

计算理论

教材:

[S] 唐常杰等译, Sipser著, 计算理论导引, 机械工业.

参考资料:

[L] Lewis等著, 计算理论基础, 清华大学.

计算理论

第三部分 计算复杂性

第7章 时间复杂性

1. 时间复杂性

$\{ 0^k 1^k \mid k \geq 0 \}$ 的时间复杂性分析

2. 不同模型的运行时间比较

单带与多带 确定与非确定

3. P类与NP类

4. NP完全性及NP完全问题

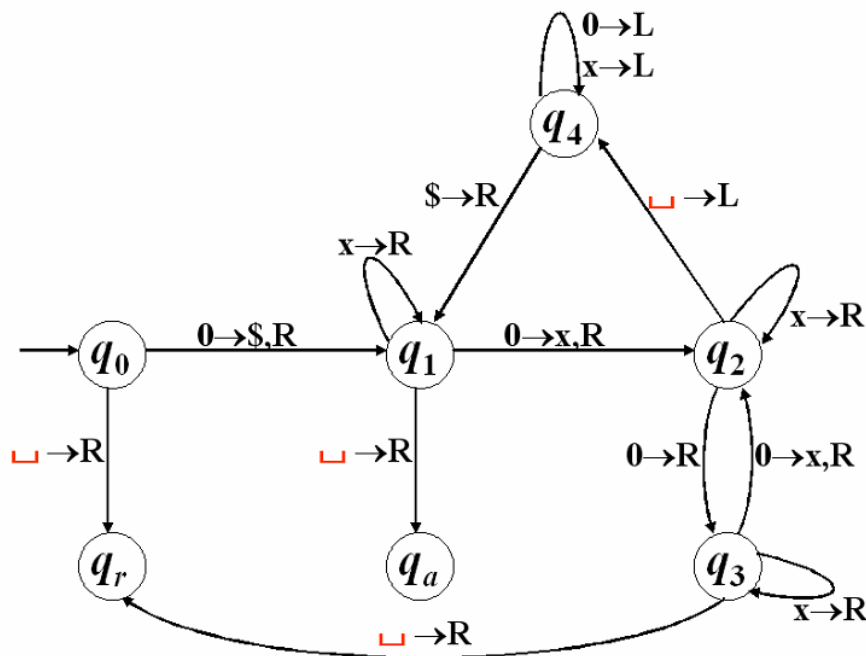
一. 时间复杂度

- 时间复杂度定义
- $\{ 0^k 1^k \mid k \geq 0 \}$ 的时间复杂度分析

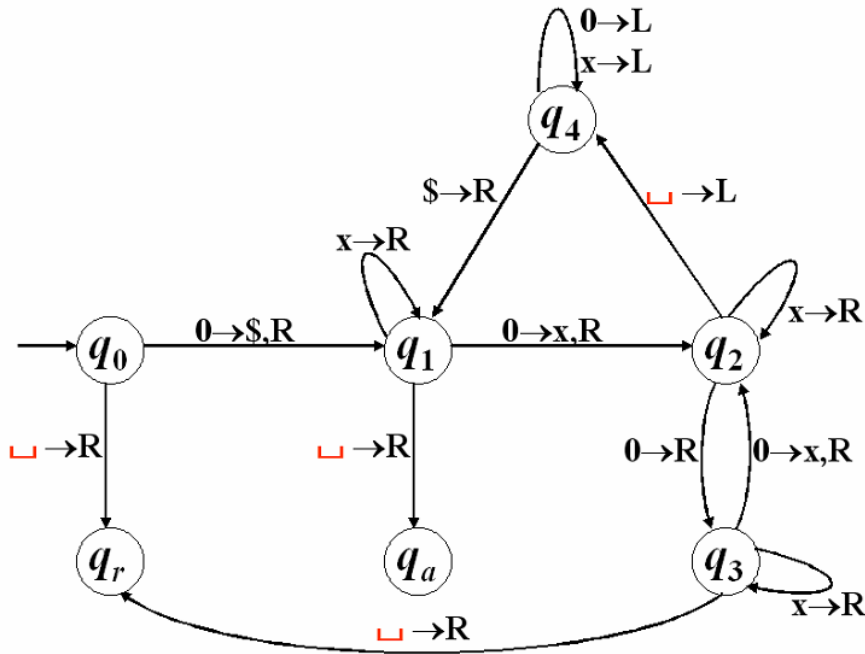
时间复杂性(P153)

- 判定器M的运行时间或时间复杂度是 $f:N \rightarrow N$,
 $f(n)$ 是M在所有长为n的输入上运行的最大步数.
- 若 $f(n)$ 是M的运行时间, 则称
M在时间 $f(n)$ 内运行 或 M是 $f(n)$ 时间图灵机

举例:



时间复杂性



$q_0 0000$

$\$q_1 000$

...

$\$xq_2 00$

...

$\$x0q_3 0$

$q_4 \$xxx_$

$\$x0xq_2_$

...

$\$x0q_4 x_$

...

$\$xq_4 0x_$

$\$xxx_q_a$

$\$q_4 x0x_$

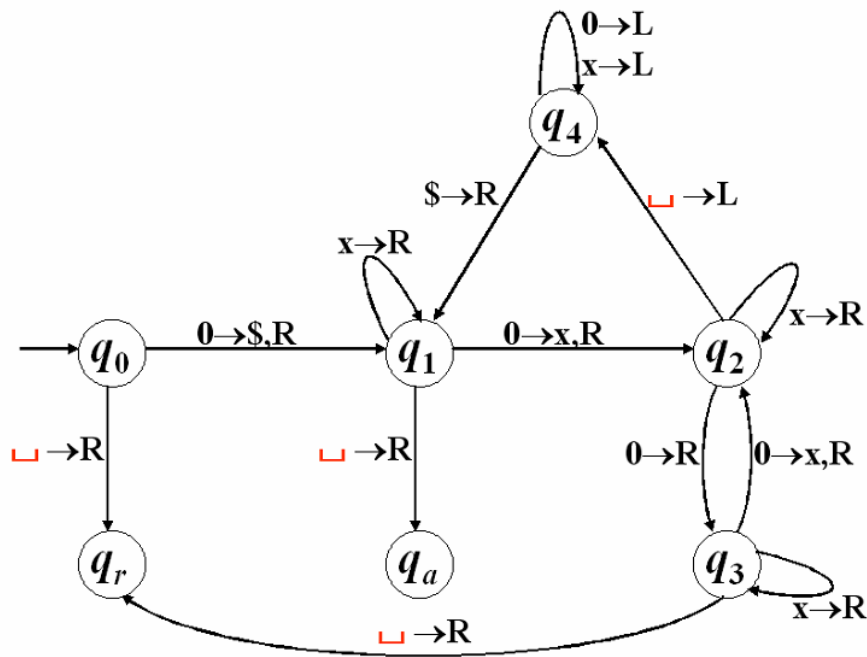
$q_4 \$x0x_$

$\$q_1 x0x_$

...

时间复杂性(P91)

- 判定器M的运行时间或时间复杂度是 $f:N \rightarrow N$,
 $f(n)$ 是M在所有长为n的输入上运行的最大步数.



$$f(1) = 2$$

$$f(2) = 7$$

$$f(3) = 4, \dots$$

$$f(2^k) = (2k+1)2^k + 1,$$

$$f(2n+1) = 2n+2, \dots$$

$$n+1 \leq f(n) \leq 3n \log n$$

大O与小o记法(P154)

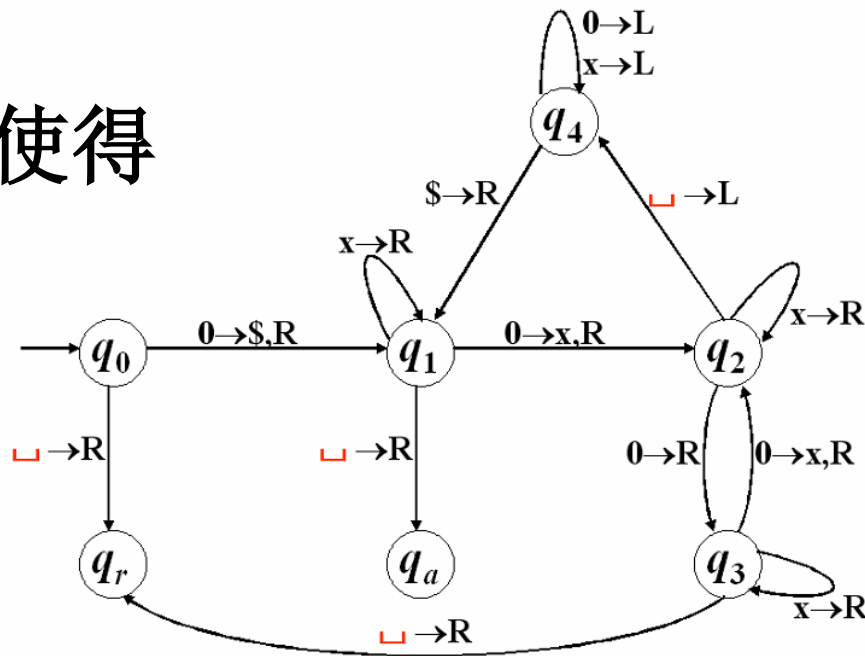
对于函数 $f, g: \mathbf{N} \rightarrow \mathbf{R}^+$,

记 $f(n) = O(g(n))$, 若存在 $c > 0$ 使得

$$\overline{\lim}_{n \rightarrow \infty} \frac{f(n)}{g(n)} \leq c$$

记 $f(n) = o(g(n))$, 若

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$$



$$f(n) = O(n \log n)$$

图灵机 M_1 (P155)

讨论语言 $A = \{ 0^k 1^k \mid k \geq 0 \}$ 的复杂性:

M_1 = “对输入串 w :

- 1) 扫描带, 如果在1的右边发现0, 则拒绝.
- 2) 如果0和1都在带上, 就重复下一步.
- 3) 扫描带, 删除一个0和一个1.
- 4) 如果带上同时没有0和1, 就接受.”

时间分析: $f(1) = 3$, $f(6) = 42$, $f(n) \geq 1$,

(1) $2n = O(n)$, (4) $n = O(n)$,

{ (2) $2n = O(n)$ + (3) $2n = O(n)$ } $\times (n/2) = O(n^2)$

所以 M_1 的运行时间是 $O(n^2)$.

000111
*00111
\$00x11
\$\$0xx1
\$\$\$xxx
accept
 $12 + 7 \times 3 + 3 = 36$

000011
*00011
\$000x1
\$\$00xx
\$\$\$0xx
reject
 $12 + 9 \times 2 + 4 = 34$

001100
*01100
reject
5

时间复杂性类(P155)

定义: 对于函数 $t:N \rightarrow N$,

时间复杂性类 $\text{TIME}(t(n))$ 定义为:

$\text{TIME}(t(n)) = \{ L \mid \text{存在 } O(t(n)) \text{ 时间 TM 判定 } L \}$

因为 M_1 是时间 $O(n^2)$ 图灵机,

所以 $A = \{0^k 1^k : k \geq 0\} \in \text{TIME}(n^2)$.

是否存在更快的 TM 判定 A 呢?

图灵机 M_2 (P155)

M_2 = “对输入串 w :

- 1) 扫描带, 若1的右边有0, 则拒绝.
- 2) 若0, 1都在带上, 重复以下步骤.
- 3) 检查带上0, 1总数的奇偶性, 若是奇数, 就拒绝.
- 4) 再次扫描带,
第1个0开始, 隔1个0删除1个0;
第1个1开始, 隔1个1删除1个1.
- 5) 若带上同时没有0和1, 则接受.
否则拒绝.”

```
0000011111
*000011111
$0x0xx1x1x
$xx0xxx1x
$xxxXXXXXX
accept
20×3+10=70
```

```
000111
*00111
$0xx1x
$xxxxx
accept
12×2+6=30
```

```
0001111
*001111
$0xx1x1
reject
```

```
00111111
*0111111
$0x1x1x1
$xxx1xx
$xxxx1xx
reject
```

```
00011111
*0011111
$0xx1x1x
$xxxx1x
reject
```

图灵机 M_2 (P155)

M_2 = “对输入串 w :

1) 扫描带, 若1的右边有0, 则拒绝. $O(n)$

2) 若0,1都在带上, 重复以下步骤. $O(n)$

3) 检查带上0,1总数的奇偶性,
若是奇数, 就拒绝. $O(n)$

4) 再次扫描带,
第1个0开始, 隔1个0删除1个0; $O(n)$
第1个1开始, 隔1个1删除1个1.

5) 若带上同时没有0和1, 则接受. $O(n)$

否则拒绝.”

$\times \log n$

总时间:

$O(n \log n)$

$\{0^k 1^k | k \geq 0\} \in \text{TIME}(n \log n)$ (P156)

由 M_2 知道 $A \in \text{TIME}(n \log n)$. 有没有更快的 TM 识别 A ?

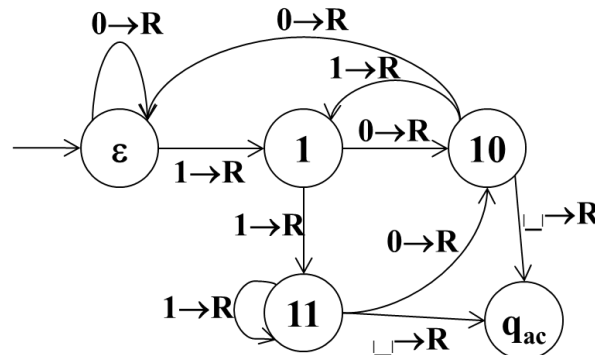
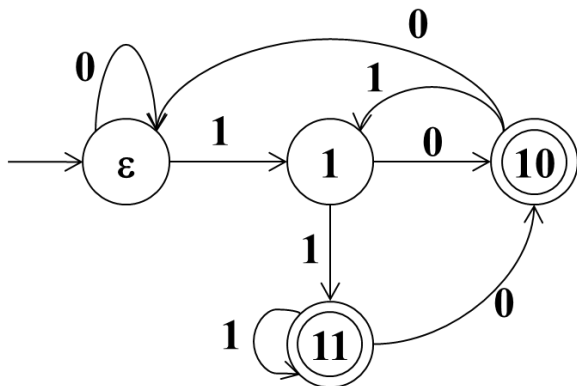
对于单带确定图灵机, 由

定理: 时间 $o(n \log n)$ 的单带图灵机判定的语言
是正则语言.

$\text{TIME}(o(n \log n)) \subseteq \text{正则语言类} \subseteq \text{TIME}(n) \subseteq \text{TIME}(o(n \log n))$

正则语言类 = $\text{TIME}(n) = \text{TIME}(o(n \log n))$

非正则语言 $\{0^k 1^k | k \geq 0\} \notin \text{TIME}(o(n \log n))$



计算理论

第三部分 计算复杂性

第7章 时间复杂性

1. 时间复杂性

$\{ 0^k 1^k \mid k \geq 0 \}$ 的时间复杂性分析

2. 不同模型的运行时间比较

单带与多带 确定与非确定

3. P类与NP类

4. NP完全性及NP完全问题

二.

不同模型的时间复杂度比较

- 单带与多带
- 确定与非确定

单带与多带运行时间比较(P156-7)

$\{ 0^k 1^k \mid k \geq 0 \}$ 有 $O(n)$ 时间双带图灵机

M_3 = “对输入串 w :

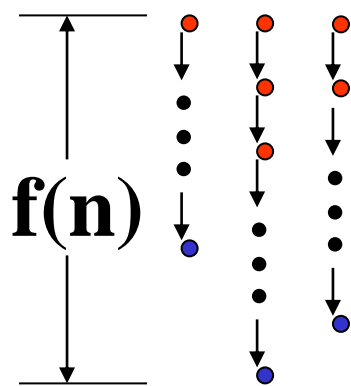
- 1) 扫描1带,如果在1的右边发现0,则拒绝.
- 2) 将1带的1复制到2带上.
- 3) 每删除一个1带的0就删除一个2带的1.
- 4) 如果两带上同时没有0和1,就接受.”

定理: 设函数 $t(n) \geq n$, 则每个 $t(n)$ 时间多带TM
和某个 $O(t^2(n))$ 时间单带TM等价.

