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

   $\{ 0^k1^k \mid k \geq 0 \}$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

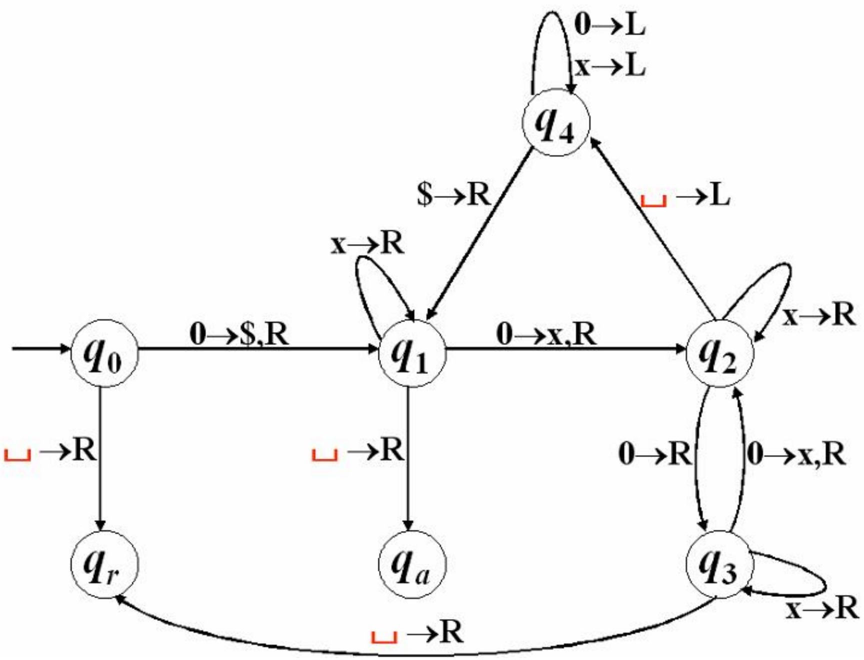
- 时间复杂度定义
- $\{\, 0^k 1^k \mid k \ge 0 \,\}$的时间复杂度分析

时间复杂性(P153)

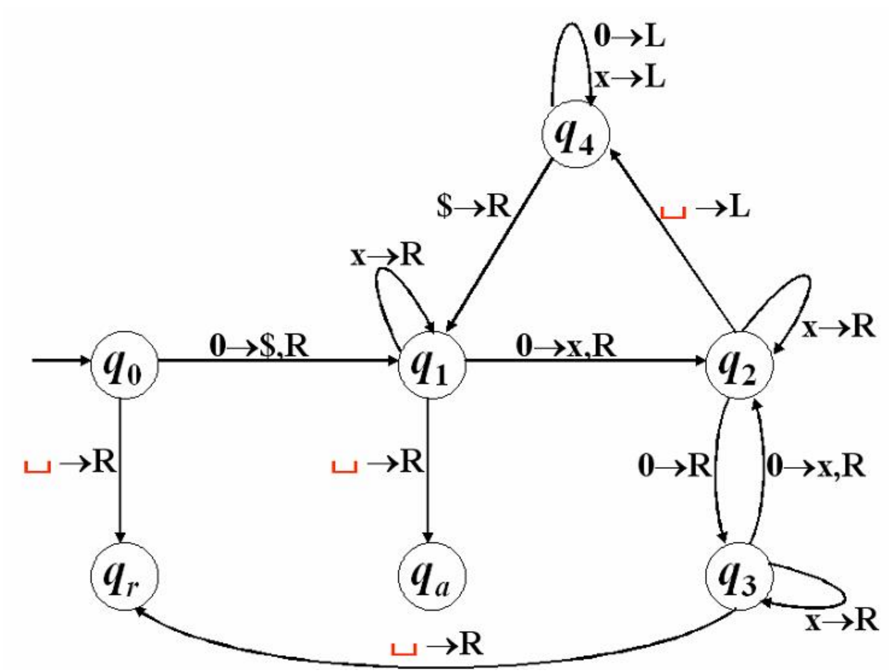
- 判定器M的运行时间或时间复杂度是f:N→N, f(n)是M在所有长为n的输入上运行的最大步数.
- 若f(n)是M的运行时间, 则称

  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

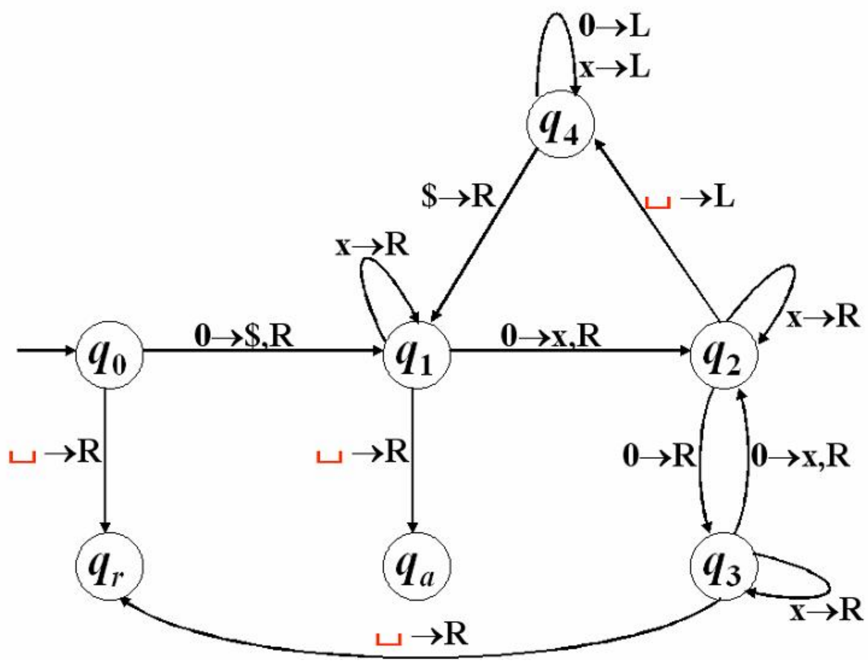


$q_0$0000
\$$q_1$000
\$x$q_2$00
\$x0$q_3$0
\$x0xq_2_
*\$x0q_4x_*
\$x$q_4$0x_
*\$$q_4$x0x_*
q_4\$x0x_
*\$$q_1$x0x_*
…

…
…
q_4\$xxx_
…
…
\$xxx_q_a

时间复杂性(P91)

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



$$f(1) = 2$$
$$f(2) = 7$$
$$f(3) = 4, \ldots$$
$$f(2^k) = (2k+1)2^k+1,$$
$$f(2n+1) = 2n+2, \ldots$$
$$n+1 \le f(n) \le 3n\log n$$

大O与小o记法(P154)

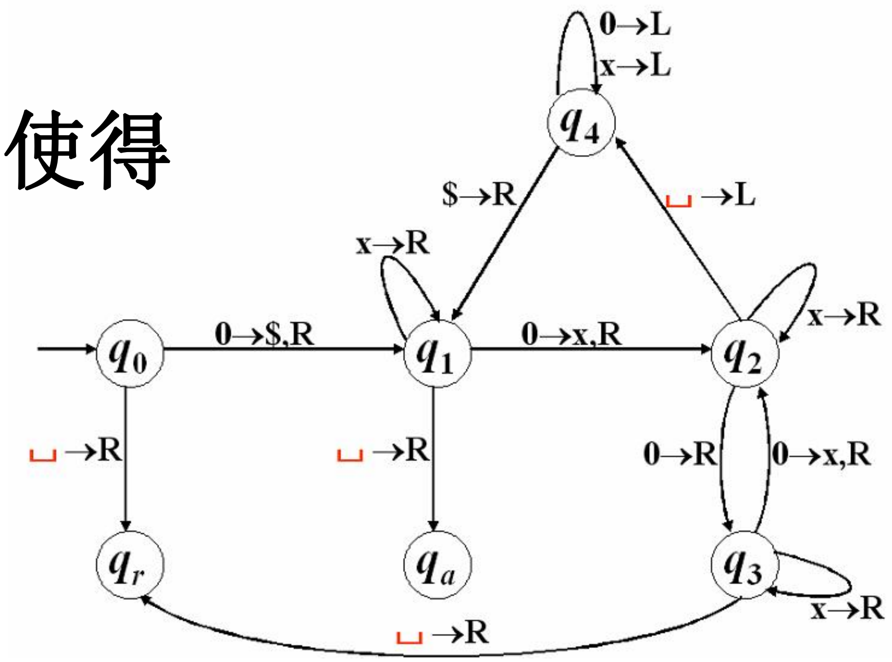
对于函数f,g:N→R$^+$,

记$f(n)=O(g(n))$,若存在$c>0$使得

$$\overline{\lim_{n\to\infty}}\frac{f(n)}{g(n)}\le c$$

记$f(n)=o(g(n))$,若

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



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

图灵机M_1(P155)

讨论语言$A = \{ 0^k1^k \mid k\geq 0 \}$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
$12+7\times3+3=36$

000011
*00011
$000x1
$$00xx
$$$0xx
reject
$12+9\times2+4=34$

001100
*01100
reject
5

时间复杂性类(P155)

定义: 对于函数$t:N\to N$,

时间复杂性类 TIME(t(n)) 定义为:

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

因为M_1是时间$O(n^2)$图灵机,

所以$A = \{0^k1^k : k\geq 0\} \in TIME(n^2)$.

