.

推论 如果 $\Pi \in \mathrm{NP}$并且存在 NP完全问题 Π' 使得 $\Pi' \le_p \Pi$, 则Π是NP完全的.

证明NP完全性的“捷径”

(1) 证明$\Pi \in \mathrm{NP}$;

(2) 找到一个已知的NP完全问题Π', 并证明$\Pi' \le_p \Pi$.

SAT问题与Cook定理

合式公式是由变元, 逻辑运算符以及圆括号按照一定的规则组成的表达式. 变元和它的否定称作文字. 有限个文字的析取称作简单析取式. 有限个简单析取式的合取称作合取范式.

给定每一个变元的真假值称作一个赋值. 如果赋值 t 使得合式公式F为真, 则称t是F的成真赋值，这时也称F是可满足的.

例如 $F_1=(x_1\vee x_2)\wedge(\neg x_1\vee x_2\vee x_3)\wedge\neg x_2$ 是一个合取范式.

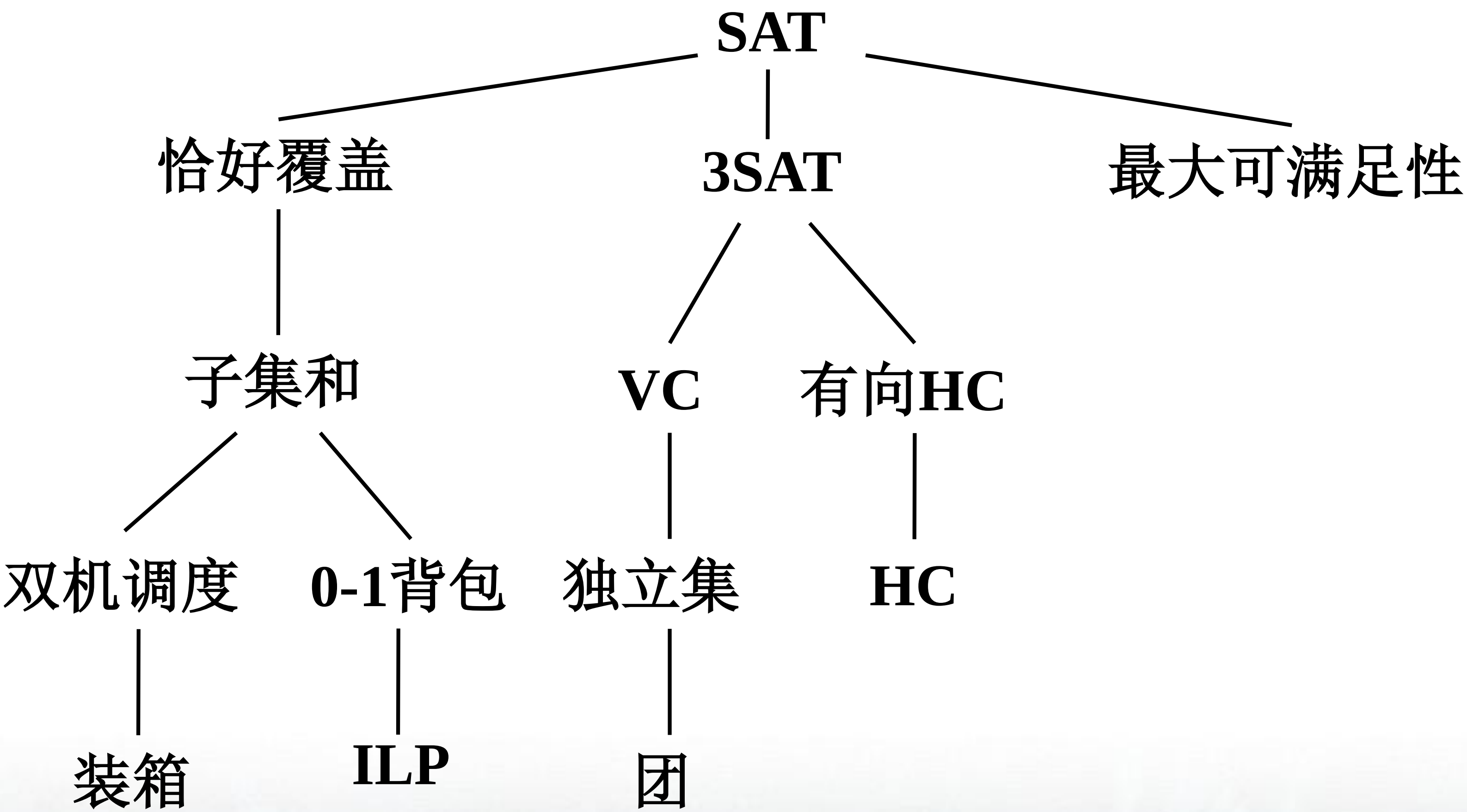
令$t(x_1)=1$, $t(x_2)=0$, $t(x_3)=1$是F_1的成真赋值, F_1是可满足的.

$F_2=(x_1\vee\neg x_2\vee x_3)\wedge(\neg x_1\vee\neg x_2\vee x_3)\wedge x_2\wedge\neg x_3$不是可满足的.

可满足性问题 (SAT): 任给一个合取范式F, 问F是可满足的吗?

定理 (Cook-Levin定理) SAT是NP完全的.