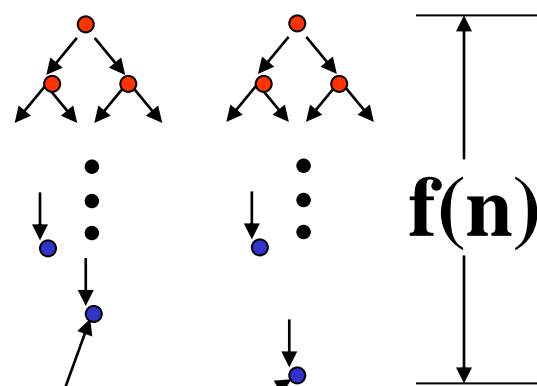
非确定判定器的运行时间(P157)

定义: 对非确定型判定器 N , 其运行时间 $f(n)$ 是在**所有**长为 n 的输入上, **所有**分支的最大步数.

时间 $f(n)$ 确定图灵机



时间 $f(n)$ 非确定图灵机

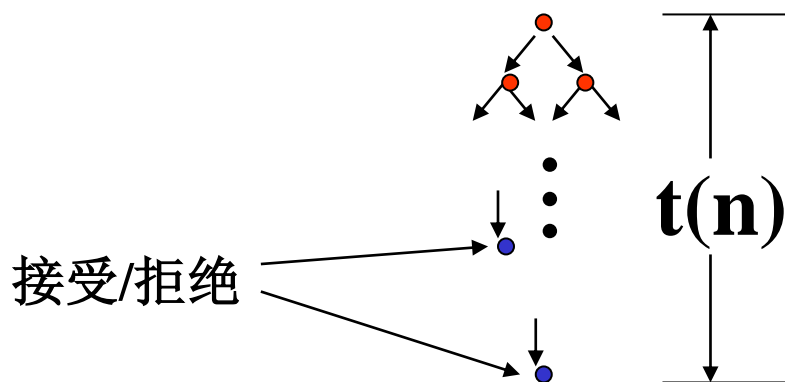


接受 或 拒绝

NTM的运行时间(P158)

定义: 对非确定型判定器 N , 其运行时间 $f(n)$ 是在所有长为 n 的输入上, 所有分支的最大步数.

定理: 设 $t(n) \geq n$, 则每个 $t(n)$ 时间NTM 都有一个 $2^{O(t(n))}$ 时间单带确定TM与之等价.



定理: 设 $t(n) \geq n$, 则 $\text{NTIME}(t(n)) \subseteq \text{TIME}(2^{O(t(n))})$

计算理论

第三部分 计算复杂性

第7章 时间复杂性

1. 时间复杂性

$\{ 0^k 1^k \mid k \geq 0 \}$ 的时间复杂性分析

2. 不同模型的运行时间比较

单带与多带 确定与非确定

3. P类与NP类

4. NP完全性及NP完全问题

三. P与NP

多项式时间(P158)

运行时间相差多项式可以认为是小的
相差指数可以认为是大的.

例如: n^3 与 2^n ,对于 $n=1000$.

有关素性测试: $\text{Prime} = \{ p \mid p \text{是素数} \}$

如何编码? 一进制,二进制,十进制?

典型的指数时间算法来源于蛮力搜索.

有时通过深入理解问题可以避免蛮搜.

2001年Prime被证明存在多项式时间算法.

P类(P159)

定义:P是单带确定TM在
多项式时间内可判定的问题,即

$$P = \cup_k \text{TIME}(n^k)$$

P类的重要性在于:

- 1) 对于所有与单带确定TM等价的模型,P不变.
- 2) P大致对应于在计算机上实际可解的问题.
研究的核心是一个问题是否属于P类.

NP类(P165)

定义:NP是单带非确定TM在
多项式时间内可判定的问题,即

$$\mathbf{NP} = \cup_k \mathbf{NTIME}(n^k)$$

$$\mathbf{EXP} = \cup_k \mathbf{TIME}(2^{O(n^k)})$$

$$\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{EXP}$$

$$\mathbf{P} \subset \mathbf{EXP}$$

一些P问题(P159)

有些问题初看起来不属于P

求最大公因子: 欧几里德算法,
辗转相除法

模p指数运算 $a^b \bmod p$

上下文无关语言 有 $O(n^3)$ 判定器

素性测试 等等

以增加空间复杂性来减小时间复杂性

快速验证(P163)

HP = {<G,s,t> | G是包含从s到t的
哈密顿路径的有向图}

CLIQUE = {<G,k> | G是有k团的无向图}

目前没有快速算法,但其成员是可以快速验证的.

注意:**HP**的补可能不是可以快速验证的.

快速验证的特点:

1. 只需要对语言中的串能快速验证.
2. 验证需要借助额外的信息:证书,身份证.

NP问题(P165)

团:无向图的完全子图(所有节点都有边相连).

CLIQUE = { $\langle G, k \rangle$ | G 是有 k 团的无向图 }

定理: **CLIQUE** \in NP.

N=“对于输入 $\langle G, k \rangle$,这里 G 是一个图:

- 1)非确定地选择 G 中 k 个节点的子集 c .
- 2)检查 G 是否包含连接 c 中节点的所有边.
- 3)若是,则接受;否则,拒绝.”

哈密顿路径问题 $HP \in NP$ (对比P164)

HP = { $\langle G, s, t \rangle$ | G 是包含从 s 到 t 的
哈密顿路径的有向图 }

P 时间内判定 **HP** 的 **NTM**:

N_1 = “对于输入 $\langle G, s, t \rangle$:

- 1) 非确定地选 G 的所有节点的排列 p_1, \dots, p_m .
- 2) 若 $s = p_1, t = p_m$, 且对每个 i , (p_i, p_{i+1}) 是 G 的边, 则接受; 否则拒绝.”

P与NP(P166)

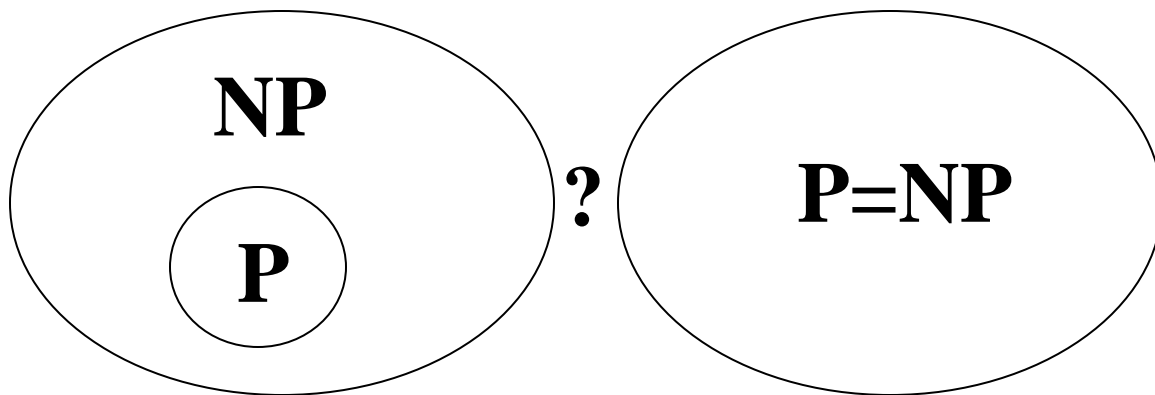
P=成员资格可以**快速判定**的语言类.

NP=成员资格可以**快速验证**的语言类.

显然有 $P \subseteq NP$

但是否有 $P = NP$?

看起来难以想象,但是现在没有证明.



当代数学与
理论计算机
共同的难题.

计算理论

第三部分 计算复杂性

第7章 时间复杂性

1. 时间复杂性

$\{ 0^k 1^k \mid k \geq 0 \}$ 的时间复杂性分析

2. 不同模型的运行时间比较

单带与多带 确定与非确定

3. P类与NP类

4. NP完全性及NP完全问题

四. NP完全

- NP完全性的定义
- SAT是NP完全问题
- 一些NP完全问题

NP完全性(P166)

Cook(美)和Levin(苏联)于1970's证明

NP中某些问题的复杂性与

整个NP类的复杂性相关联, 即:

若这些问题中的任一个找到P时间算法, 则 $P=NP$.

这些问题称为NP完全问题.

理论意义: 两方面

1) 研究P与NP关系可以只关注于一个问题的算法.

2) 可由此说明一个问题目前还没有快速算法.

合取范式(P167-8)

- **布尔变量**: 取值为1和0(True, False)的变量.
- **布尔运算**: AND(\wedge), OR (\vee), NOT (\neg). **布尔公式**.
例: $\phi_1 = ((\neg x) \wedge y) \vee (x \wedge (\neg z))$, $\phi_2 = (\neg x) \wedge x$
- 称 ϕ **可满足**, 若存在布尔变量的0,1赋值使得 $\phi=1$. 例 ϕ_1, ϕ_2 .
 ϕ **不可满足** $\Leftrightarrow \neg \phi$ 永真
- **文字**: 变量或变量的非,如 x 或 $\neg x$.
- **子句**: 由 \vee 连接的若干文字,如 $x_1 \vee (\neg x_2) \vee x_3 \vee x_4$.
- **合取范式(cnf)**: 由 \wedge 连接的若干子句,如
 $((\neg x_1) \vee x_2 \vee (\neg x_3)) \wedge (x_2 \vee (\neg x_3) \vee x_4 \vee x_5) \wedge ((\neg x_4) \vee x_5)$
- **k-cnf (conjunctive normal form)**
每个子句的文字数不大于k: 3cnf, 2cnf

可满足问题SAT(P167-8)

- 可满足性问题:

$\text{SAT} = \{ \langle \phi \rangle \mid \phi \text{ 是可满足的布尔公式} \}$ **NP完全**

- 二元可满足性问题:

$\text{2SAT} = \{ \langle \phi \rangle \mid \phi \text{ 是可满足的2cnf} \}$ **$\in \text{P}$**

- 三元可满足性问题:

$\text{3SAT} = \{ \langle \phi \rangle \mid \phi \text{ 是可满足的3cnf} \}$ **NP完全**

二元可满足问题 2SAT $\in P$ (ex7.23)

1. 当2cnf中有子句是单文字 x , 则反复执行(直接)清洗

1.1 由 x 赋值, 1.2 删去含 x 的子句, 1.3 删去含 $\neg x$ 的文字
若清洗过程出现相反单文子子句, 则清洗失败并结束

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

$$\rightarrow (x_3 \vee \neg x_2) \wedge (\neg x_2) \wedge (x_3 \vee x_4) \wedge (\neg x_3 \vee x_5) \wedge (\neg x_4 \vee \neg x_5) \wedge (\neg x_3 \vee x_4)$$

$$\rightarrow (x_3 \vee x_4) \wedge (\neg x_3 \vee x_5) \wedge (\neg x_4 \vee \neg x_5) \wedge (\neg x_3 \vee x_4)$$

2. 若无单文字子句, 则任选变量赋真/假值各(赋值)清洗一次
若两次都清洗失败, 则回答不可满足.

$$x_3=1 \rightarrow (x_5) \wedge (\neg x_4 \vee \neg x_5) \wedge (x_4) \rightarrow (\neg x_4) \wedge (x_4) \text{ 失败}$$

$$x_3=0 \rightarrow (x_4) \wedge (\neg x_4 \vee \neg x_5) \rightarrow (\neg x_5) \rightarrow \emptyset \text{ 成功}$$

3. 若成功清洗后有子句剩下, 则继续2. 否则, 回答可满足.

注: 见[S]习题7.23, 作者答案与清洗算法等价. 贪心.

$3SAT \in NP(P173)$

三元可满足性问题:

$$3SAT = \{ \langle \phi \rangle \mid \phi \text{ 是可满足的 3cnf} \}$$

P时间内判定3SAT的NTM:

N=“对于输入 $\langle \phi \rangle$, ϕ 是一个3cnf公式,

1)非确定地选择各变量的赋值T.

2)若在赋值T下 $\phi=1$, 则接受;否则拒绝.”

第2步在公式长度的多项式时间内运行.

3SAT ∈ P? (补充)

$3SAT = \{ \langle \phi \rangle \mid \phi \text{ 是可满足的 3cnf} \}$

清洗算法对 3cnf 是否有效? 举例对比:

$(x_3 \vee \neg x_3) \wedge (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$

$x_3=1$ 清洗无矛盾; $x_1=0$ 和 1 都清洗失败, 不可满足.

$(x_3 \vee \neg x_3) \wedge (\neg x_3 \vee x_1 \vee x_2) \wedge (\neg x_3 \vee x_1 \vee \neg x_2) \wedge (\neg x_3 \vee \neg x_1 \vee x_2) \wedge (\neg x_3 \vee \neg x_1 \vee \neg x_2)$

$x_3=1$ 清洗无矛盾; $x_1=0$ 和 1 都清洗失败, 返 $x_3=0$

3cnf 清洗不能避免搜索, 指数时间.

目前还不知道 3SAT 是否属于 P.

归约引理: 若 $A \leq_p B$ 且 $B \in P$, 则 $A \in P$ (P168)

- 定义: 多项式时间可计算函数 $f: \Sigma^* \rightarrow \Sigma^*$. (例如 $f(u) = u0$)
若 \exists 多项式时间图灵机, $\forall w$ 输入, 停机时带上的串为 $f(w)$

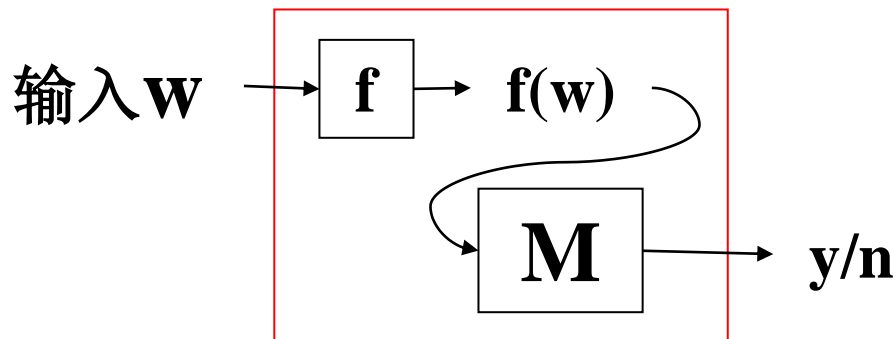
- 定义: 称 A 可多项式时间映射归约到 B ($A \leq_p B$),

若存在多项式时间可计算函数 $f: \Sigma^* \rightarrow \Sigma^*$,

$$\forall w \in \Sigma^*, w \in A \Leftrightarrow f(w) \in B.$$

函数 f 称为 A 到 B 的多项式时间归约.

通俗地说: f 将 A 的实例编码转换为 B 的实例编码.



利用 f 和 B 的判定器
构造 A 的判定器

C-L定理: $\text{SAT} \in \text{P} \Leftrightarrow \text{P} = \text{NP}$ (P167-8)

- **定义: 多项式时间可计算函数** $f: \Sigma^* \rightarrow \Sigma^*$. (例如 $f(u) = u0$)
若 \exists 多项式时间图灵机, $\forall w$ 输入, 停机时带上的串为 $f(w)$
- **定义: 称A可多项式时间映射归约到B ($A \leq_p B$),**
若存在**多项式时间可计算函数** $f: \Sigma^* \rightarrow \Sigma^*$,
$$\forall w \in \Sigma^*, w \in A \Leftrightarrow f(w) \in B.$$

函数 f 称为A到B的**多项式时间归约**.
通俗地说: f 将A的实例编码转换为B的实例编码.
- **Cook-Levin定理:** 对任意 $A \in \text{NP}$ 都有 $A \leq_p \text{SAT}$.
- **归约引理:** 若 $A \leq_p B$, 且 $B \in \text{P}$, 则 $A \in \text{P}$.
- **推论:** 若 $\text{SAT} \in \text{P}$, 则 $A \in \text{P}$, 则 $\text{NP} = \text{P}$.

归约定理: 若 $A \leq_P B$ 且 $B \in P$, 则 $A \in P$ (P168)

证明: 设 $f: \Sigma^* \rightarrow \Sigma^*$ 是 A 到 B 的 P 时间归约,

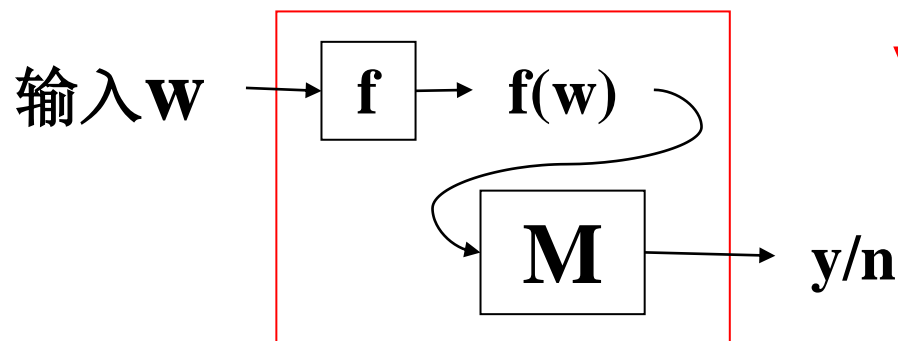
B 有 P 时间判定器 M , 则

$N =$ “输入 w , 计算 $M(f(w))$, 输出 M 的运行结果”

在多项式时间内判定 A .