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
$xx0xxxx1x
$xxxxxxxxx
accept
20×3+10=70

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

0001111
*001111
$0xx1x1
reject

00111111
*0111111
$0x1x1x1
$xxxx1xx
$xxxx1xx
reject

00011111
*0011111
$0xx1x1x
$xxxxx1x
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. $O(n)$

$\times \log n$

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.

总时间:

5)若带上同时没有0和1,则接受. $O(n)$ $O(n\log n)$

否则拒绝."

$\{0^k1^k|k\geq 0\}\in$TIME($n$log$n$) (P156)

由M_2知道A∈TIME(n log n). 有没有更快的TM识别A?

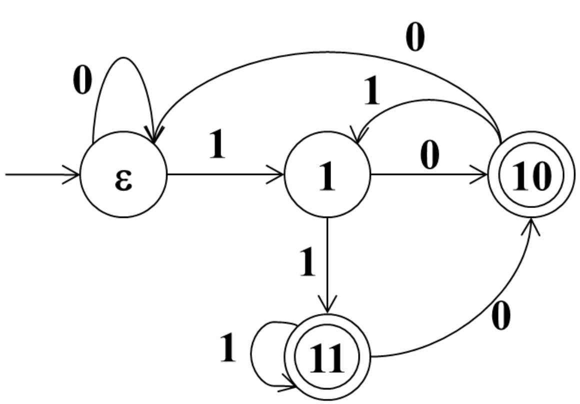
对于单带确定图灵机, 由

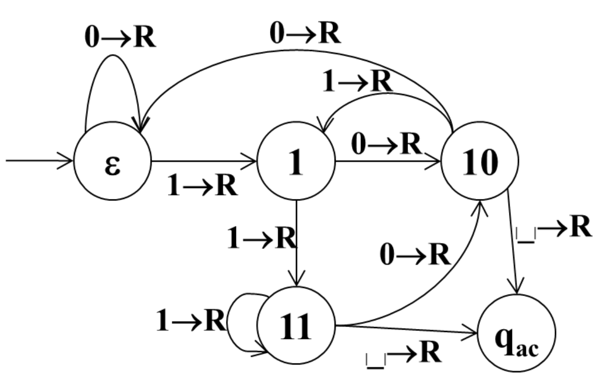
定理: 时间$o(n\log n)$的单带图灵机判定的语言是正则语言.

TIME($o(n\log n)$) ⊆ 正则语言类 ⊆ TIME(n) ⊆ TIME($o(n\log n)$)

正则语言类 = TIME(n) = TIME($o(n\log n)$)

非正则语言 $\{0^k1^k \mid k\geq 0\}\notin$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

   $\{ 0^k1^k \mid k\geq 0 \}$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)

$\{ 0^k1^k \mid k\geq 0 \}$ 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.

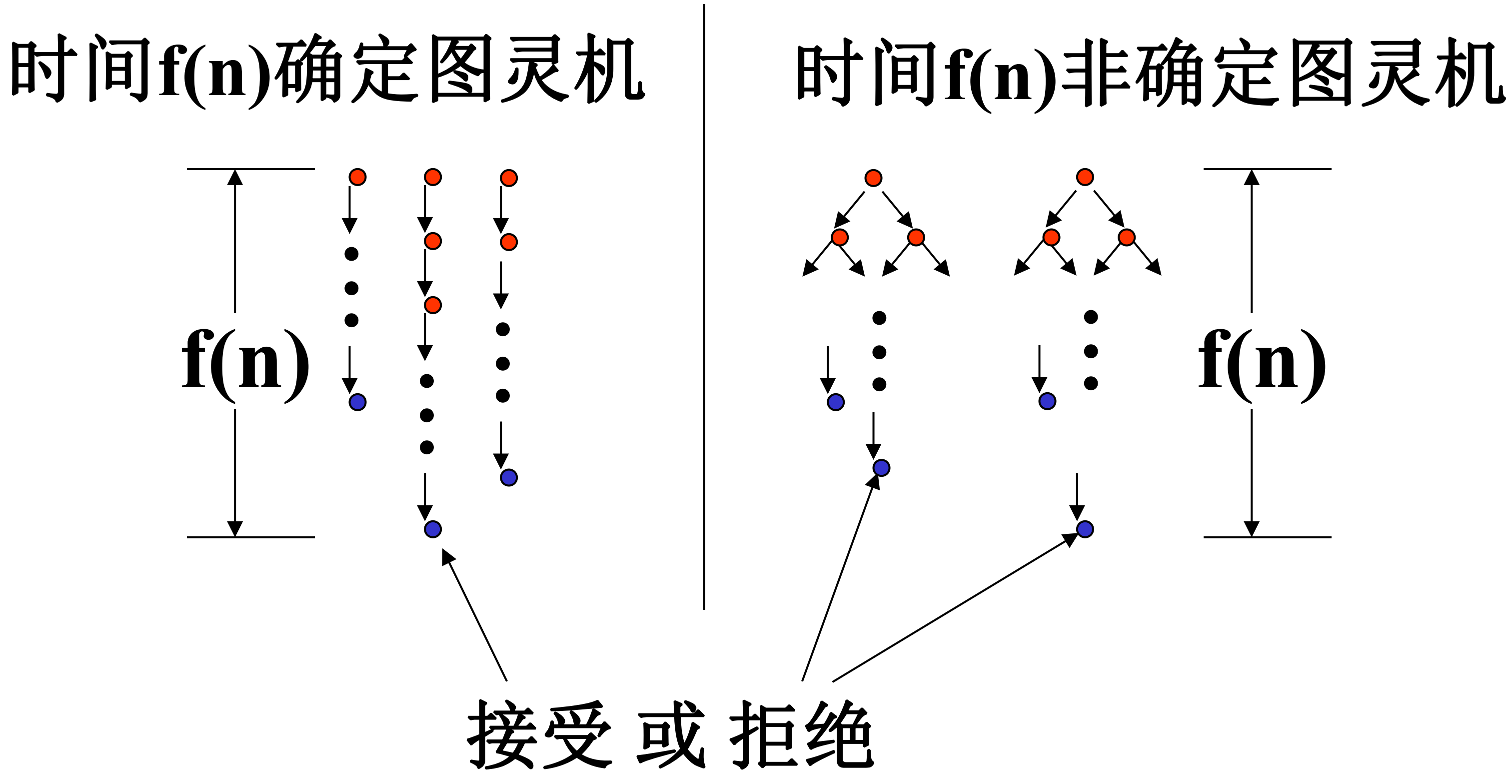
4) 如果两带上同时没有0和1,就接受."

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)

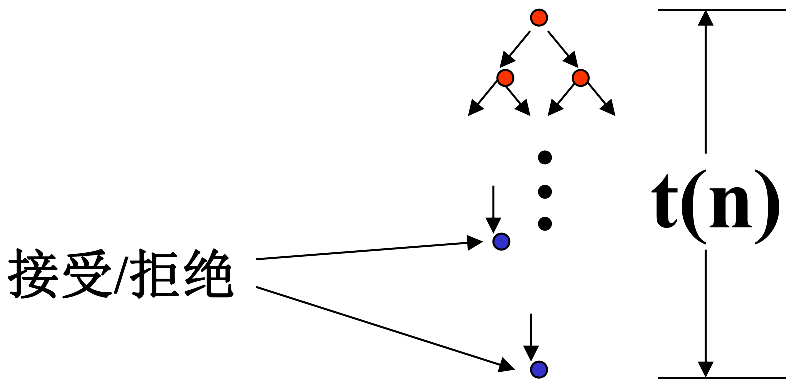
定义: 对非确定型判定器N, 其运行时间f(n)是在所有长为n的输入上, 所有分支的最大步数.