几个NP完全问题



最大可满足性与三元可满足性

最大可满足性(MAX-SAT): 任给关于变元$x_1, x_2,\ldots, x_n$的简单析取式$C_1,C_2,\ldots,C_m$及正整数K, 问存在关于变元$x_1, x_2,\ldots, x_n$的赋值使得$C_1,C_2,\ldots,C_m$中至少有K个为真吗?

设判定问题$\Pi=<D,Y>$, $\Pi'=<D',Y'>$, 如果$D'\subseteq D$, $Y'= D'\cap Y$, 则Π'是Π的特殊情况, 称作Π的子问题.

例如

“给定一个平面图G, 问G是哈密顿图吗?”是HC的子问题.

SAT是MAX-SAT的子问题: 取$K=m$.

MAX-SAT

限制法: 如果已知Π的某个子问题Π'是NP难的, 则Π也是NP难的——一般情况不会比特殊情况容易. 容易把Π'多项式时间变换到Π: 只需把Π'的实例I看作Π特殊情况的实例, 即可得到Π对应的实例.

定理　MAX-SAT是NP完全的.

证　MAX-SAT的非多项式时间算法: 猜想一个赋值, 检查是否有K个简单析取式满足.

要证 SAT$\leq_p$MAX-SAT. 任给SAT的实例I: 关于变元 $x_1, x_2,\ldots, x_n$的合取范式$F=C_1\wedge C_2\wedge\ldots\wedge C_m$, 其中$C_1,C_2,\ldots,C_m$是简单析取式, 对应的MAX-SAT的实例$f(I)$: 简单析取式$C_1,C_2,\ldots,C_m$和正整数$K=m$.

3SAT

3元合取范式: 每一个简单析取式恰好有3个文字的合取范式.

三元可满足性(3SAT): 任给一个3元合取范式F, 问F是可满足的吗?

定理 3SAT是NP完全的.

证 显然 3SAT$\in$NP.

要证 SAT $\le_p$3SAT. 任给一个合取范式F, 要构造对应的 3元合取范式$F' = f(F)$, 使得F是可满足的当且仅当F'是可满足的.

设$F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$, 对应的$F' = F_1' \wedge F_2' \wedge \ldots \wedge F_m'$, F_j' 是对应C_j 的合取范式, 并且

C_j是可满足的当且仅当 F_j'是可满足的.

证明

(1) $C_j=z_1$. 引入两个新变元y_{j1}, y_{j2}, 令

$F_j' =(z_1\vee y_{j1}\vee y_{j2})\wedge(z_1\vee\neg y_{j1}\vee y_{j2})\wedge(z_1\vee y_{j1}\vee\neg y_{j2})\wedge(z_1\vee\neg y_{j1}\vee\neg y_{j2})$.

(2) $C_j=z_1\vee z_2$. 引入一个新变元y_j, 令

$$F_j' = (z_1\vee z_2\vee y_j)\wedge(z_1\vee z_2\vee\neg y_j).$$

(3) $C_j=z_1\vee z_2\vee z_3$. 令 $F_j' = C_j$.

(4) $C_j=z_1\vee z_2\vee\ldots\vee z_k$, $k\geq 4$. 引入k-3个新变元$y_{j1}, y_{j2},\ldots,y_{j(k-3)}$, 令

$$F_j' =(z_1\vee z_2\vee y_{j1})\wedge(\neg y_{j1}\vee z_3\vee y_{j2})\wedge (\neg y_{j2}\vee z_4\vee y_{j3})$$
$$\wedge\ldots\wedge(\neg y_{j(k-4)}\vee z_{k-2}\vee y_{j(k-3)})\wedge (\neg y_{j(k-3)}\vee z_{k-1}\vee z_k).$$

设赋值 t 满足C_j, 则存在i使得 $t(z_i)=1$. 当i=1或2时, 令$t(y_{js})=0$ $(1\leq s\leq k-3)$; 当$i=k-1$或k时, 令$t(y_{js})=1$ $(1\leq s\leq k-3)$; 当$3\leq i\leq k-2$时, 令$t(y_{js})=1(1\leq s\leq i-2)$, $t(y_{js})=0(i-1\leq s\leq k-3)$. 则有 $t(F_j')=1$.

易见构造F'可在多项式时间完成.

变换实例

布尔变元：$x_1, x_2, \ldots, x_5$

$$C=(x_1 \vee x_2) \wedge x_3 \wedge (x_1 \vee x_2 \vee \neg x_3 \vee x_4 \vee x_5)$$

变元：$y_{11}, y_{21}, y_{22}, y_{31}, y_{32}$

$$C' = (x_1 \vee x_2 \vee y_{11}) \wedge (x_1 \vee x_2 \vee \neg y_{11}) \wedge (x_3 \vee y_{21} \vee y_{22})$$
$$\wedge (x_3 \vee \neg y_{21} \vee y_{22}) \wedge (x_3 \vee y_{21} \vee \neg y_{22}) \wedge (x_3 \vee \neg y_{21} \vee \neg y_{22})$$
$$\wedge (x_1 \vee x_2 \vee y_{31}) \wedge (\neg y_{31} \vee \neg x_3 \vee y_{32}) \wedge (\neg y_{32} \vee x_4 \vee x_5)$$

局部替换法 要证$\Pi_1 \le_p \Pi_2$. 当Π_2是Π_1的子问题或两者的结构相似时, 往往可以把Π_1的实例的每一个子结构替换成对应的Π_2实例的子结构.

总结

- 易解问题和难解问题
- 判定问题和组合优化问题
- 多项式时间变换
- P和NP问题
- NPC问题的证明
  - SAT、MAX-SAT、3-SAT问题