问题: 若 f 是 n^a 时间归约, M 是 n^b 时间判定器, 则 N 时间?

设 $|w| = n$, 则 $|f(w)| \leq n^a$, 则 $M(f(w))$ 时间 $\leq n^{ab}$.



$$\forall w \in \Sigma^*, w \in A \Leftrightarrow f(w) \in B.$$

利用 f 和 B 的判定器
构造 A 的判定器

定理: $3SAT \leq_p CLIQUE$ (P168)

$3SAT = \{ \langle \phi \rangle \mid \phi \text{ 是可满足的 3cnf 公式} \}$

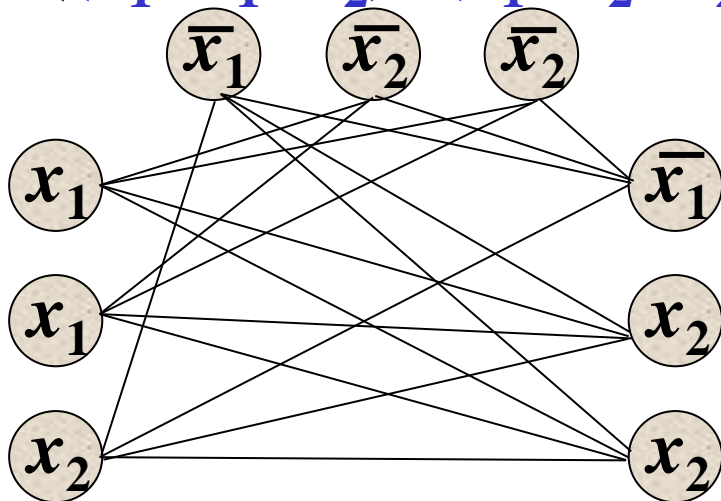
$CLIQUE = \{ \langle G, k \rangle \mid G \text{ 是有 } k \text{ 团的无向图} \}$.

证明: 设 $\phi = (a_1 \vee b_1 \vee c_1) \wedge \dots \wedge (a_k \vee b_k \vee c_k)$, 有 k 个子句.

令 $f(\phi) = \langle G, k \rangle$, G 有 k 组节点, 每组 3 个;

同组节点无边相连, 相反标记无边相连.

例: $f((x_1 \vee x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2 \vee x_2)) = \langle G, 3 \rangle$



需证: $\langle \phi \rangle \in 3SAT$

\Leftrightarrow

$\langle (G, k) \rangle \in CLIQUE$

$\forall \phi, \phi \in 3SAT \Leftrightarrow f(\phi) \in CLIQUE$ (P169)

$\langle \phi \rangle (\langle (x_1 \vee x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2 \vee x_2) \rangle) \in 3SAT$

$\Leftrightarrow \exists$ 变量赋值 $(x_1=0, x_2=1)$ 使得 $\phi=1$

$\Leftrightarrow \exists k$ 团 (每组挑一个真顶点得到 k 团, 非同组非相反)

$\Leftrightarrow f(\phi) (\langle G, 3 \rangle) \in CLIQUE.$

F 在 $|\langle \phi \rangle|$ 的多项式时间内可计算:

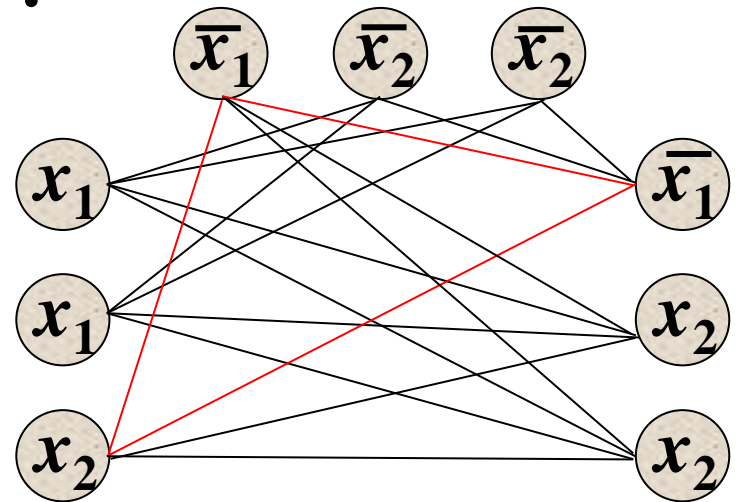
设 ϕ 的长度 $3k = O(k)$,

则 G 的顶点数 $3k = O(k)$,

G 的边数是 $O(k^2)$

可见 $f(\phi) = \langle G, k \rangle$ 的构造

可在 k 的多项式时间内完成.



NP完全性(P169,175)

- 定义:语言B称为**NP完全**的(NPC),若它满足:

1) $B \in NP$; 2) $\forall A \in NP$, 都有 $A \leq_p B$.

- 定理1: $A \leq_p B + B \in P \Rightarrow A \in P$.

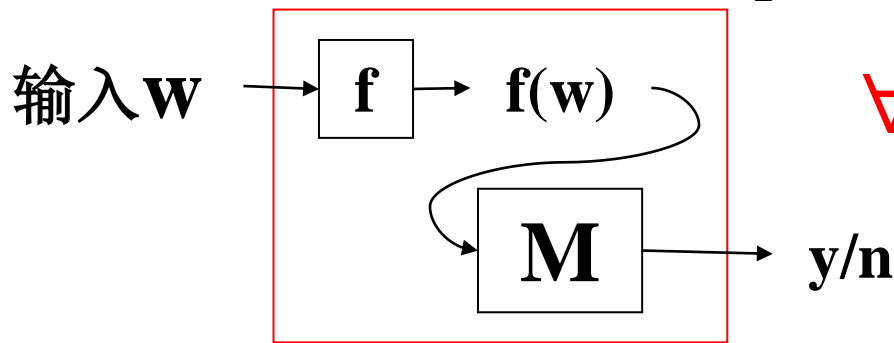
- 定理2: 若B是NPC, 且 $B \in P$, 则 $P=NP$.

证明: $\forall A \in NP, A \leq_p B + B \in P \Rightarrow A \in P$

- 定理3: 若B是NPC, $B \leq_p C$, 且 $C \in NP$, 则C是NPC.

证明: $\forall A \in NP, (A \leq_p B) + (B \leq_p C) \Rightarrow A \leq_p C$

- $3SAT$ 是NPC + $3SAT \leq_p CLIQUE \Rightarrow CLIQUE$ 是NPC



$\forall w \in \Sigma^*, w \in A \iff f(w) \in B.$

利用f和B的判定器
构造A的判定器

Cook-Levin定理的证明步骤(补充)

- 定义:语言B称为NP完全的(NPC),若它满足:

1) $B \in NP$;

2) $\forall A \in NP$, 都有 $A \leq_p B$.

- Cook-Levin定理: SAT是NP完全问题.

证明步骤:

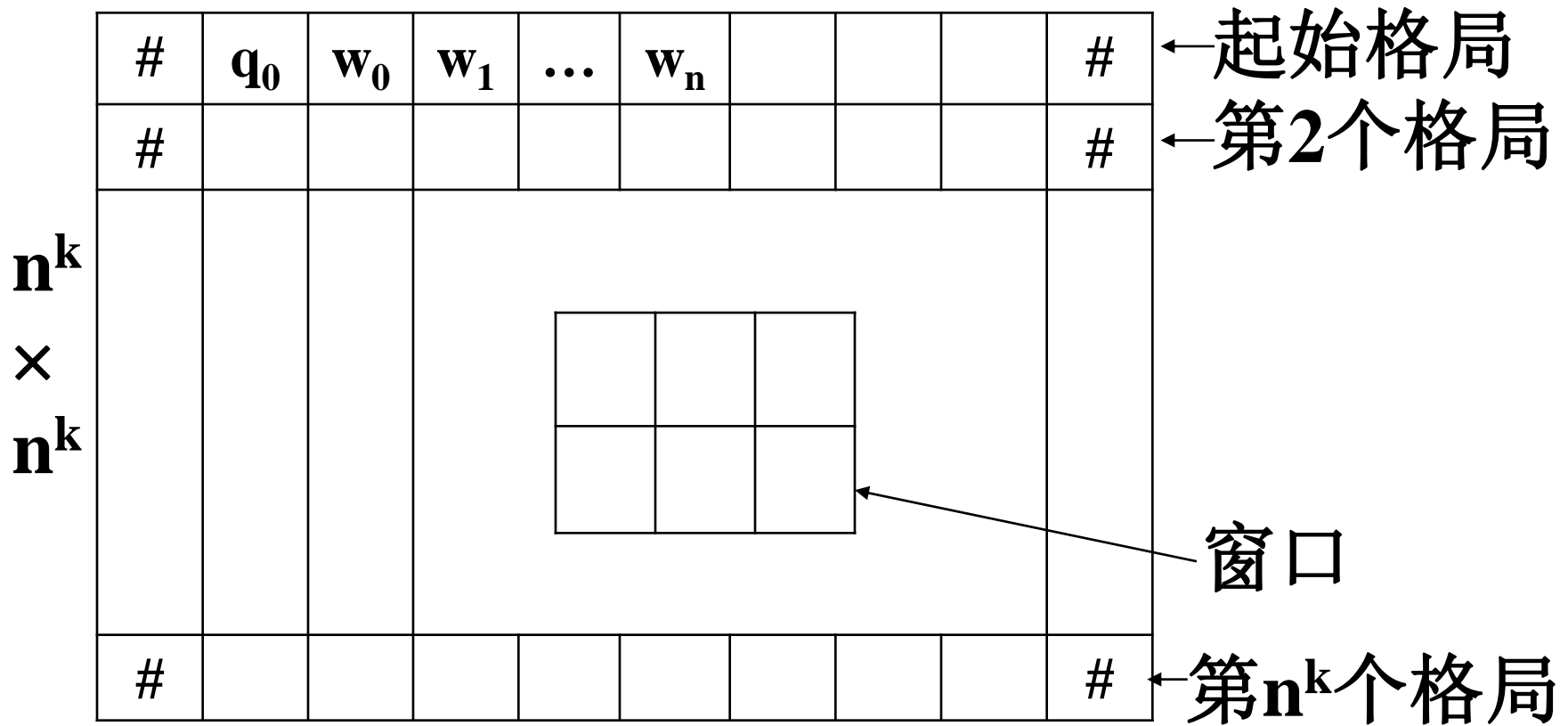
1. $SAT \in NP$ (已证)

2. $\forall A \in NP, A \leq_p SAT$

$\forall A \in \text{NP}$, 都有 $A \leq_p \text{SAT}$ (P170)

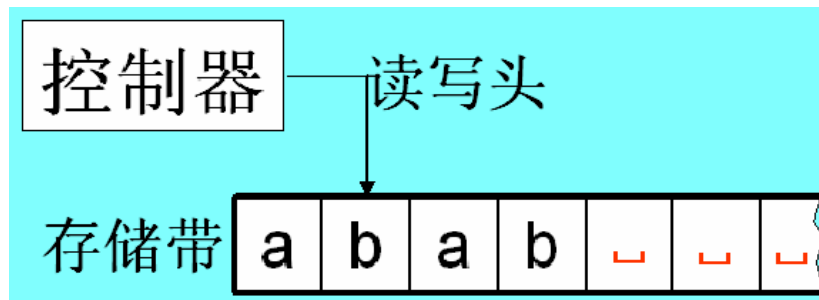
- 思想: 将字符串对应到布尔公式
利用接受的形式定义.
- 过程: 任取 $A \in \text{NP}$, 设 N 是 A 的 n^k 时间 NTM.
 - $\forall w (|w|=n), N$ 接受 w
 - $\Leftrightarrow N$ 对 w 有长度小于 n^k 的接受格局序列
 - \Leftrightarrow 能填好 N 在 w 上的画面 (一个 $n^k \times n^k$ 表格)
 - $\Leftrightarrow \phi = f(w)$ 可满足 ($|\langle \phi \rangle| = O(n^{2k})$)
- 结论: SAT 是 NP 完全的

N接受 $w \Leftrightarrow$ 能填好N在 w 上的表(P170)



能填好表: 第一行是起始格局
 上一行能产生(或等于)下一行
 表中有接受状态

回忆图灵机(TM)形式化定义(P88)

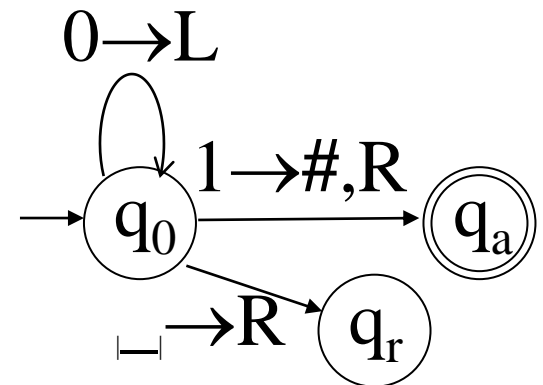


TM是一个7元组 $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

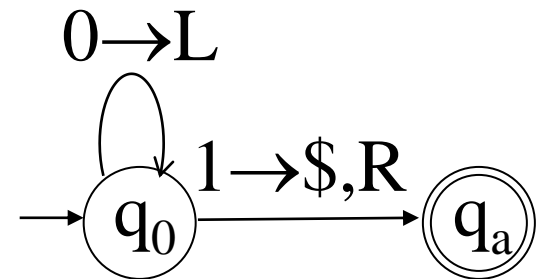
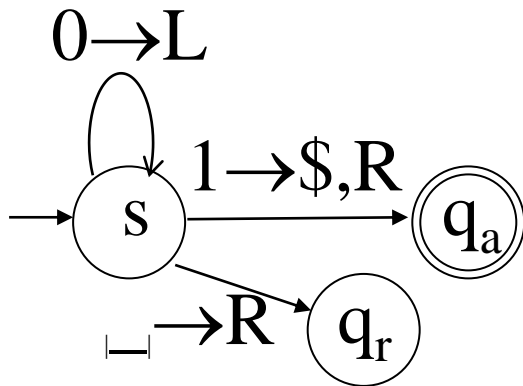
- 1) Q 是状态集.
- 2) Σ 是输入字母表,不包括空白符 \sqcup .
- 3) Γ 是带字母表,其中 $\sqcup \in \Gamma, \Sigma \subset \Gamma$.
- 4) $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ 是转移函数.
- 5) $q_0 \in Q$ 是起始状态. 6) $q_a \in Q$ 是接受状态.
- 7) $q_r \in Q$ 是拒绝状态, $q_a \neq q_r$.

回忆图灵机格局的定义(P88-9)

- 描述图灵机运行的每一步需要如下信息：
控制器的**状态**；存储带上**字符串**；读写头的**位置**.
- 定义：对于图灵机 $M=(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ ，
设 $q \in Q, u, v \in \Gamma^*$ ，则**格局** uqv 表示
 - 1) 当前控制器**状态**为 q ；
 - 2) **存储带**上字符串为 uv (其余为空格)；
 - 3) **读写头**指向 v 的第一个符号.
- 起始格局, 接受格局, 拒绝格局.