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$NTIME(t(n)) \subseteq TIME(2^{O(t(n))})$

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

   $\{ 0^k1^k \mid k\geq 0 \}$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: Prime = { p | p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

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

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

$$NP = \cup_k NTIME(n^k)$$

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

$$P \subseteq NP \subseteq EXP$$

$$P \subset EXP$$

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 = { <G,k> | G是有k团的无向图 }

定理: CLIQUE∈NP.

N="对于输入<G,k>,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={ <G,s,t> | 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<G,s,t>:

1)非确定地选G的所有节点的排列$p_1,\ldots p_m$.

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)

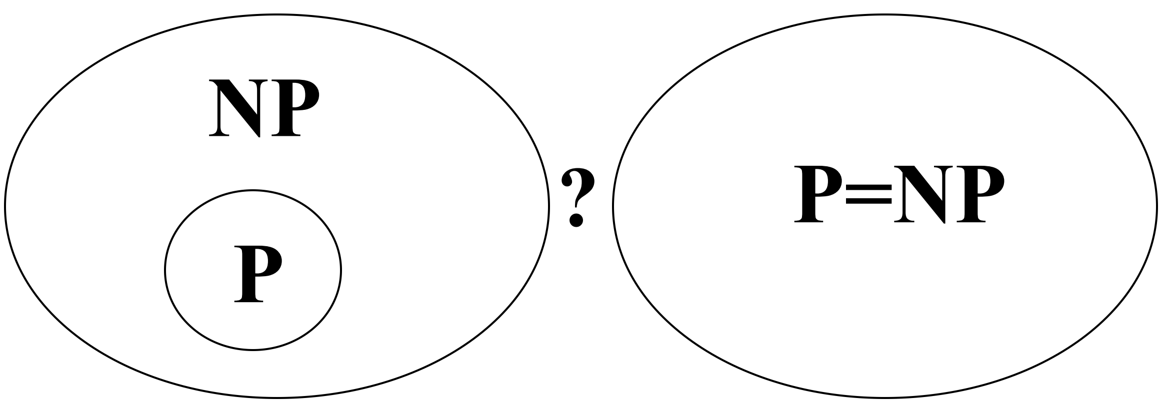
P=成员资格可以快速判定的语言类.

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

   $\{ 0^k1^k \mid k \geq 0 \}$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ϕ=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:

  SAT = { <ϕ> | ϕ是可满足的布尔公式 } NP完全

- 二元可满足性问题:

  2SAT = { <ϕ> | ϕ是可满足的2cnf } ∈ P

- 三元可满足性问题:

  3SAT = { <ϕ> | ϕ是可满足的3cnf } NP完全

二元可满足问题2SAT∈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.

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

$x_3=0 \rightarrow (x_4) \wedge (\neg x_4 \vee \neg x_5) \rightarrow (\neg x_5) \rightarrow \varnothing$ 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∈NP(P173)

三元可满足性问题:

3SAT = { <ϕ> | ϕ是可满足的3cnf }

P时间内判定3SAT的NTM:

N="对于输入< ϕ>, 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 ϕ=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 = { <ϕ> | ϕ是可满足的3cnf }

清洗算法对3cnf是否有效? 举例对比:

$(x_3 \lor \neg x_3) \land (x_1 \lor x_2) \land (x_1 \lor \neg x_2) \land (\neg x_1 \lor x_2) \land (\neg x_1 \lor \neg x_2)$

x_3=1清洗无矛盾; x_1=0和1都清洗失败, 不可满足.

$(x_3 \lor \neg x_3) \land (\neg x_3 \lor x_1 \lor x_2) \land (\neg x_3 \lor x_1 \lor \neg x_2) \land (\neg x_3 \lor \neg x_1 \lor x_2) \land (\neg x_3 \lor \neg x_1 \lor \neg x_2)$

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$f:\Sigma^*\to\Sigma^*$. (例如$f(u)=u0$)

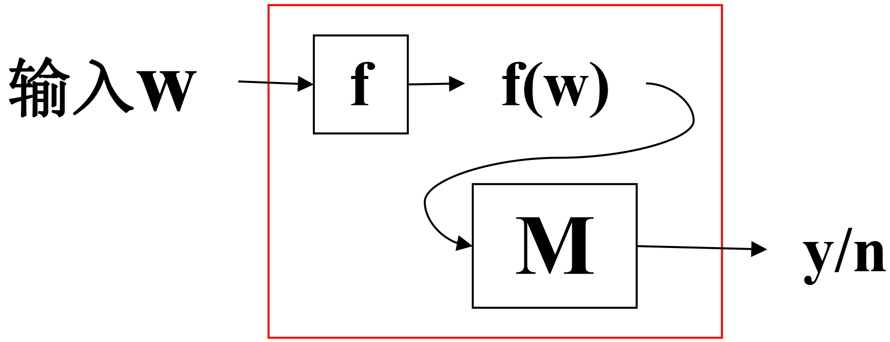
  若∃多项式时间图灵机, ∀w输入, 停机时带上的串为f(w)
- 定义:称A可多项式时间映射归约到B ($A\leq_p B$),

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

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

  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: SAT$\in$P$\Leftrightarrow$P=NP(P167-8)

- 定义:多项式时间可计算函数$f:\Sigma^*\to\Sigma^*$. (例如$f(u)=u0$)
  若∃多项式时间图灵机, ∀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 ($A\le_P B$),
  若存在多项式时间可计算函数$f:\Sigma^*\to\Sigma^*$,
  $\forall w\in\Sigma^*,\ w\in A \Leftrightarrow f(w)\in B$.
  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$\in$NP都有$A \le_P SAT$.
- 归约引理: 若 $A \le_P B$, 且 $B\in P$, 则 $A\in P$.
- 推论: 若SAT$\in$P, 则 NP = P.

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

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

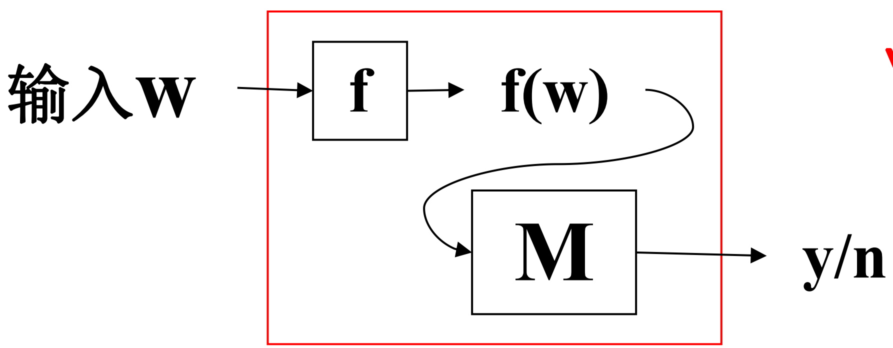
B有P时间判定器M, 则

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



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

利用f和B的判定器
构造A的判定器

定理: 3SAT $\leq_P$ CLIQUE (P168)

3SAT = $\{<\phi> \mid \phi$是可满足的3cnf公式 $\}$

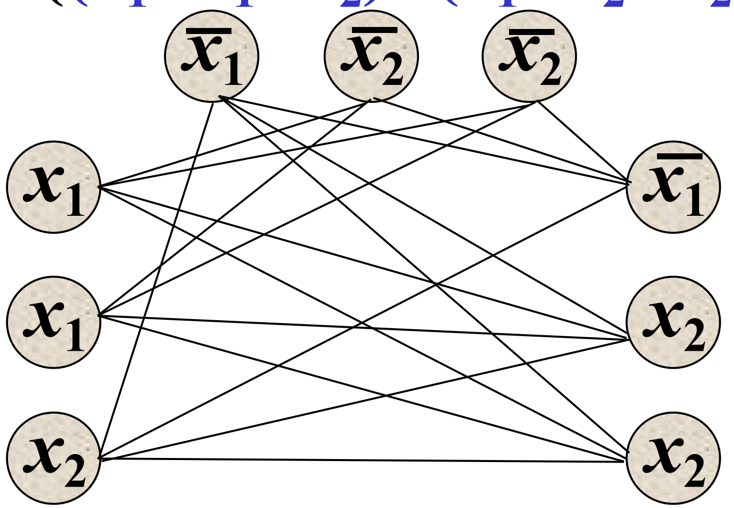
CLIQUE = $\{ <G,k> \mid G$是有k团的无向图 $\}$.