格局演化举例(补充)



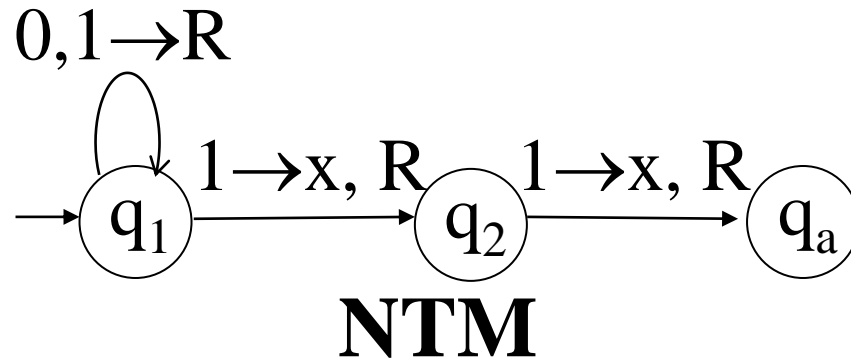
省略拒绝状态

s 0 1	s 1 0	s _ _
s 0 1	\$ q_a 0	_ q_r _
...	接受	拒绝
循环		

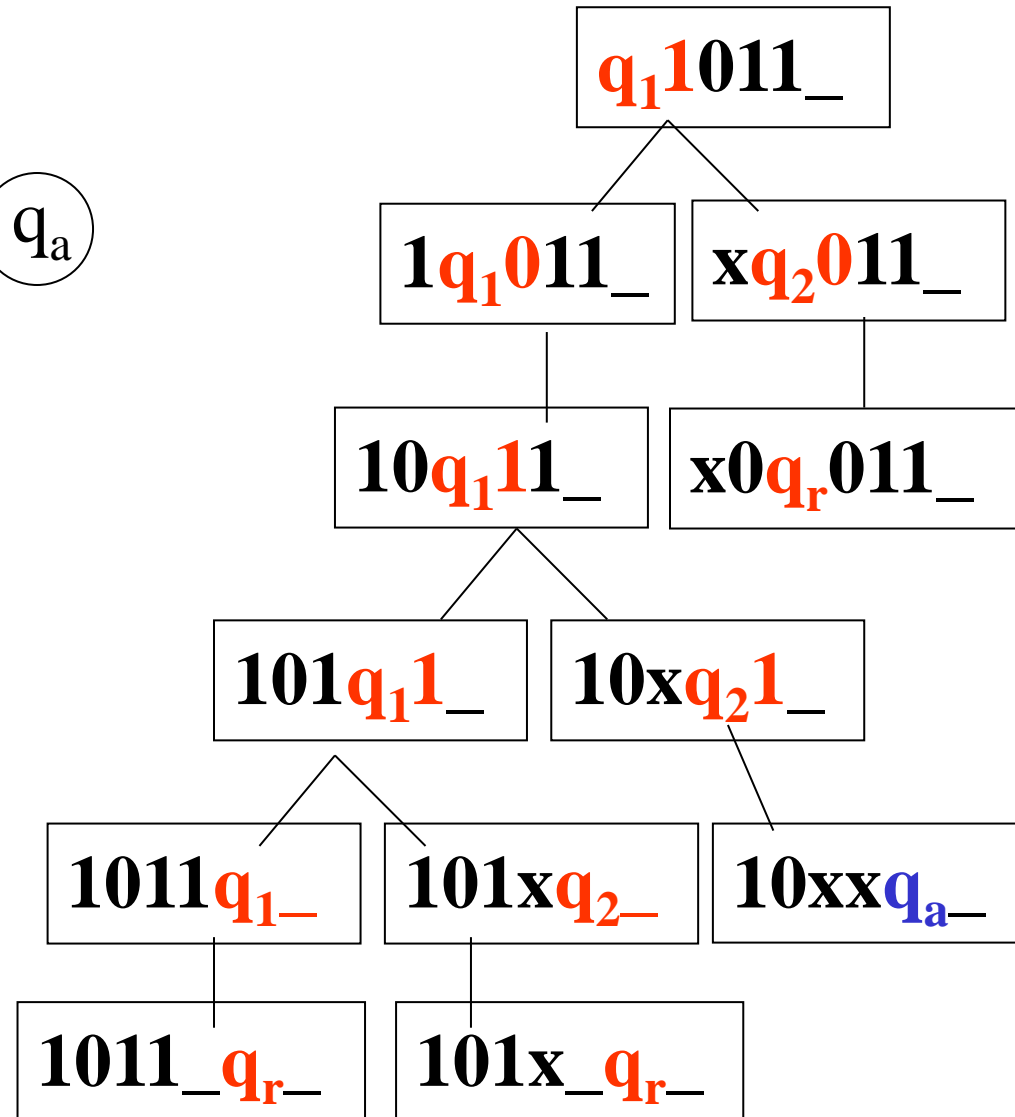
#	s	1	0	_	_	#
#	\$	q _a	0	_	_	#

#	s	1	0	_	_	#
#	\$	q _a	0	_	_	#
#	\$	q _a	0	_	_	#
#	\$	q _a	0	_	_	#

N接受 $w \Leftrightarrow$ 能填好N在 w 上的表(补充)



#	q ₁	1	0	1	1	_	#
#	1	q ₁	0	1	1	_	#
#	1	0	q ₁	1	1	_	#
#	1	0	x	q ₂	1	_	#
#	1	0	x	x	q _a	_	#
#	1	0	x	x	q _a	_	#
#	1	0	x	x	q _a	_	#
#	1	0	x	x	q _a	_	#



构造布尔公式 $\phi=f(w)$ (补充)

- 能填好画面 $\Leftrightarrow \phi=f(w)$ 可满足
- $f(w)=\langle \phi \rangle$, $\phi = \phi_{\text{cell}} \wedge \phi_{\text{start}} \wedge \phi_{\text{move}} \wedge \phi_{\text{accept}}$ •
- 对于任意赋值:
 1. $\phi_{\text{cell}}=1 \Leftrightarrow$ 每格有且只有一个符号;
 2. $\phi_{\text{start}}=1 \Leftrightarrow$ 第一行是起始格局;
 3. $\phi_{\text{accept}}=1 \Leftrightarrow$ 表格中有接受状态;
 4. $\phi_{\text{move}}=1 \Leftrightarrow$ 每行由上一行格局产生.
- $\forall w, w \in A \Leftrightarrow \langle \phi \rangle \in \text{SAT}$ 即 $A \leq_m \text{SAT}$
- 若 $|\langle \phi \rangle|$ 是 $|w|$ 的多项式, 则有 $A \leq_p \text{SAT}$

构造 ϕ_{cell} (P170)

ϕ 的变量: $x_{i,j,s}$, $i,j=1,\dots,n^k$, $s \in Q \cup \Gamma \cup \{\#\}$ //全体符号

$x_{i,j,s}$: 第 i 行第 j 列是否填了符号 s

$$\phi_{\text{cell}} = \bigwedge_{1 \leq i,j \leq n^k} \{ [\bigvee_s x_{i,j,s}] \wedge [\bigwedge_{s \neq t} \overline{(x_{i,j,s} \wedge x_{i,j,t})}] \}$$

$$\bigvee_s x_{i,j,s} = 1 \Leftrightarrow (i,j)\text{格中至少有一个符号}$$

$$\bigwedge_{s \neq t} \overline{(x_{i,j,s} \wedge x_{i,j,t})} = 1 \Leftrightarrow (i,j)\text{格中至多有一个符号}$$

$$\text{例: } (x_{i,j,1} \vee x_{i,j,2} \vee x_{i,j,3}) \wedge \overline{(x_{i,j,1} \wedge x_{i,j,2})} \wedge \overline{(x_{i,j,1} \wedge x_{i,j,3})} \wedge \overline{(x_{i,j,2} \wedge x_{i,j,3})}$$

- 长 $O(n^{2k})$

- $\phi_{\text{cell}} = 1 \Leftrightarrow$ 每格有且只有一个符号;

构造 ϕ_{start} (P171)

$$\phi_{\text{start}} = x_{1,1,\#} \wedge x_{1,2,q_0} \wedge x_{1,3,w_1} \wedge \cdots \wedge x_{1,n^k,\#}$$

- 长 $O(n^k)$
- $\phi_{\text{start}} = 1 \Leftrightarrow$ 第一行是起始格局;

构造 ϕ_{accept} (P171)

$$\phi_{\text{accept}} = \bigvee_{1 \leq i, j \leq n^k} x_{i, j, q_{\text{accept}}}$$

- 长 $O(n^{2k})$
- $\phi_{\text{accept}} = 1 \Leftrightarrow$ 表格中有接受状态

构造 ϕ_{move} (P171)

$$\phi = \phi_{\text{cell}} \wedge \phi_{\text{start}} \wedge \phi_{\text{move}} \wedge \phi_{\text{accept}}.$$

ϕ_{move} 确定表的每行是上一行的合法结果.

只需判断每个 2×3 窗口是否“合法”.

合法窗口(P171)

设 $\delta(q_1, a) = \{(q_1, b, R), \delta(q_1, b) = \{(q_2, c, L), (q_2, a, R)\}$,

合法
窗口

a	q ₁	b
q ₂	a	c

a	q ₁	b
a	a	q ₂

d	a	q ₁
d	a	b

#	b	a
#	b	a

a	b	a
a	b	q ₂

b	b	b
c	b	b

非法
窗口

a	b	a
a	a	a

a	q ₁	b
q ₁	a	a

a	q ₁	b
q ₁	a	q ₁

$$\phi_{\text{move}} = \bigwedge_{1 \leq i, j \leq n^k} \left\{ \bigvee_{\substack{a_1, a_2, \dots, a_6 \\ \text{是合法窗口}}} [x_{i, j-1, a_1} \wedge \dots \wedge x_{i+1, j+1, a_6}] \right\}$$

合法窗口有常数个(P171)

设 $\delta(q_1, a) = \{(q_1, b, R), \delta(q_1, b) = \{(q_2, c, L), (q_2, a, R)\}$,

合法
窗口

a	q ₁	b
q ₂	a	c

#	b	a
#	b	a

a	q ₁	b
a	a	q ₂

a	b	a
a	b	q ₂

d	a	q ₁
d	a	b

b	b	b
c	b	b

$$\phi_{\text{move}} = \bigwedge_{1 \leq i, j \leq n^k} \left\{ \bigvee_{\substack{a_1, a_2, \dots, a_6 \\ \text{是合法窗口}}} [x_{i, j-1, a_1} \wedge \dots \wedge x_{i+1, j+1, a_6}] \right\} O(n^{2k})$$

N的一个转移函数规则对应常数个合法窗口
与N的转移函数无关的合法窗口有常数个

2×2窗口不能正确判断(补充)

设 $(q_2, c, L) \in \delta(q_1, b)$, $(q_3, f, L) \in \delta(q_1, e)$, a 是任意符号

合法
窗口

a	q₁	b
q₂	a	c

a	q₁	e
q₃	a	f

非法
窗口

a	q₁	b
q₃	a	c

$A \leq_P SAT$, SAT是NPC(P172)

$$\phi_{\text{accept}} = \bigvee_{1 \leq i, j \leq n^k} x_{i,j,q_{\text{accept}}}$$

$$\phi_{\text{cell}} = \bigwedge_{1 \leq i, j \leq n^k} \{ [\bigvee_s x_{i,j,s}] \wedge [\bigwedge_{s \neq t} (\overline{x_{i,j,s}} \vee \overline{x_{i,j,t}})] \}$$

$$\phi_{\text{start}} = x_{1,1,\#} \wedge x_{1,2,q_0} \wedge x_{1,3,w_1} \wedge \cdots \wedge x_{1,n^k,\#}$$

$$\phi_{\text{move}} = \bigwedge_{1 \leq i, j \leq n^k} \{ \bigvee_{\substack{a_1, a_2, \dots, a_6 \\ \text{是合法窗口}}} [x_{i,j-1,a_1} \wedge \cdots \wedge x_{i+1,j+1,a_6}] \}$$

$$(1) \ f(\mathbf{w}) = \langle \phi \rangle = \langle \phi_{\text{cell}} \wedge \phi_{\text{start}} \wedge \phi_{\text{move}} \wedge \phi_{\text{accept}} \rangle$$

$$(2) \ w \in A \Leftrightarrow \langle \phi \rangle \in SAT,$$

$$(3) \ \text{令 } |\mathbf{w}| = \mathbf{n}, \text{ 则 } |\langle \phi \rangle| = O(\mathbf{n}^{2k})$$

推论:3SAT是NP完全的(P173)

只需将前面的 ϕ 改造为3cnf公式.

$$\phi = \phi_{\text{cell}} \wedge \phi_{\text{start}} \wedge \phi_{\text{move}} \wedge \phi_{\text{accept}}.$$

$$\phi_{\text{start}} = x_{1,1,\#} \wedge x_{1,2,q_0} \wedge x_{1,3,w_1} \wedge \cdots \wedge x_{1,n^k,\#}$$

$$\phi_{\text{accept}} = \bigvee_{1 \leq i, j \leq n^k} x_{i,j,q_{\text{accept}}}$$

$$\phi_{\text{cell}} = \bigwedge_{1 \leq i, j \leq n^k} \{ [\bigvee_s x_{i,j,s}] \wedge [\bigwedge_{s \neq t} (\overline{x_{i,j,s}} \vee \overline{x_{i,j,t}})] \}$$

降子句长度: 给定赋值T $a_1 \vee a_2 \vee \dots \vee a_k = 1 \iff$

$\exists z$ 赋值, 在T下 $(a_1 \vee a_2 \vee \dots \vee a_{k-2} \vee z) \wedge (\neg z \vee a_{k-1} \vee a_k) = 1$

1个k-文字子句 变为 k-2个3-文字子句

$$|\phi_{\text{accept}}|: n^{2k} \rightarrow 3n^{2k}. \quad |\phi_{\text{cell}}|: (|S| + |S|^2)n^{2k} \rightarrow (3|S| + |S|^2)n^{2k}.$$

ϕ_{move} 的改造(P173)

$$\phi_{\text{move}} = \bigwedge_{1 \leq i, j \leq n^k} \left\{ \bigvee_{\substack{a_1, a_2, \dots, a_6 \\ \text{是合法窗口}}} [x_{i, j-1, a_1} \wedge \dots \wedge x_{i+1, j+1, a_6}] \right\}$$

分配律 $(a \wedge b) \vee c = (a \vee c) \wedge (b \vee c)$

$$(a \wedge b) \vee (c \wedge d) \vee (e \wedge f) = (a \vee c \vee e) \wedge (a \vee c \vee f) \wedge \dots$$

长度由 2×3 变为 3×2^3 .

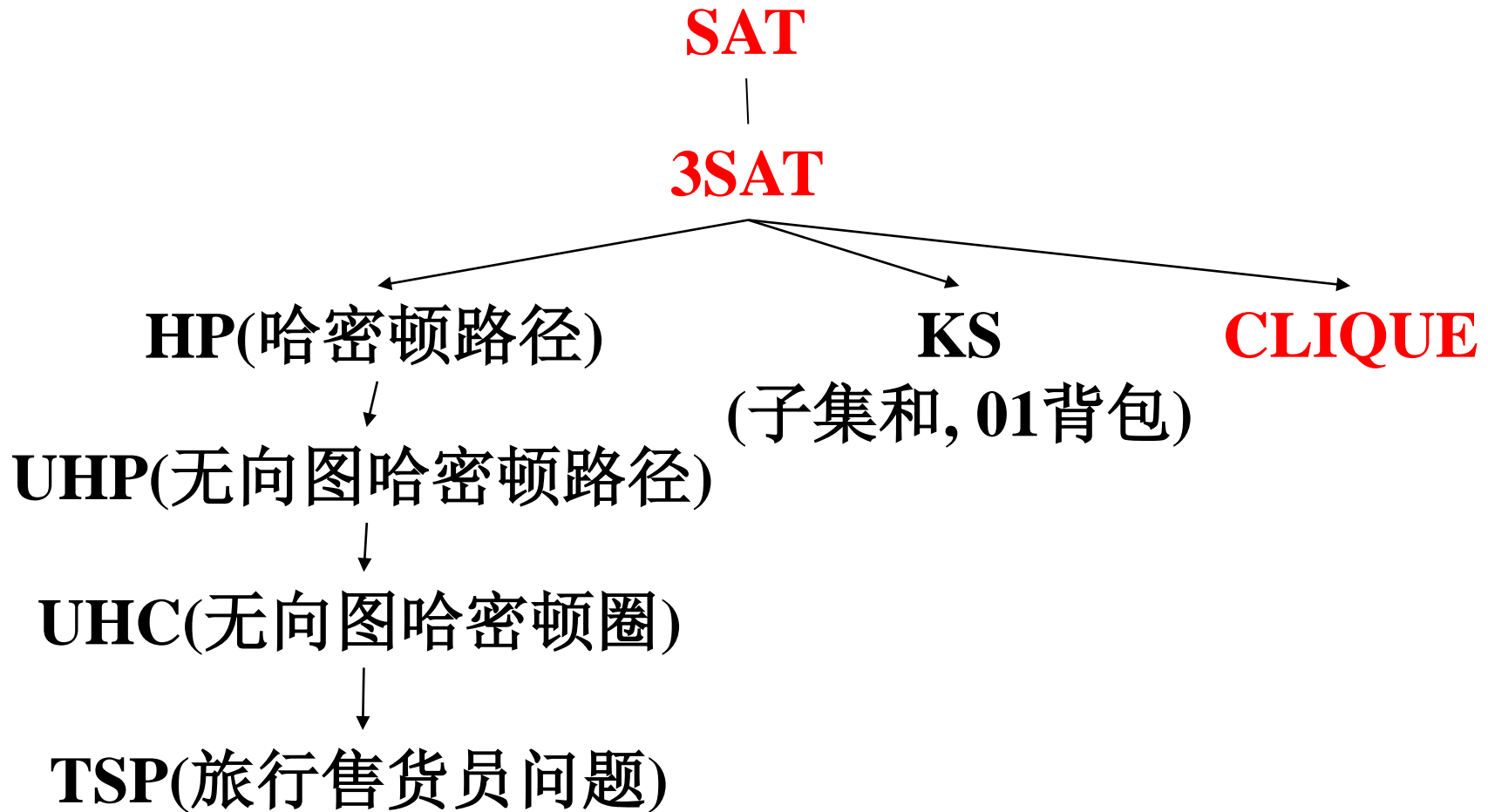
设合法窗口有 M 个, 则 ϕ_{move} 原长度是 $6Mn^{2k}$,

改造为cnf范式后, ϕ_{move} 长度是 $M \times 6^M \times n^{2k}$.

改造为3cnf后, 长度为 $3 \times (M-2) \times 6^M \times n^{2k}$.

所以3SAT是NP完全的.

其它NP完全问题(补充)



HP是NPC($3SAT \leq_p HP$)(P175)

$HP = \{ \langle G, s, t \rangle \mid G \text{ 是有向图, 有从 } s \text{ 到 } t \text{ 的哈密顿路径} \}$

任取3cnf公式 $\phi = (a_1 \vee b_1 \vee d_1) \wedge \dots \wedge (a_k \vee b_k \vee d_k)$,

不妨设有 k 个子句 c_1, \dots, c_k , n 个变量 x_1, \dots, x_n ,

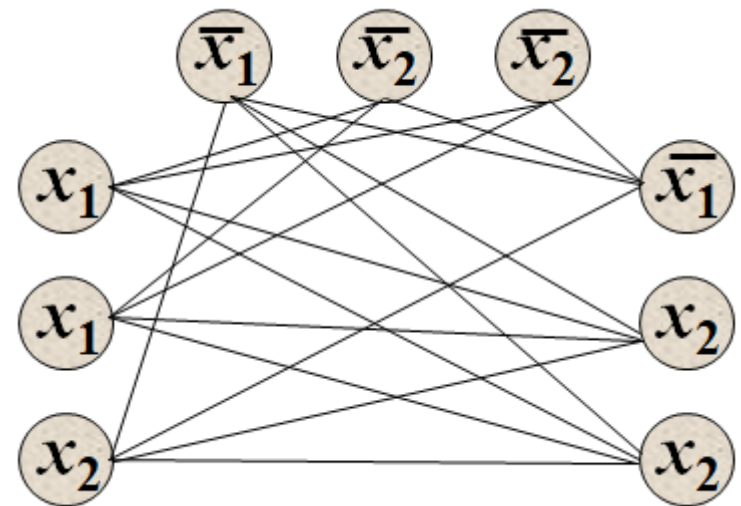
构造 $f(\phi) = \langle G, s, t \rangle$ 使得 ϕ 可满足 $\Leftrightarrow G$ 有从 s 到 t 的HP

一般由3cnf公式构造图有

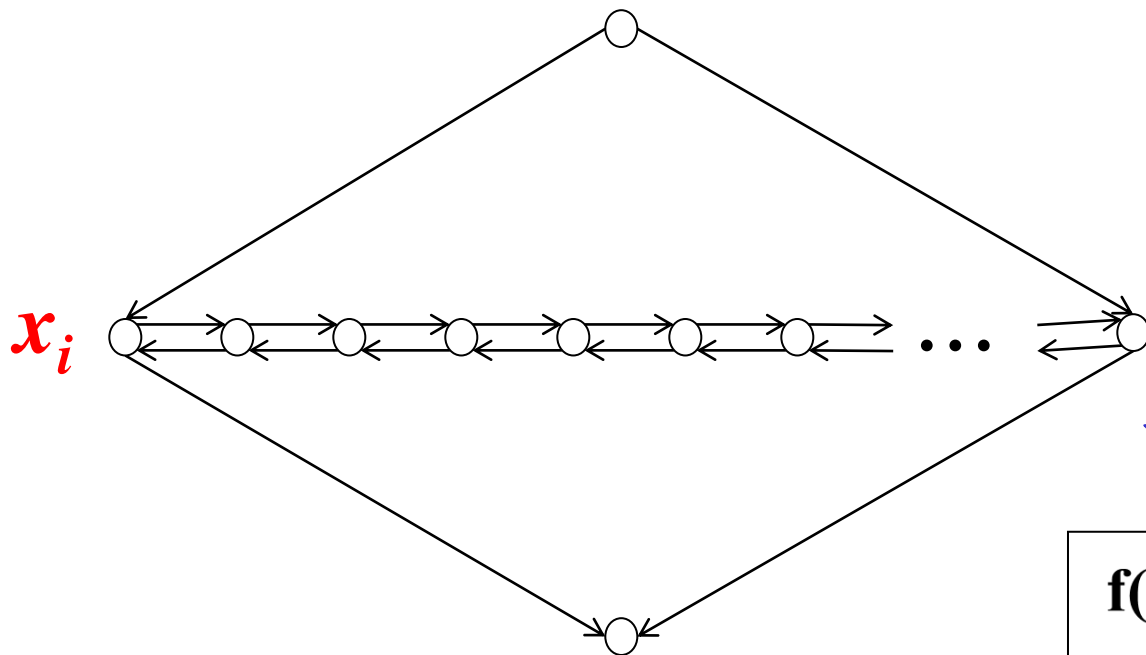
变量构件, 子句构件, 联接构件

如右图3SAT到CLIQUE归约中

有子句构件和联接构件



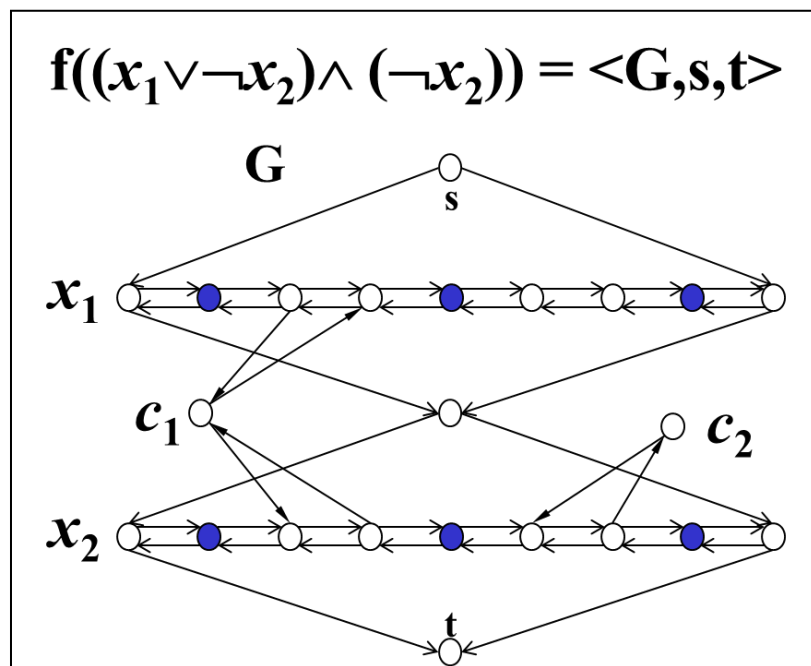
变量构件和子句构件(P175)



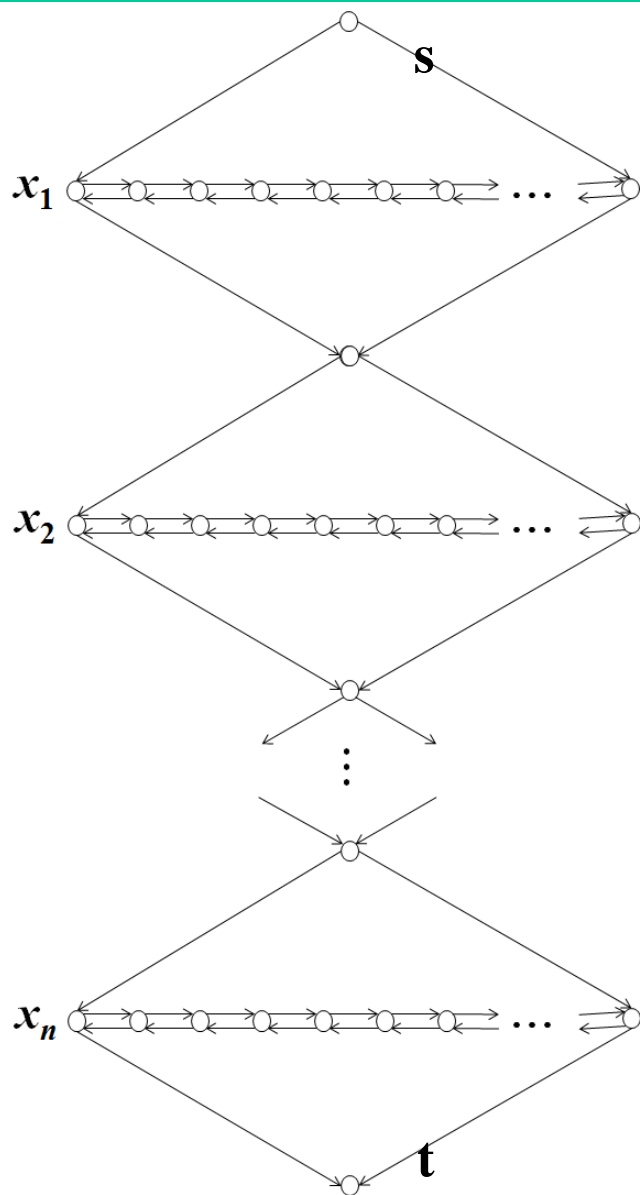
变量 x_i 表示为一个钻石结构

$\bigcirc \ c_j$

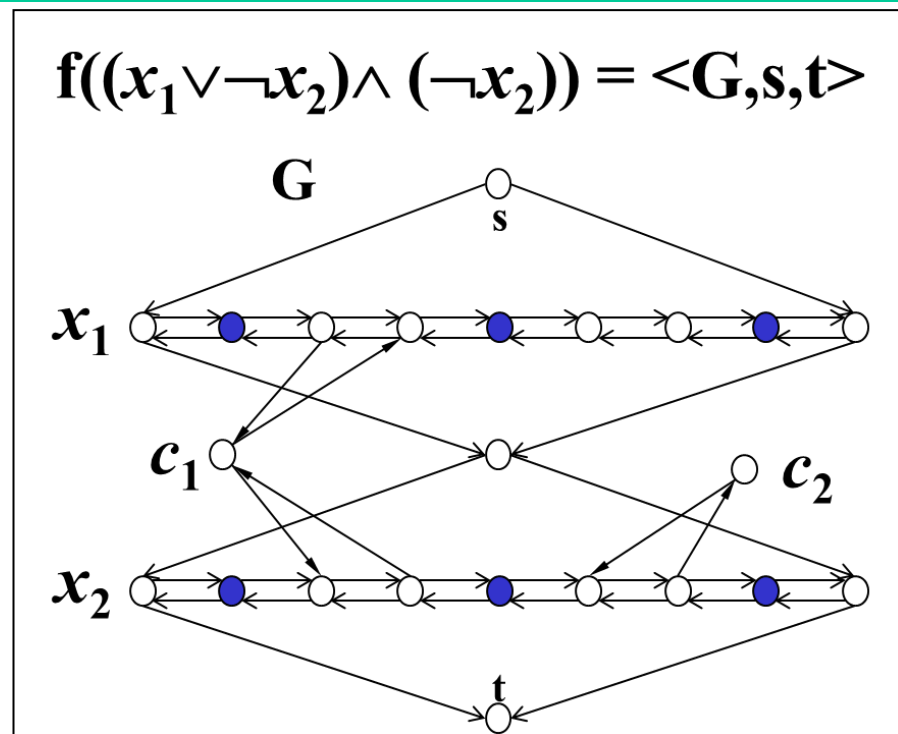
子句 c_j 表示为一个节点



图G的总体结构



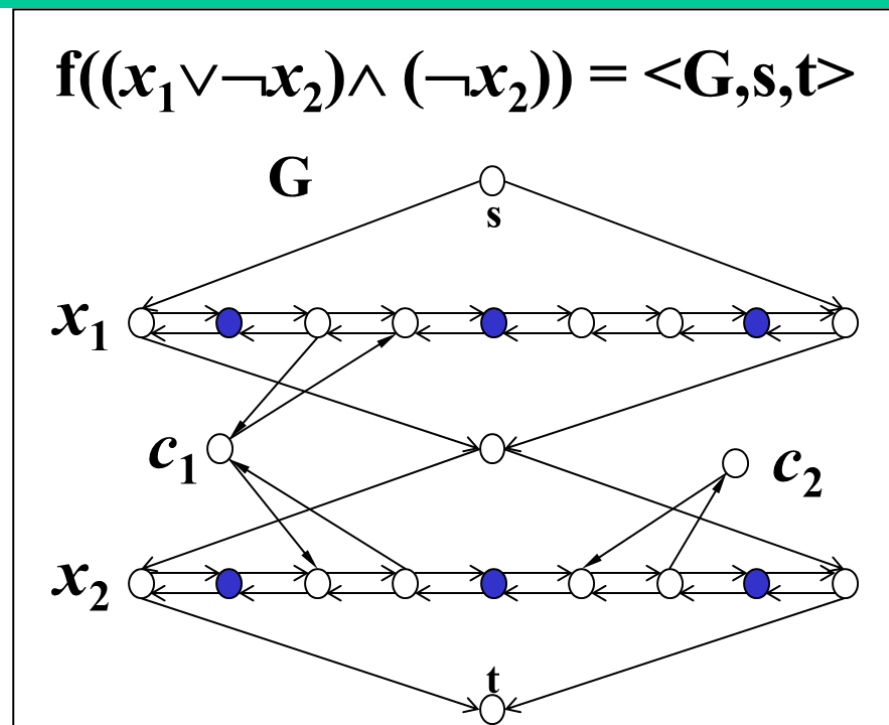
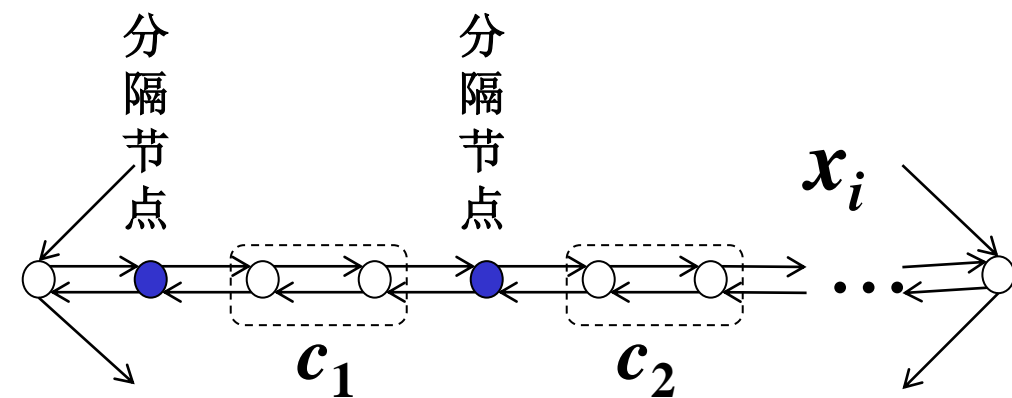
- c_1
- c_2
- ⋮
- c_k



对应
 n 个变量 x_1, \dots, x_n ,
 k 个子句 c_1, \dots, c_k ,
 起点 s , 终点 t

这个图有哪些
 哈密顿路径?