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

令 $f(\phi) = <G,k>$, 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: $f((x_1\vee x_1\vee x_2)\wedge(\overline{x_1}\vee\overline{x_2}\vee\overline{x_2})\wedge(\overline{x_1}\vee x_2\vee x_2)) = <G,3>$



需证:$<\phi>\in$3SAT

$\Leftrightarrow$

$<(G,k)>\in$CLIQUE

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

$\langle\phi\rangle(\langle(x_1\vee x_1\vee x_2)\wedge(\overline{x_1}\vee\overline{x_2}\vee\overline{x_2})\wedge(\overline{x_1}\vee x_2\vee x_2)\rangle) \in 3SAT$

$\Leftrightarrow$ ∃变量赋值(x_1=0, x_2=1)使得ϕ=1

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

$\Leftrightarrow$ $f(\phi)$ (<G,3>) $\in$ CLIQUE.

F在|<ϕ>|的多项式时间内可计算:

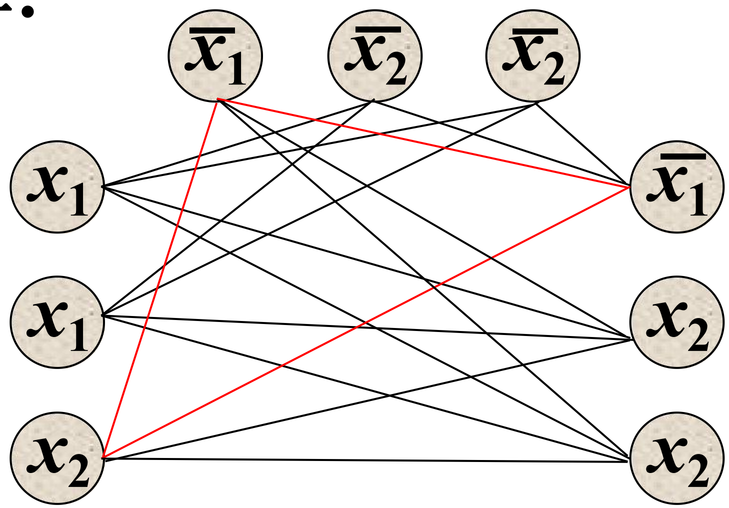
设ϕ的长度3k=O(k),

则G的顶点数3k=O(k),

G的边数是O(k^2)

可见 $f(\phi)$ = <G,k>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$A \leq_P B$.

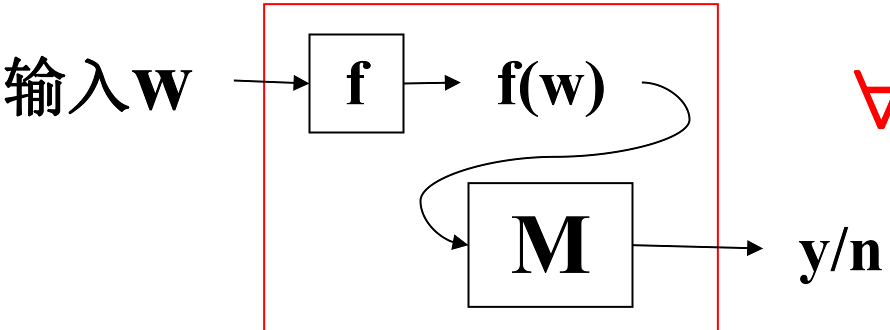
- 定理1: $A \leq_P B + B \in P \Rightarrow A \in P$.
- 定理2: 若B是NPC, 且$B \in P$, 则P=NP.

  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



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

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$A \le_p B$.

- Cook-Levin定理: SAT是NP完全问题.

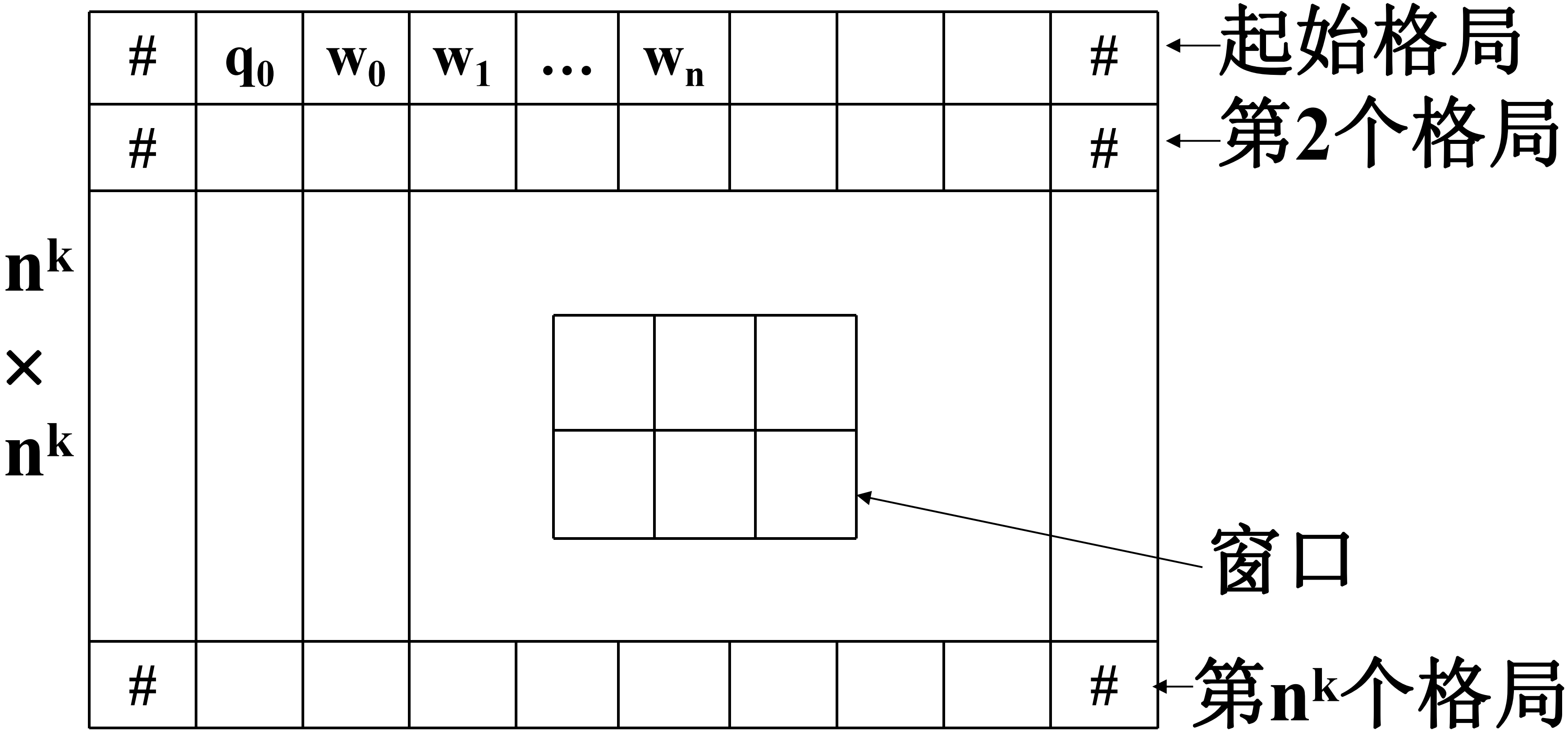
证明步骤:

1. $SAT \in NP$(已证)
2. $\forall A \in NP, A \le_p SAT$

$\forall A \in NP$, 都有 $A \leq_p SAT$ (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$A \in NP$, 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⇔能填好N在w上的表(P170)

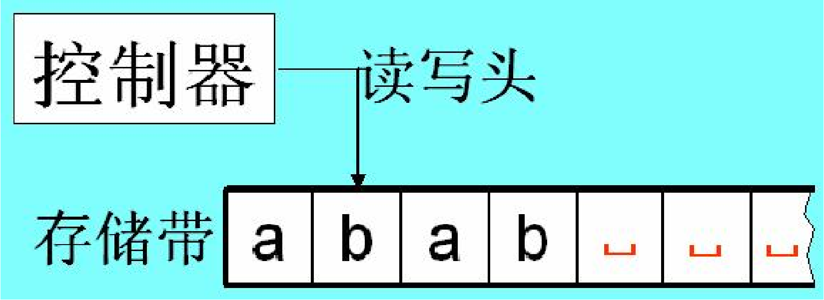


能填好表: 第一行是起始格局

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␣.