钻石构件中的水平节点



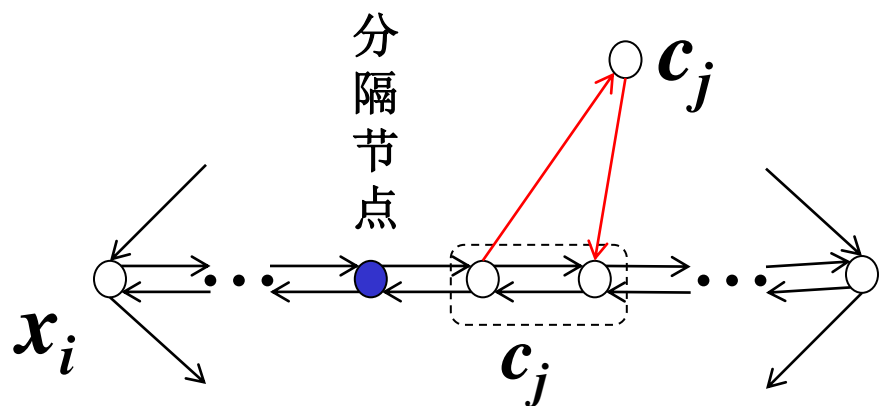
n : 变量个数, k : 子句个数

水平行除两端的两个节点外有 $3k+1$ 个节点

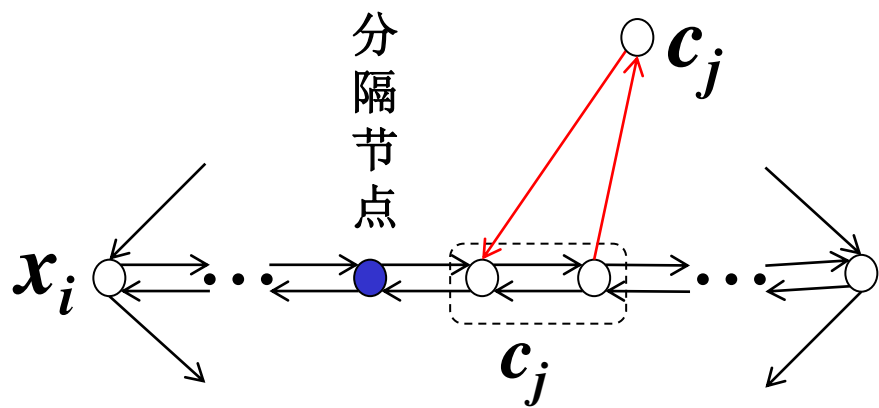
每个子句对应一对节点(共 $2k$ 个)

用分隔节点隔开($k+1$ 个)

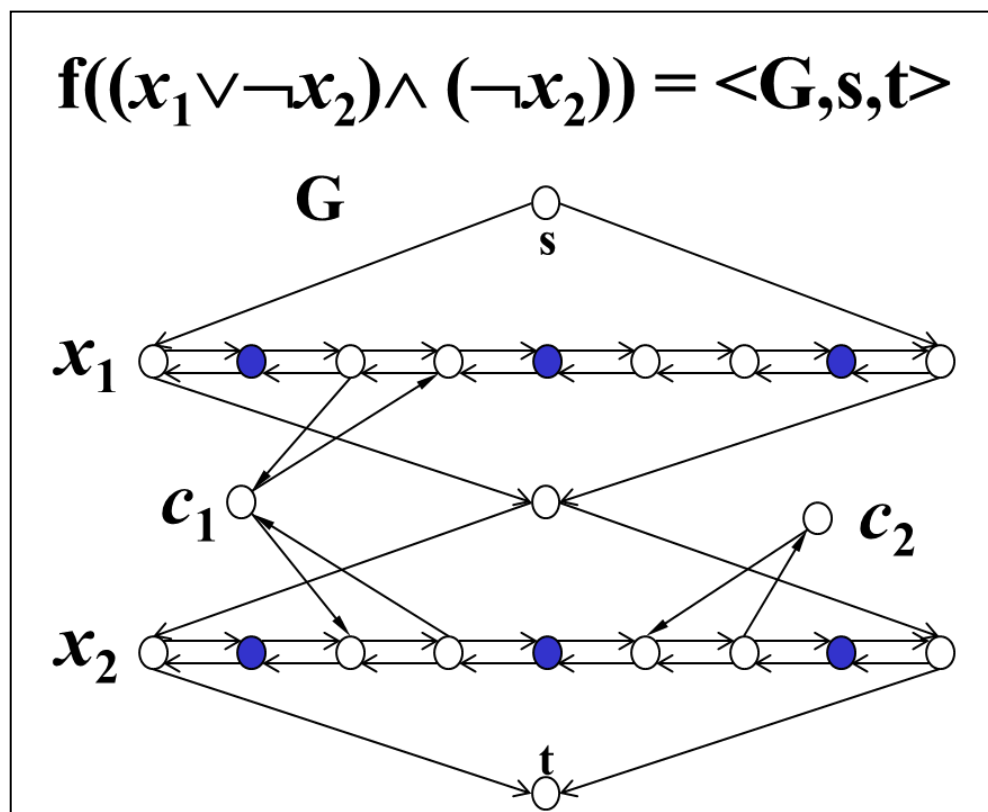
变量与子句构件的连接



当子句 c_j 含有文字 x_i 时添加的边
左-右式路径可以通过



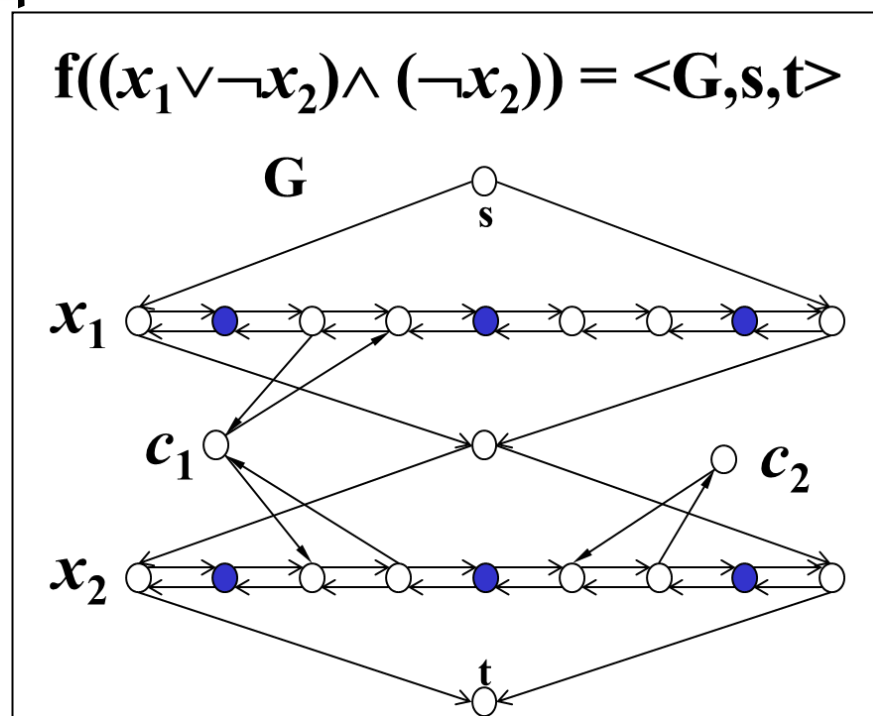
当子句 c_j 含有文字 $\neg x_i$ 时添加的边
右-左式路径可以通过



可满足赋值对应正规路径

ϕ 可满足 \Rightarrow G 有如下从 s 到 t 哈密顿路径

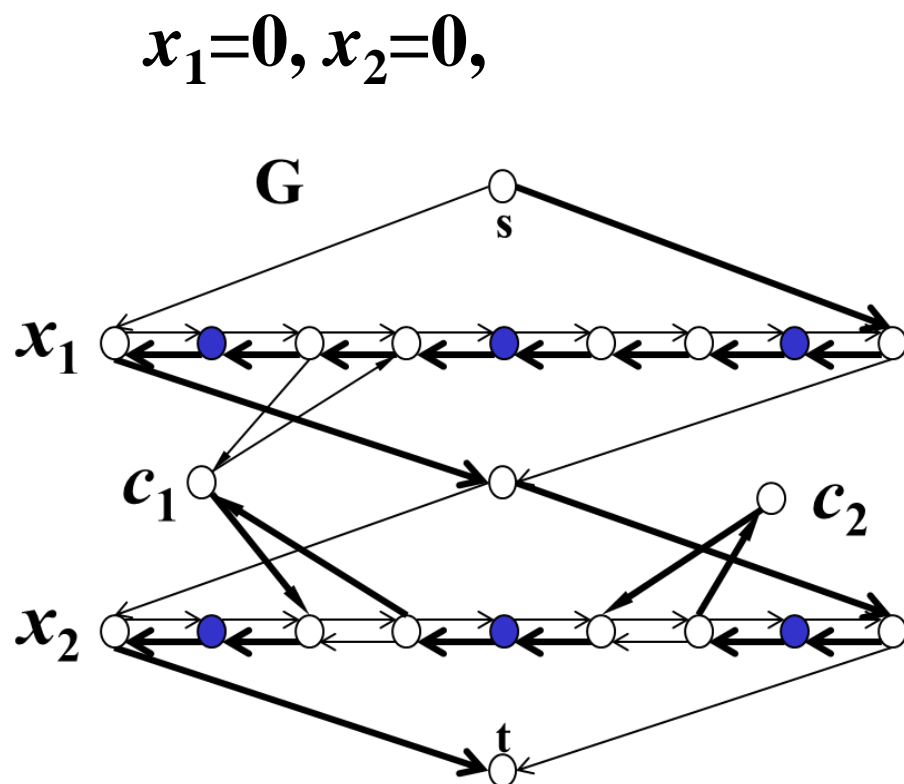
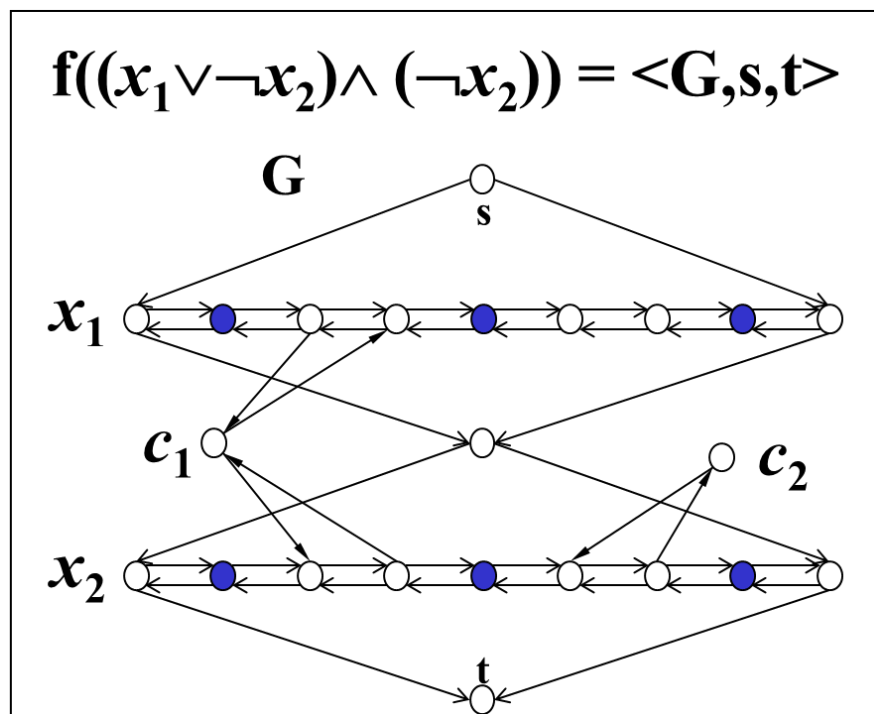
- 从上至下
- 赋值1的变量左-右式通过钻石
- 赋值0的变量右-左式通过钻石
- c_j 选一真文字经过一次
- 称这种路径为正规路径



可满足赋值对应正规路径

ϕ 可满足 \Rightarrow G 有如下从 s 到 t 哈密顿路径

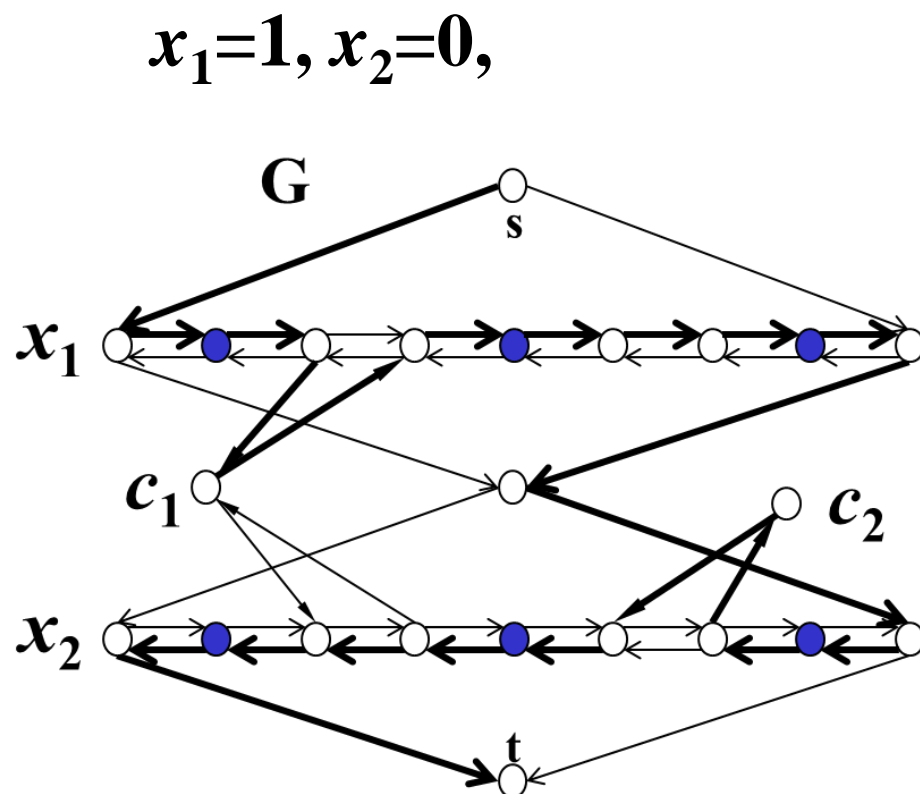
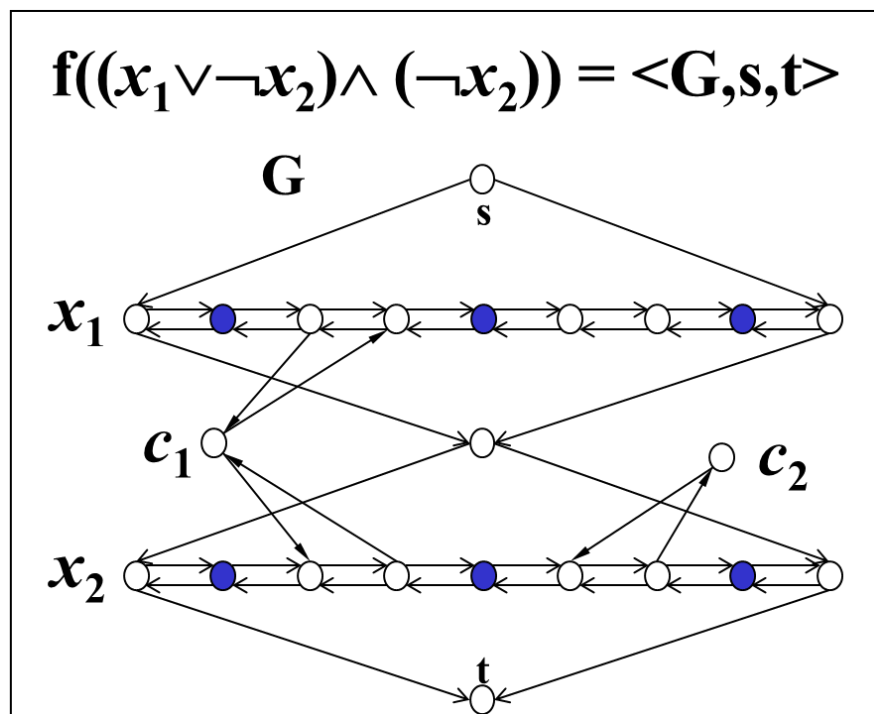
- 从上至下
- 赋值1的变量左-右式通过钻石
- 赋值0的变量右-左式通过钻石
- c_j 选一真文字经过一次
- 称这种路径为正规路径



可满足赋值对应正规路径

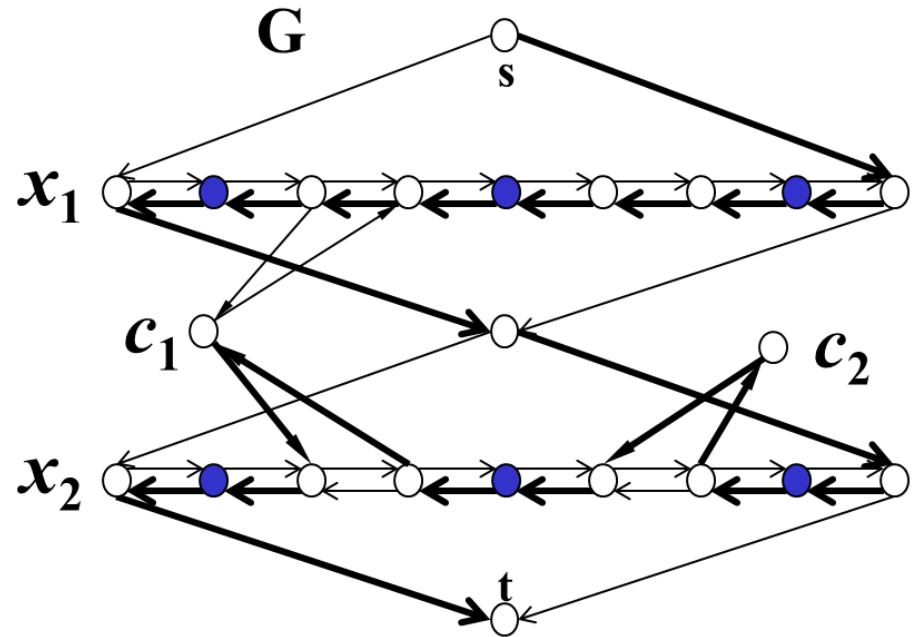
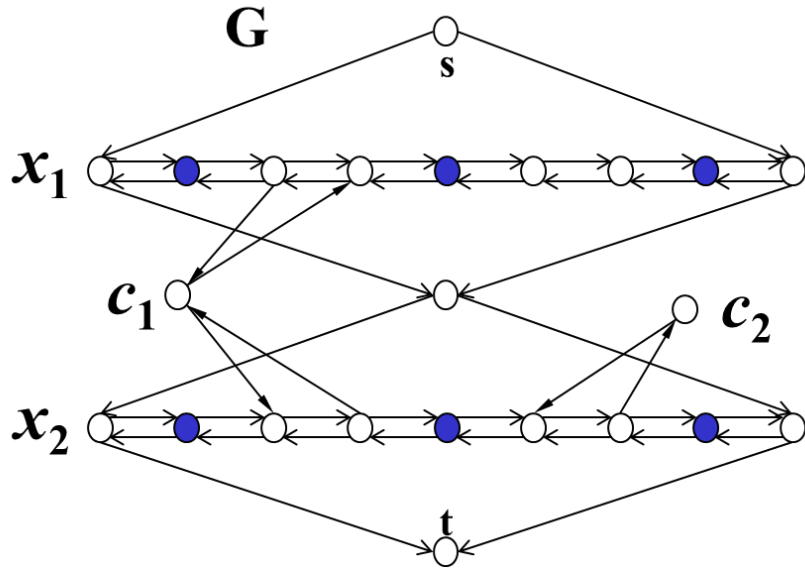
ϕ 可满足 \Rightarrow G 有如下从 s 到 t 哈密顿路径

- 从上至下
- 赋值1的变量左-右式通过钻石
- 赋值0的变量右-左式通过钻石
- c_j 选一真文字经过一次
- 称这种路径为正规路径



正规路径对应可满足赋值

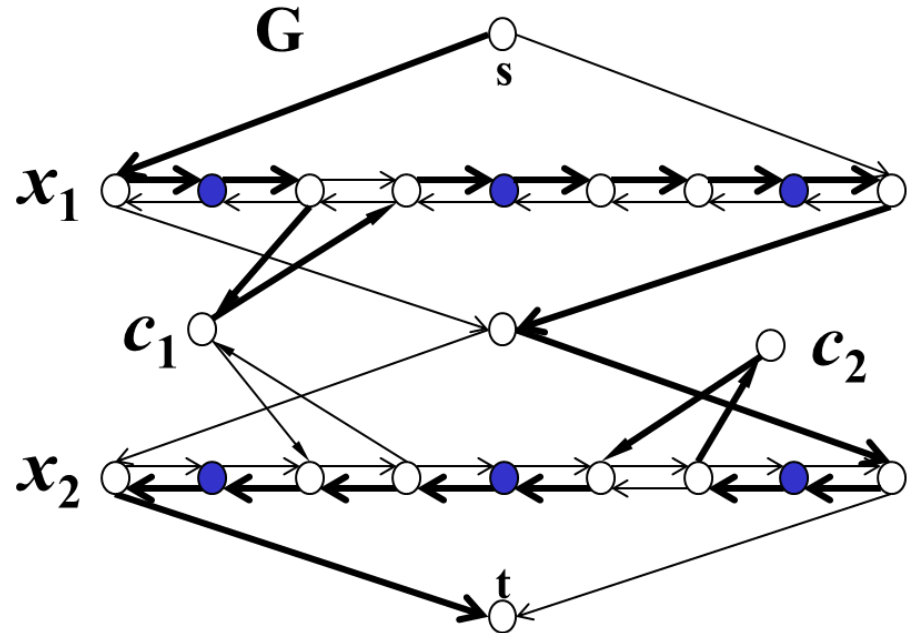
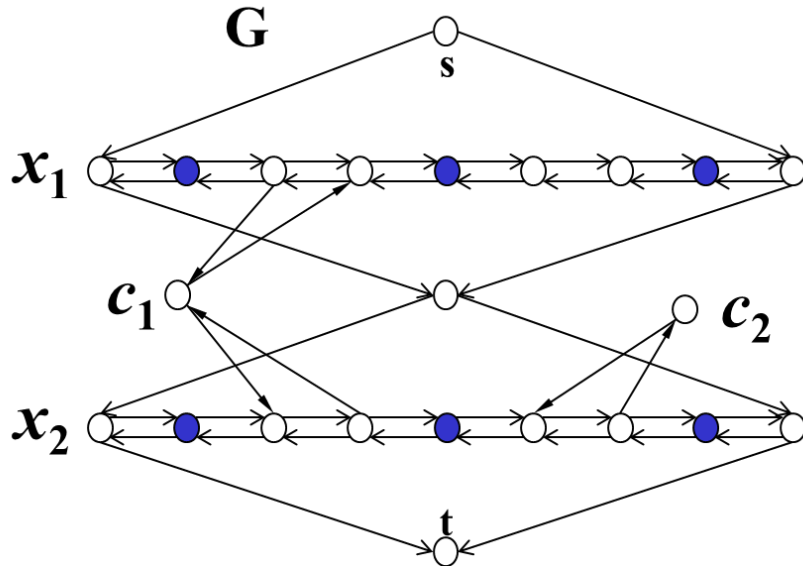
$$f((x_1 \vee \neg x_2) \wedge (\neg x_2)) = \langle G, s, t \rangle$$



- 由左-右 或 右-左 式穿过钻石可确定变量赋值,
 - c_j 被穿过说明在对应变量赋值下 $c_j = 1$,
则公式 ϕ 可满足
- 右边正规路径对应 $x_1 = 0, x_2 = 0$.

正规路径对应可满足赋值

$$f((x_1 \vee \neg x_2) \wedge (\neg x_2)) = \langle G, s, t \rangle$$



- 由左-右 或 右-左 式穿过钻石可确定变量赋值,
 - c_j 被穿过说明在对应变量赋值下 $c_j=1$,
则公式 ϕ 可满足
- 右边正规路径对应 $x_1=1, x_2=0$.

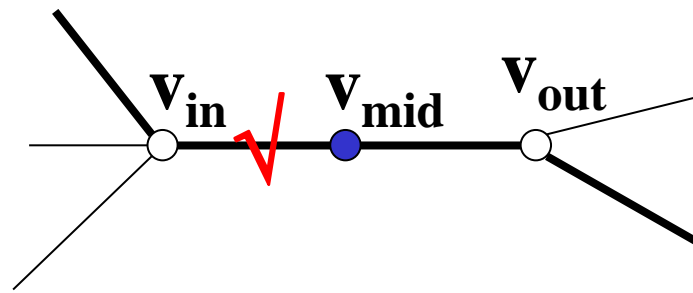
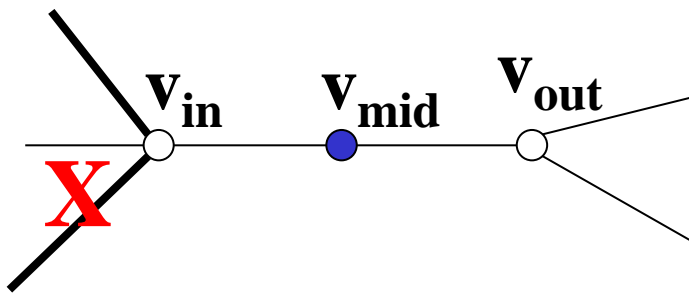
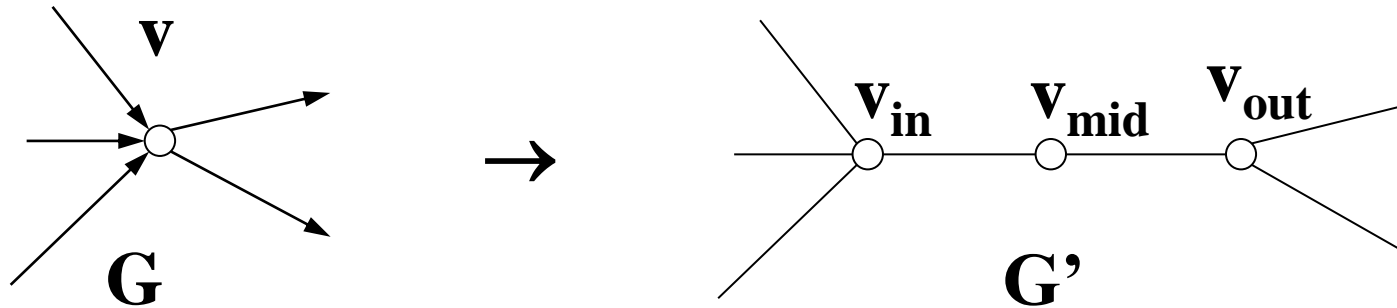
无向图哈密顿路径问题是NPC

$HP = \{ \langle G, s, t \rangle \mid G \text{ 是有从 } s \text{ 到 } t \text{ 哈密顿路径的有向图} \}$

$UHP = \{ \langle G, s, t \rangle \mid G \text{ 是有从 } s \text{ 到 } t \text{ 哈密顿路径的无向图} \}$

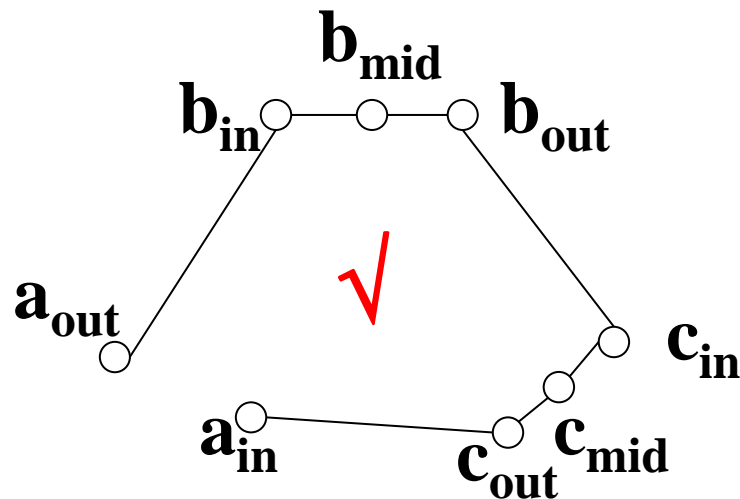
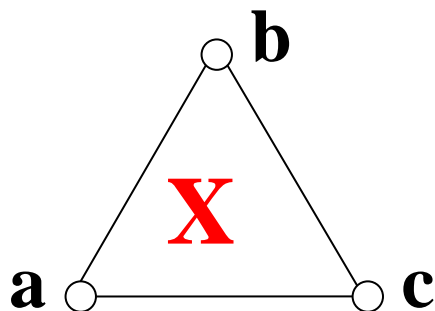
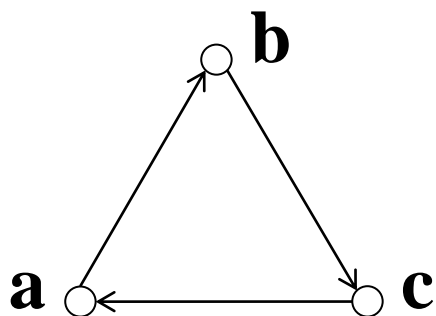
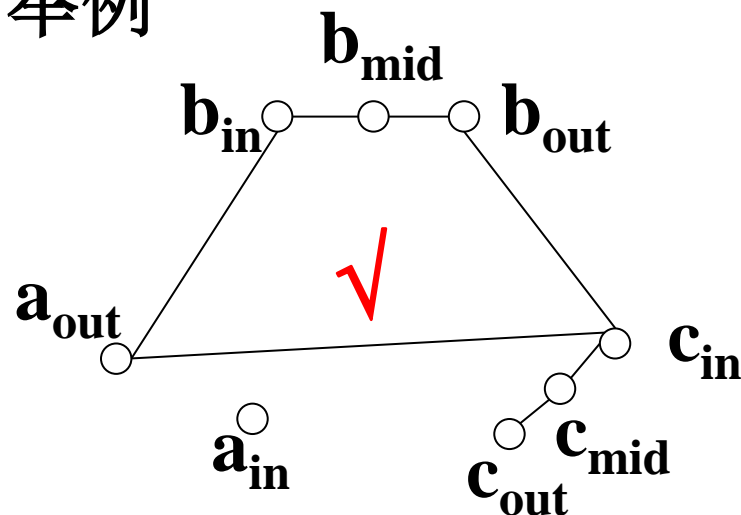
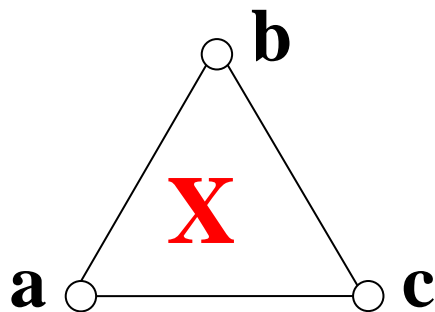
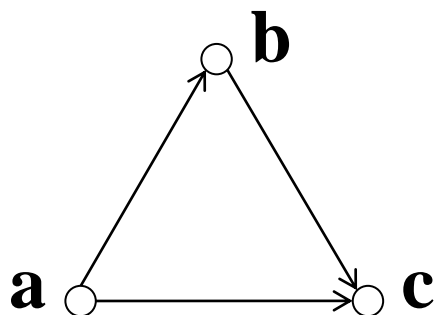
证明: $HP \leq_p UHP$, 映射归约如下 $\langle G, s, t \rangle \rightarrow \langle G', s_{out}, t_{in} \rangle$

s 对应 s_{out} , t 对应 t_{in} , 其它每个节点 v 对应 v_{in}, v_{mid}, v_{out} ,



HP_p ≤ UHP

映射归约 $\langle G, a, a \rangle \rightarrow \langle G', a_{out}, a_{in} \rangle$ 举例



UHC是NP完全的(补充)

$UHC = \{ \langle G \rangle \mid G \text{ 是有哈密顿回路的无向图} \}$

(1) $UHC \in NP$

构造多项式时间内判定UHC的非确定图灵机:

N=“对于输入 $\langle G \rangle$, G是无向图,

1)非确定地选择G所有节点的一个排列 v_1, v_2, \dots, v_n .

2)若 $(v_1, v_2, \dots, v_n, v_1)$ 是G的路径, 则接受; 否则拒绝.”