3) Γ是带字母表,其中␣$\in\Gamma$, $\Sigma\subset\Gamma$.

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:

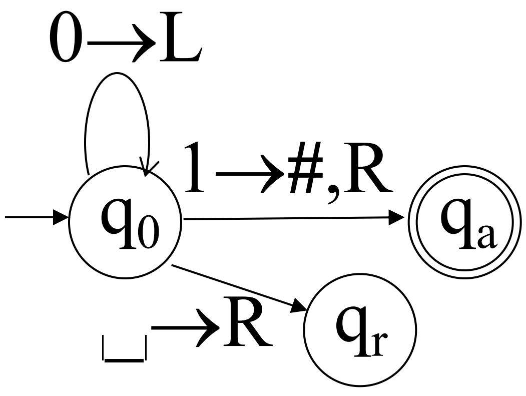
  控制器的状态；存储带上字符串; 读写头的位置.
- 定义: 对于图灵机$M=(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$,

  设$q \in Q$, $u,v \in \Gamma^*$, 则格局 uqv 表示

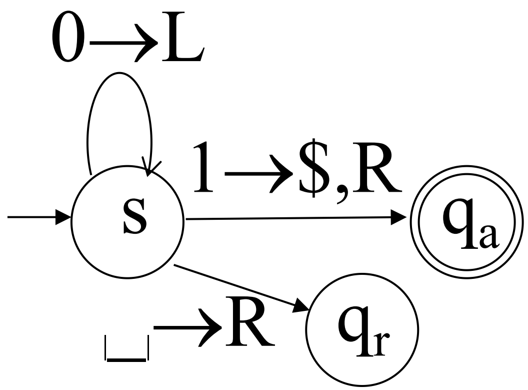
  1) 当前控制器状态为q；

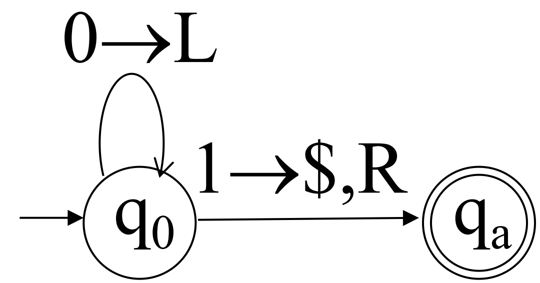
  2) 存储带上字符串为uv(其余为空格);

  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 0 1

…

循环

s 1 0

$ q_a 0

接受

s _ _

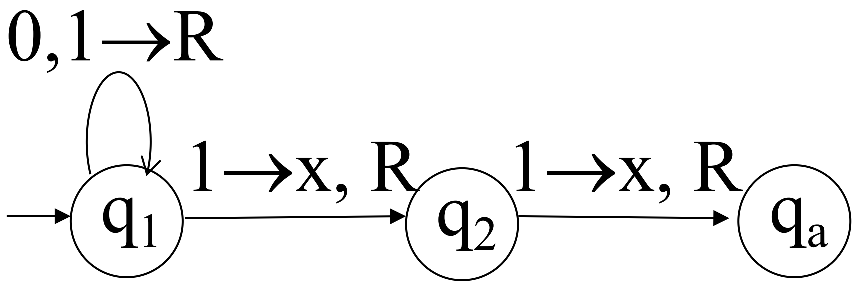
_ q_r _

拒绝

#	s	1	0	_	_	#
#	$	q_a	0	_	_	#

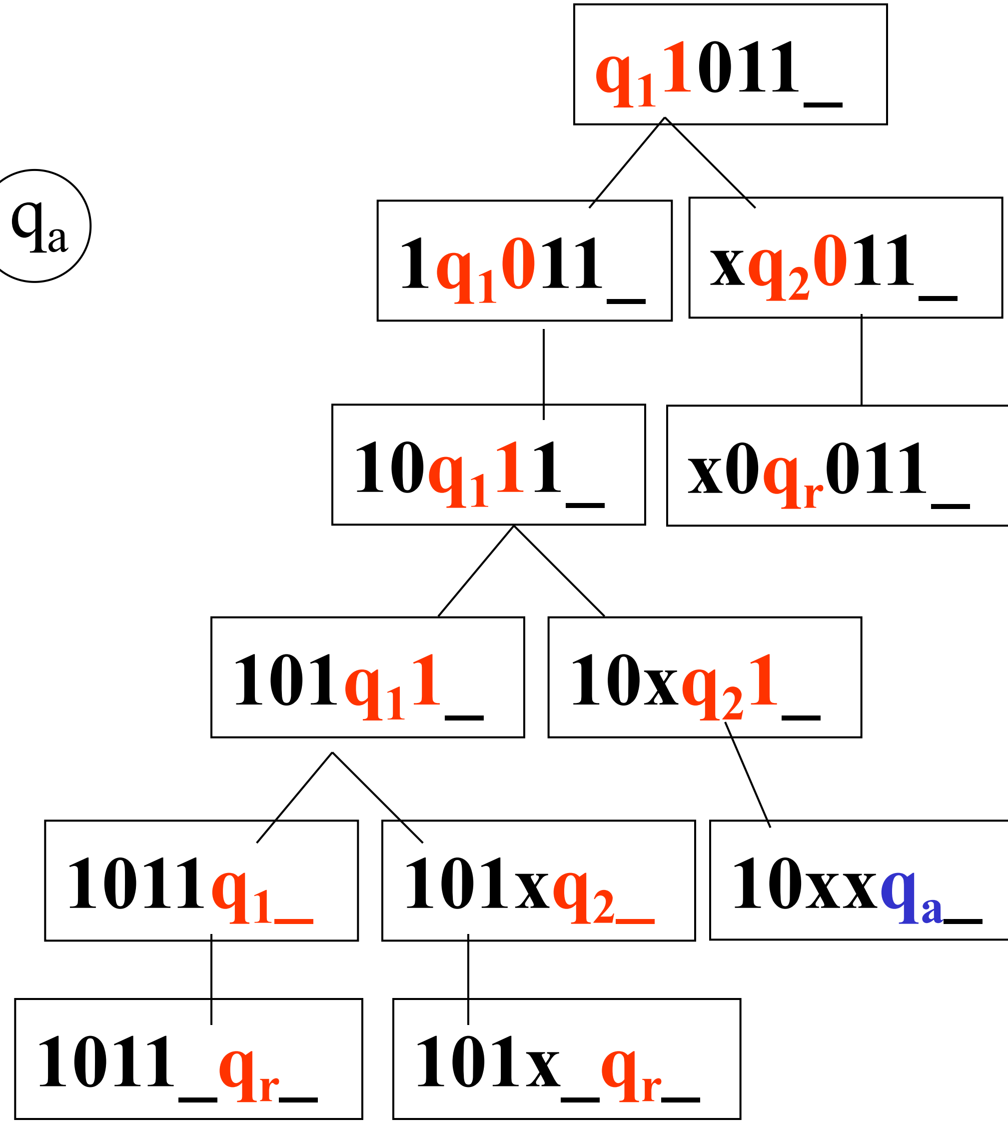
#	s	1	0	_	_	#
#	$	q_a	0	_	_	#
#	$	q_a	0	_	_	#
#	$	q_a	0	_	_	#

N接受w⇔能填好N在w上的表(补充)



NTM

#	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ϕ=f(w) (补充)

- 能填好画面 $\Leftrightarrow$ ϕ=f(w)可满足
- f(w)=<ϕ>， $\phi = \phi_{cell} \wedge \phi_{start} \wedge \phi_{move} \wedge \phi_{accept}$.
- 对于任意赋值:

1. $\phi_{cell} = 1$ $\Leftrightarrow$ 每格有且只有一个符号;
2. $\phi_{start} = 1$ $\Leftrightarrow$ 第一行是起始格局;
3. $\phi_{accept} = 1$ $\Leftrightarrow$ 表格中有接受状态;
4. $\phi_{move} = 1$ $\Leftrightarrow$ 每行由上一行格局产生.