(2) $UHP \leq_p UHC$

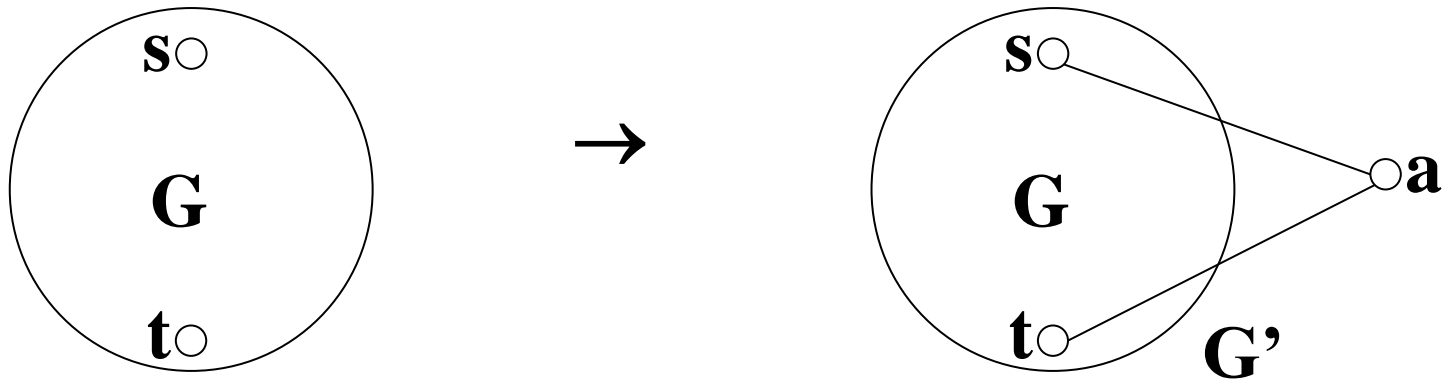
由(1), (2)和UHP是NP完全的, 得UHC是NP完全的

$\text{UHP} \leq_p \text{UHC}$

$\text{UHP} = \{ \langle G, s, t \rangle \mid G \text{ 是有从 } s \text{ 到 } t \text{ 哈密顿路径的无向图} \}$

$\text{UHC} = \{ \langle G \rangle \mid G \text{ 是有哈密顿回路的无向图} \}$

证明: 映射归约如下 $\langle G, s, t \rangle \rightarrow \langle G' \rangle$



$\langle G, s, t \rangle \rightarrow \langle G' \rangle$ 增加一个节点两条边, 多项式时间

G 有从 s 到 t 的哈密顿路径 $\Leftrightarrow G'$ 有哈密顿回路

TSP是NP完全的(补充)

$TSP = \{ \langle G, s, w, b \rangle \mid \text{无向图 } G \text{ 有从 } s \text{ 出发回到 } s, \text{ 权和 } \leq b \text{ 的哈密顿回路} \}$ //将TSP修改成决定性问题

(1) $TSP \in NP$. 构造多项式时间内判定TSP的NTM:

$N =$ “对于输入 $\langle G, s, w, b \rangle$, G 是无向图, s 是节点, w 是权, $b \geq 0$,

1) 非确定地选择 G 所有节点的排列 s, v_2, \dots, v_n .

2) 若 (s, v_2, \dots, v_n, s) 是 G 的路径, 且路径权和 $\leq b$, 则接受;

3) 否则拒绝.”

(2) $UHC \leq_p TSP$

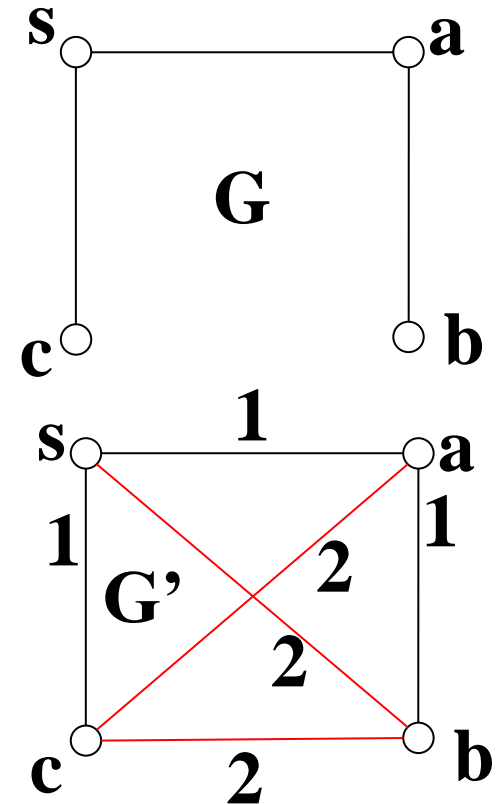
由(1), (2)和UHC是NP完全的, 得TSP是NP完全的

$\text{UHC} \leq_p \text{TSP}$

$\text{UHC} = \{ \langle G \rangle \mid G \text{ 是有哈密顿回路(HC)的无向图} \}$

$\text{TSP} = \{ \langle G, s, w, b \rangle \mid G \text{ 有 } s \text{ 出发费用} \leq b \text{ 的哈密顿回路} \}$

- 设 $G=(V,E)$, $s \in V = \{v_1, \dots, v_n\}$ // n 个节点
- 令 $G'=(V, V \times V)$, $f(\langle G \rangle) = \langle G', s, w, n \rangle$,
- 定义权 w :
$$w[v_i, v_j] = \begin{cases} 0 & \text{若 } i = j \\ 1 & \text{若 } (v_i, v_j) \in E \\ 2 & \text{其它} \end{cases}$$
- $f(\langle G \rangle)$ 增加边数 $\leq n^2$, 多项式时间可计算
- G 有 HC $\Rightarrow G'$ 有 s 出发费用 $\leq n$ 的 HC
- G' 有 s 出发费用 $\leq n$ 的 HC
 \Rightarrow 该回路上的边都在 G 中 $\Rightarrow G$ 有 HC



0-1背包(knapsack)问题是NPC

[S]中称为子集和问题.

$KS = \{ \langle A, t \rangle \mid t \text{ 等于 } A \text{ 中一些数的和} \}$

- $KS \in NP$
- $3SAT \leq_p KS$

设 ϕ 是3cnf公式, 构造 $f(\langle \phi \rangle) = \langle A, t \rangle$

设 ϕ 有 n 个变量 x_1, \dots, x_n , k 个子句 c_1, \dots, c_k ,

构造数集 $A = \{ y_1, \dots, y_n, z_1, \dots, z_n, g_1, \dots, g_k, h_1, \dots, h_k \}$ 和数 t

- 所有数十进制表示, 根据 ϕ 构造每个数的高 n 位和低 k 位
- A 中数每位是0或1; t 的低 k 位都是3, 高 n 位都是1.

$y_1, \dots, y_n, z_1, \dots, z_n, g_1, \dots, g_k, h_1, \dots, h_k, t$ 的构造

- 所有数十进制表示, 根据 ϕ 构造每个数的高 n 位和低 k 位
- A 中数每位是0或1; t 的低 k 位都是3, 高 n 位都是1.
- 构造见下表. 总位数 $\leq (n+k+1)^2$.

	x_1	x_2	...	x_n	c_1	c_2	...	c_k
y_1 ... y_n	$yx_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$				$yc_{ij} = \begin{cases} 1 & \text{若 } c_j \text{ 中有 } x_i \\ 0 & \text{else} \end{cases}$			
z_1 ... z_n	$zx_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$				$zc_{ij} = \begin{cases} 1 & \text{若 } c_j \text{ 中有 } \neg x_i \\ 0 & \text{else} \end{cases}$			
g_1 ... g_k	0				$gc_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$			
h_1 ... h_k	0				$hc_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$			
t	1	1	...	1	3	3	...	3

- yx 区: 单位阵
- zx 区: 单位阵
- gc 区: 单位阵
- hc 区: 单位阵
- yz 行 c_j 列 ≤ 3 个1

归约举例1

$$f(\langle (x_1 \vee \neg x_2) \wedge (\neg x_2) \rangle) = \langle \{1010, 100, 1000, 111, 10, 1, 10, 1\}, 1133 \rangle$$

	x_1	x_2	...	x_n	c_1	c_2	...	c_k
y_1 ... y_n	$yx_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$				$yc_{ij} = \begin{cases} 1 & \text{若 } c_j \text{ 中有 } x_i \\ 0 & \text{else} \end{cases}$			
z_1 ... z_n	$zx_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$				$zc_{ij} = \begin{cases} 1 & \text{若 } c_j \text{ 中有 } \neg x_i \\ 0 & \text{else} \end{cases}$			
g_1 ... g_k	0				$gc_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$			
h_1 ... h_k	0				$hc_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$			
t	1	1	...	1	3	3	...	3

	x_1	x_2	c_1	c_2
y_1	1	0	1	0
y_2	0	1	0	0
z_1	1	0	0	0
z_2	0	1	1	1
g_1	0	0	1	0
g_2	0	0	0	1
h_1	0	0	1	0
h_2	0	0	0	1
t	1	1	3	3

y_1 行 c_1 列是1, 因为 c_1 含 x_1 ; y_1 行 c_2 列是0, 因为 c_2 不含 x_1 ;
 y_2 行 c_1 列是0, 因为 c_1 不含 x_2 ; y_2 行 c_2 列是0, 因为 c_2 不含 x_2 ;
 z_1 行 c_1 列是0, 因为 c_1 不含 $\neg x_1$; z_1 行 c_2 列是0, 因为 c_2 不含 $\neg x_1$;
 z_2 行 c_1 列是1, 因为 c_1 含 $\neg x_2$; z_2 行 c_2 列是1, 因为 c_2 含 $\neg x_2$.

归约举例2

$$f (\langle (x_1 \vee \neg x_2) \rangle) = \langle \{101, 10, 100, 11, 1, 1\}, 113 \rangle$$

	x_1	x_2	...	x_n	c_1	c_2	...	c_k
y_1 ... y_n	$yx_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$				$yc_{ij} = \begin{cases} 1 & \text{若 } c_j \text{ 中有 } x_i \\ 0 & \text{else} \end{cases}$			
z_1 ... z_n	$zx_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$				$zc_{ij} = \begin{cases} 1 & \text{若 } c_j \text{ 中有 } \neg x_i \\ 0 & \text{else} \end{cases}$			
g_1 ... g_k	0				$gc_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$			
h_1 ... h_k	0				$hc_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$			
t	1	1	...	1	3	3	...	3

	x_1	x_2	c_1
y_1	1	0	1
y_2	0	1	0
z_1	1	0	0
z_2	0	1	1
g_1	0	0	1
h_1	0	0	1
t	1	1	3

ϕ 可满足 $\Leftrightarrow f(\langle\phi\rangle) \in \text{KS}(\text{knapsack})$

$$f(\langle(x_1 \vee \neg x_2) \wedge (\neg x_2)\rangle) = \langle\{1010, 100, 1000, 111, 10, 1, 10, 1\}, 1133\rangle$$

	x_1	x_2	c_1	c_2
y_1	1	0	1	0
y_2	0	1	0	0
z_1	1	0	0	0
z_2	0	1	1	1
g_1	0	0	1	0
g_2	0	0	0	1
h_1	0	0	1	0
h_2	0	0	0	1
t	1	1	3	3

- 取赋值 $x_1=0, x_2=0$, (可满足)
对应选 z_1, z_2 ,
添 g_1, g_2, h_1, h_2 , 得和 t
- 取赋值 $x_1=1, x_2=0$, (可满足)
对应选 y_1, z_2 ,
添 g_2, h_1, h_2 , 得和 t
- 取赋值 $x_1=0, x_2=1$, (不可满足)
对应选 z_1, y_2 , 得不到 t
- 取赋值 $x_1=1, x_2=1$, (不可满足)
对应选 y_1, y_2 , 得不到 t

ϕ 可满足 $\Rightarrow f(<\phi>) \in KS$

$$f(<\phi>) = <A, t>,$$

$$A = \{ y_1, \dots, y_n, z_1, \dots, z_n, \\ g_1, \dots, g_k, h_1, \dots, h_k \}$$

	x_1	x_2	...	x_n	c_1	c_2	...	c_k
y_1	$yx_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$				$yc_{ij} = \begin{cases} 1 & \text{若 } c_j \text{ 中有 } x_i \\ 0 & \text{else} \end{cases}$			
...								
y_n								
z_1	$zx_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$				$zc_{ij} = \begin{cases} 1 & \text{若 } c_j \text{ 中有 } \neg x_i \\ 0 & \text{else} \end{cases}$			
...								
z_n								
g_1	0				$gc_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$			
...								
g_k								
h_1	0				$hc_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$			
...								
h_k								
t	1	1	...	1	3	3	...	3

若 ϕ 有满足赋值($<\phi> \in 3SAT$)

则对每个 x_i ,

若 $x_i=1$, 则选数 y_i ;

若 $x_i=0$, 则选数 z_i .

第 x_i 列的和是1.

对每个 c_j ,

已选数 c_j 列和 $\geq 1, \leq 3$

若=1, 则选 g_j, h_j ;

若=2, 则选 g_j ;

若=3, 则不用选

已选数第 c_j 列的和是3

即可选出子集 和 = t

即 $f(<\phi>) \in KS$

ϕ 可满足 $\Leftarrow f(<\phi>) \in KS$

$$f(<\phi>) = <A, t>,$$

$$A = \{ y_1, \dots, y_n, z_1, \dots, z_n, \\ g_1, \dots, g_k, h_1, \dots, h_k \}$$

	x_1	x_2	...	x_n	c_1	c_2	...	c_k
y_1 ... y_n	$yx_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$				$yc_{ij} = \begin{cases} 1 & \text{若 } c_j \text{ 中有 } x_i \\ 0 & \text{else} \end{cases}$			
z_1 ... z_n	$zx_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$				$zc_{ij} = \begin{cases} 1 & \text{若 } c_j \text{ 中有 } \neg x_i \\ 0 & \text{else} \end{cases}$			
g_1 ... g_k	0				$gc_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$			
h_1 ... h_k	0				$hc_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$			
t	1	1	...	1	3	3	...	3

若 $f(<\phi>) \in KS$

即存在子集和 = t

则子集中对每个*i*,

y_i, z_i 只有1个

若有 y_i , 则令 $x_i=1$;

若有 z_i , 则令 $x_i=0$.

子集中对每个*j*,

第 c_j 列的和是3

gh 行 c_j 列和 ≤ 2

yz 行 c_j 列和 $\geq 1, \leq 3$

子句 c_j 在当前赋值下=1

即 ϕ 有满足赋值

计算理论第7章作业

7.22 令 $\text{HALF-CLIQUE} = \{ \langle G \rangle \mid G \text{ 是无向图, 包含结点数至少为 } m/2 \text{ 的完全子图, } m \text{ 是 } G \text{ 的结点数} \}$ 。证明 HALF-CLIQUE 是 NP 完全的。

说明：书上的答案只是要点，考试时需要给出完整的答案。

证明：

(1) $\text{HALF-CLIQUE} \in \text{NP}$

构造如下非确定图灵机

$N =$ “对于输入 $\langle G \rangle$, G 是无向图, 有 m 个顶点

(a) 非确定地产生一个 $m/2$ 个顶点的子集

(b) 若这个子集中的任意两个顶点之间都有边相连, 则接受; 否则, 拒绝”。

因为 N 的语言是 HALF-CLIQUE , 且 N 是在多项式时间运行, 所以 $\text{HALF-CLIQUE} \in \text{NP}$ 。

计算理论第7章作业

(2) 证明CLIQUE可以多项式时间映射归约到HALF-CLIQUE.

对任意 $\langle G, k \rangle$, 其中 G 是一个无向图, k 是一个正整数。构造函数 $f(\langle G, k \rangle) = G'$ 。

设 G 有 m 个顶点。按如下方式构造 G' :

若 $k = m/2$, 则 $G = G'$;

若 $k > m/2$, 则在 G 中增加 $2k - m$ 个新顶点, 这些新顶点都是孤立点, 得到 G' ;

若 $k < m/2$, 则增加 $m - 2k$ 个新顶点, 这些新顶点之间两两都有边相连, 新顶点与 G 的所有顶点之间也都相连。

首先, f 可在多项式时间内计算完成。

其次证明 f 是CLIQUE到HALF-CLIQUE的映射归约, 即证明 G 有 k 团 $\Leftrightarrow G'$ (设有 m' 个顶点)有 $m'/2$ 个顶点的团:

若 G 有 k 团, 当 $k = m/2$ 时, $G' = G$, $m' = m$, 则 G' 也有 $k = m'/2$ 团; 当 $k > m/2$ 时, $m' = 2k$, G' 中也有 $k = m'/2$ 团; 当 $k < m/2$ 时, $m' = 2m - 2k$, G 中的 k 团加上新添的 $m - 2k$ 个顶点形成 $m - k = m'/2$ 团。

若 G' 有 $m'/2$ 团, 当 $k = m/2$ 时, $G' = G$, $m' = m$, 则 G 也有 $k = m'/2$ 团; 当 $k > m/2$ 时, $m' = 2k$, G 中也有 $k = m'/2$ 团; 当 $k < m/2$ 时, $m' = 2m - 2k$, G' 中的 $m - k$ 团至多有 $m - 2k$ 个新添顶点, 去掉新添顶点至少还有 k 个顶点, 所以 G 中有 k 团。

由(1)和(2), HALF-CLIQUE是NP完全问题。

计算理论总结

计算模型

- 有限自动机 非确定有限自动机 正则表达式
正则语言 泵引理

- 图灵机 图灵可判定语言 图灵可识别语言
可计算理论

- 停机问题非图灵可判定,
- 停机问题的补不是图灵可识别

计算复杂性

- **P, NP, EXP, NP完全**