- $\forall$w, w$\in$A $\Leftrightarrow$ <ϕ>$\in$SAT 即 A $\le_m$ SAT
- 若|<ϕ>|是|w|的多项式, 则有A $\le_p$ SAT

构造ϕ_{cell} (P170)

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

$x_{i,j,s}$: 第i行第j列是否填了符号s

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

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

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

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

- 长$O(n^{2k})$
- $\phi_{cell} = 1 \Leftrightarrow$ 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$O(\mathrm{n}^{\mathrm{k}})$
- $\phi_{\text{start}} = 1 \Leftrightarrow$ 第一行是起始格局;

构造ϕ_{accept}(P171)

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

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

构造ϕ_{move}(P171)

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

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

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

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

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

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

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	q_1	b
q_2	a	c

a	q_1	e
q_3	a	f

非法窗口

a	q_1	b
q_3	a	c

$A \le_P SAT$, SAT是NPC(P172)

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

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

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

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

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

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

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

推论:3SAT是NP完全的(P173)

只需将前面的ϕ改造为3cnf公式.

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

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

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

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

降子句长度: 给定赋值T $a_1\vee a_2\vee\ldots\vee a_k = 1 \quad \Leftrightarrow$

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

1个k-文字子句 变为 k-2个3-文字子句

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

ϕ_{move}的改造(P173)

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

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

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

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$M\times6^M\times n^{2k}$.

改造为3cnf后, 长度为$3\times(M-2)\times6^M\times n^{2k}$.

所以3SAT是NP完全的.

其它NP完全问题(补充)

SAT

3SAT

HP(哈密顿路径)

KS
(子集和, 01背包)

CLIQUE

UHP(无向图哈密顿路径)

UHC(无向图哈密顿圈)

TSP(旅行售货员问题)

HP是NPC(3SAT$\leq_P$HP)(P175)

HP={ <G,s,t> | G是有向图, 有从s到t的哈密顿路径 }

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

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

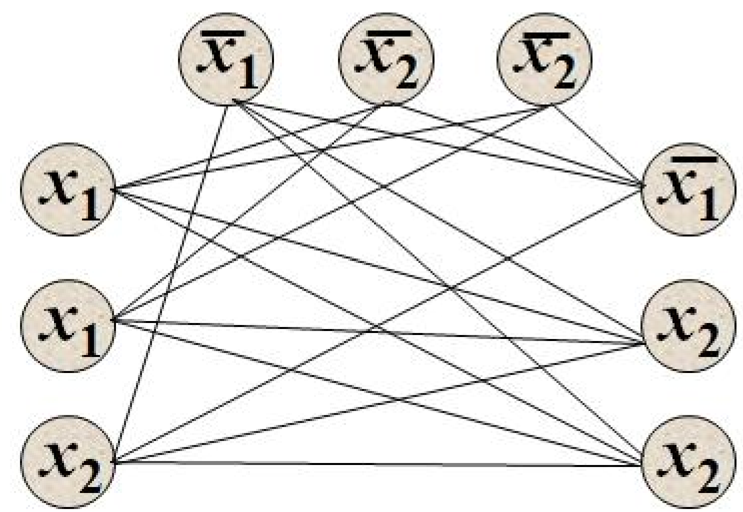
构造 f(ϕ) = <G,s,t> 使得 ϕ可满足$\Leftrightarrow$ G有从s到t的HP

一般由3cnf公式构造图有

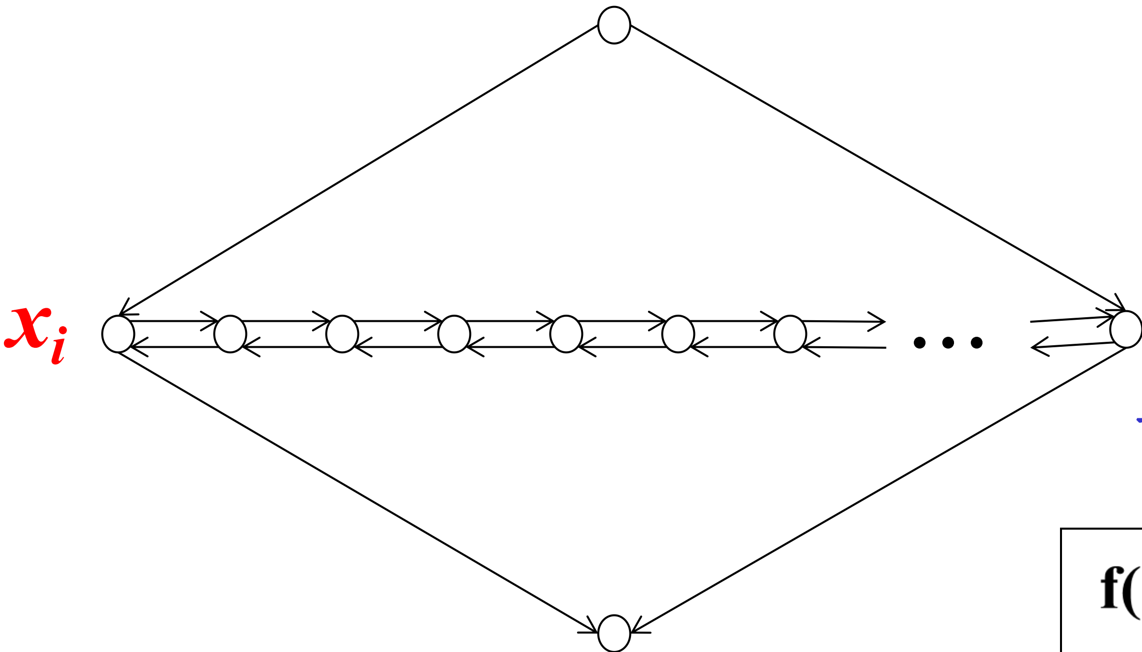
变量构件, 子句构件, 联接构件

如右图3SAT到CLIQUE归约中

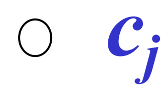
有子句构件和联接构件



变量构件和子句构件(P175)

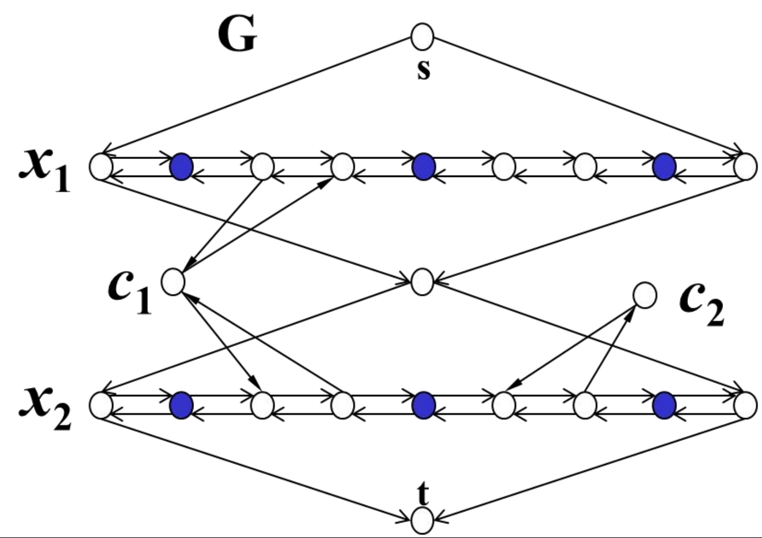


变量x_i表示为一个钻石结构

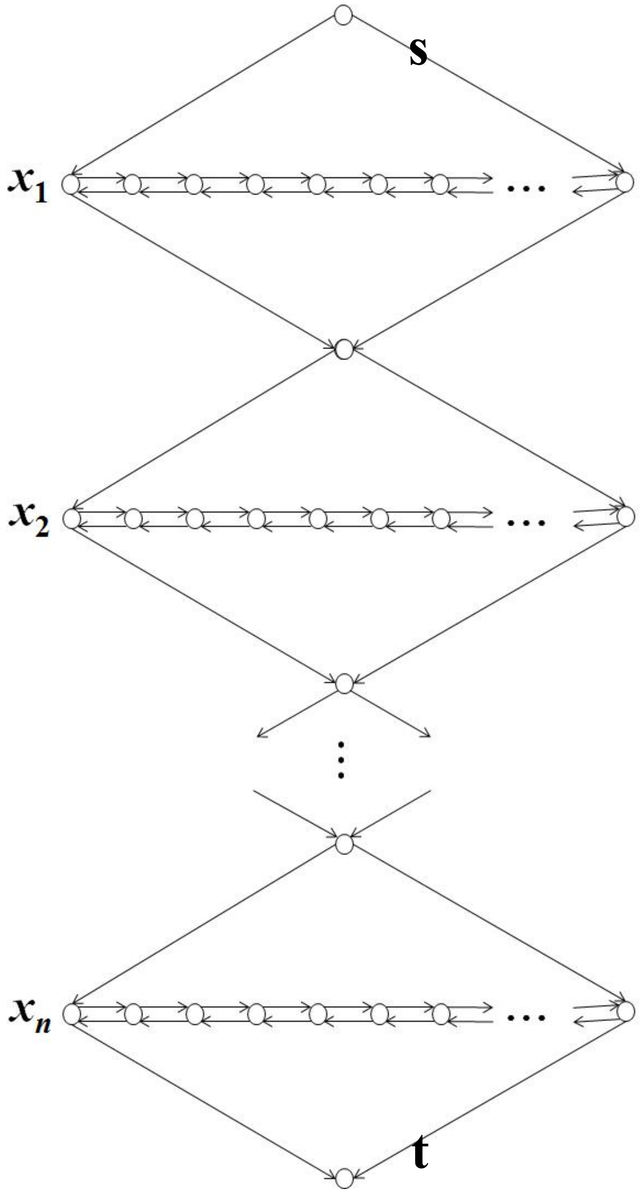


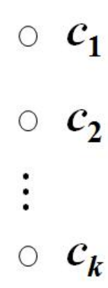
子句c_j表示为一个节点

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

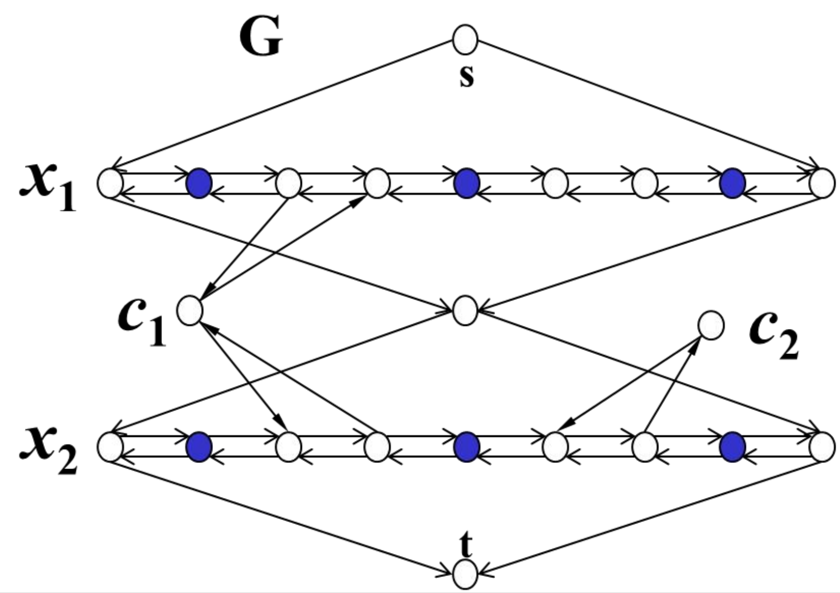


图G的总体结构





$f((x_1 \vee \neg x_2) \wedge (\neg x_2)) = $ <G,s,t>



对应

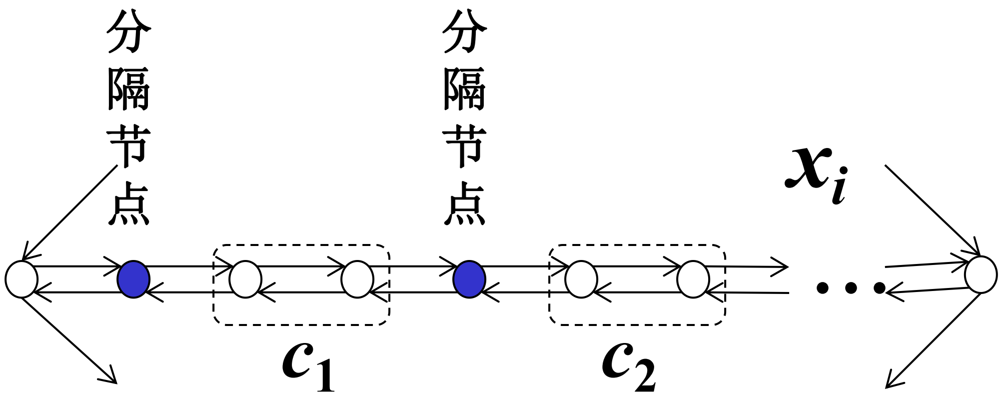
n个变量$x_1,\ldots,x_n$,

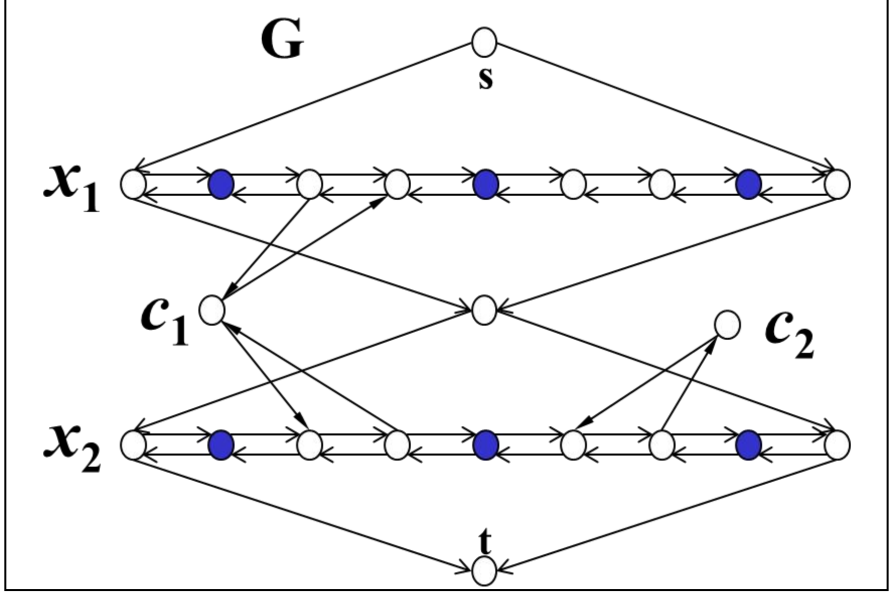
k个子句$c_1,\ldots,c_k$,

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





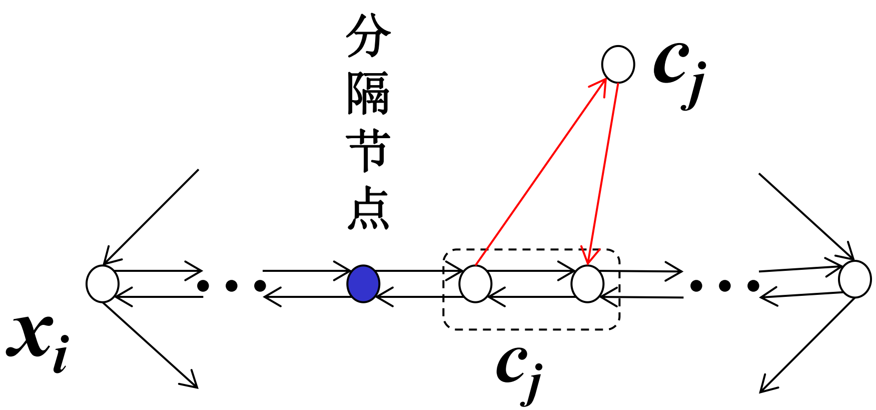
n: 变量个数, ***k***:子句个数

水平行除两端的两个节点外有**3*k*+1**个节点

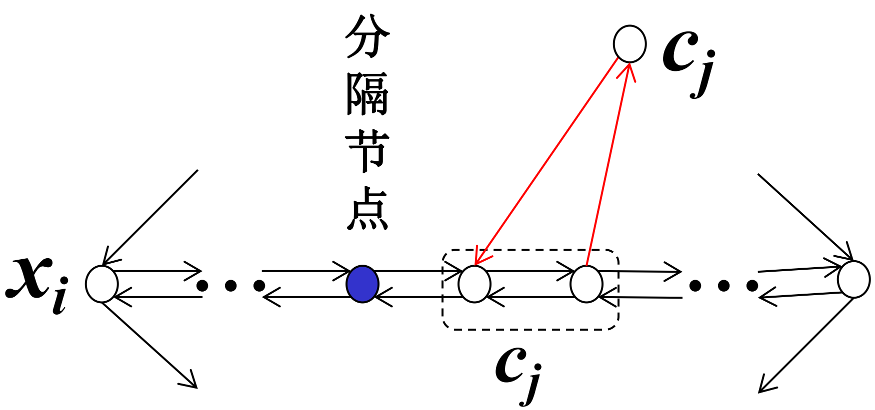
每个子句对应一对节点(共**2*k***个)

用分隔节点隔开(**k+1**个)

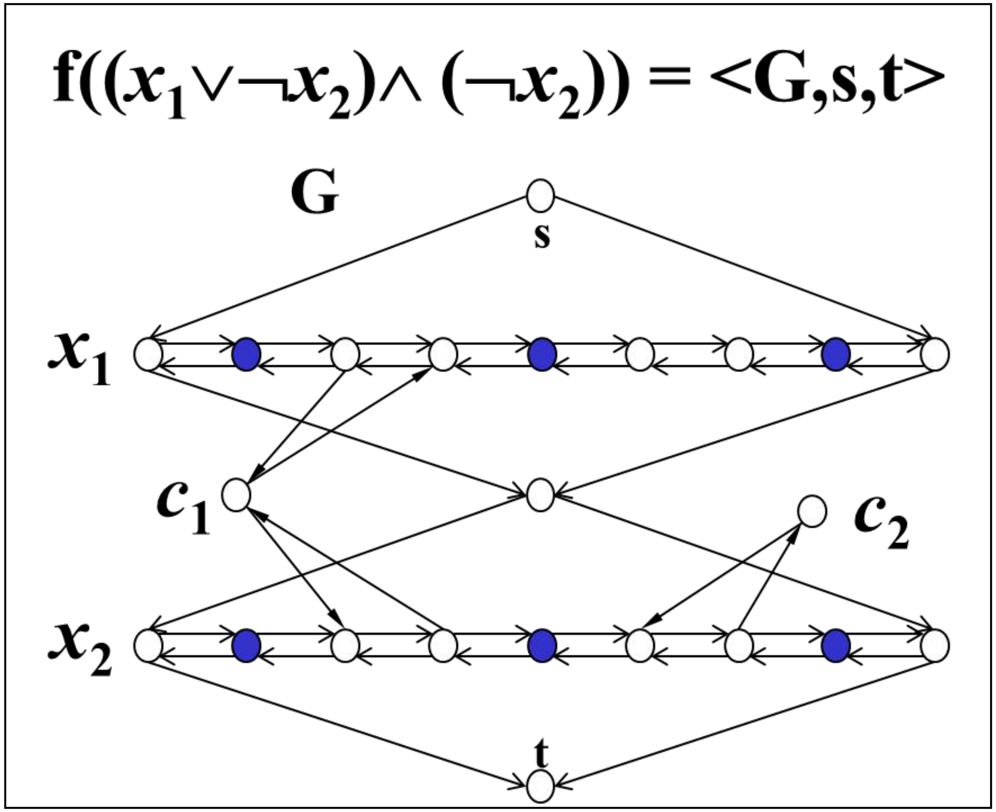
变量与子句构件的连接



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



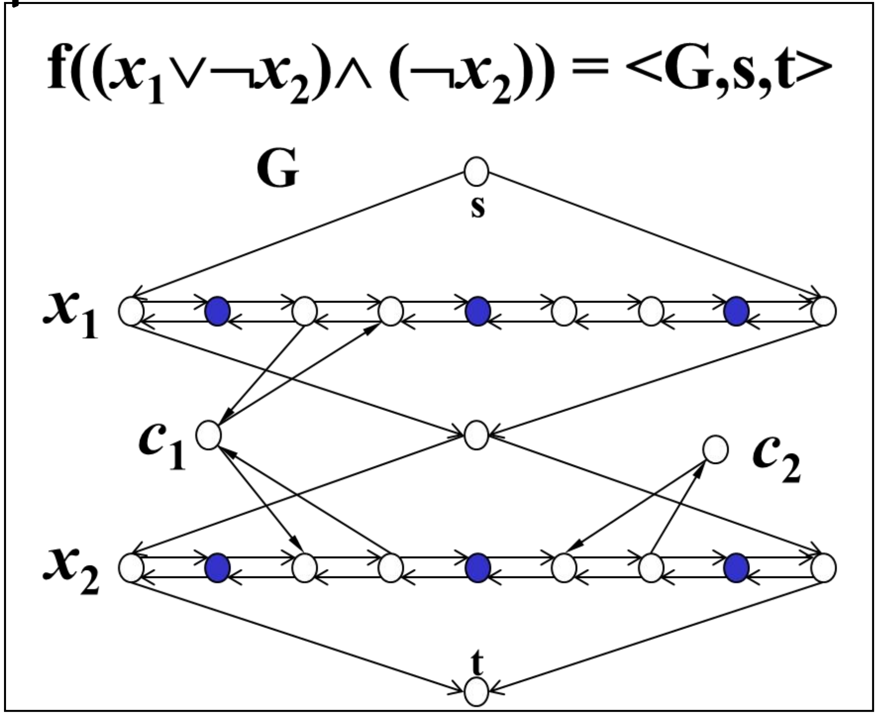
当子句c_j含有文字$\neg x_i$时添加的边
右-左式路径可以通过



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



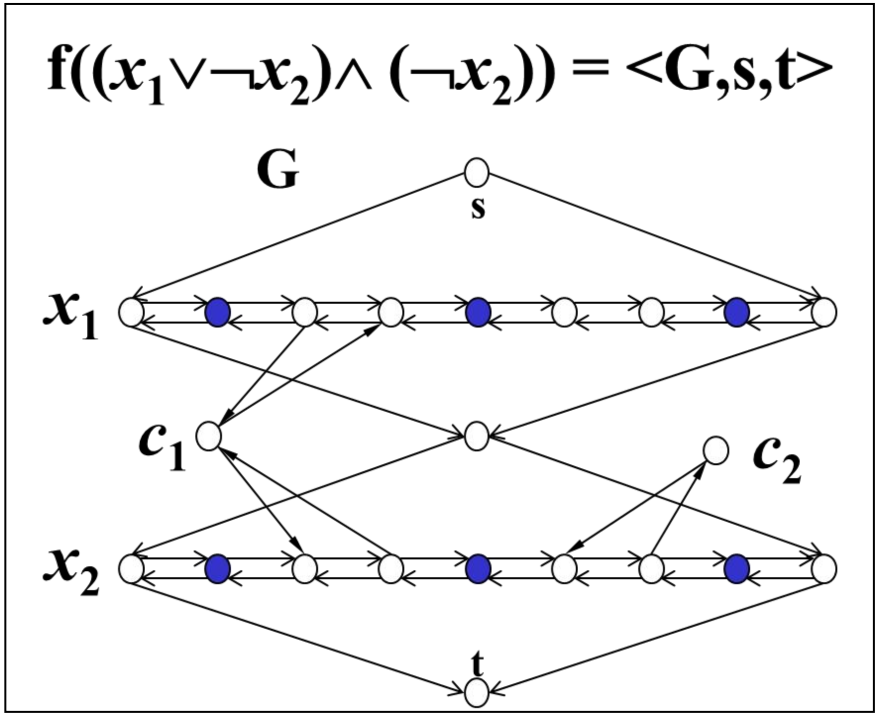
可满足赋值对应正规路径

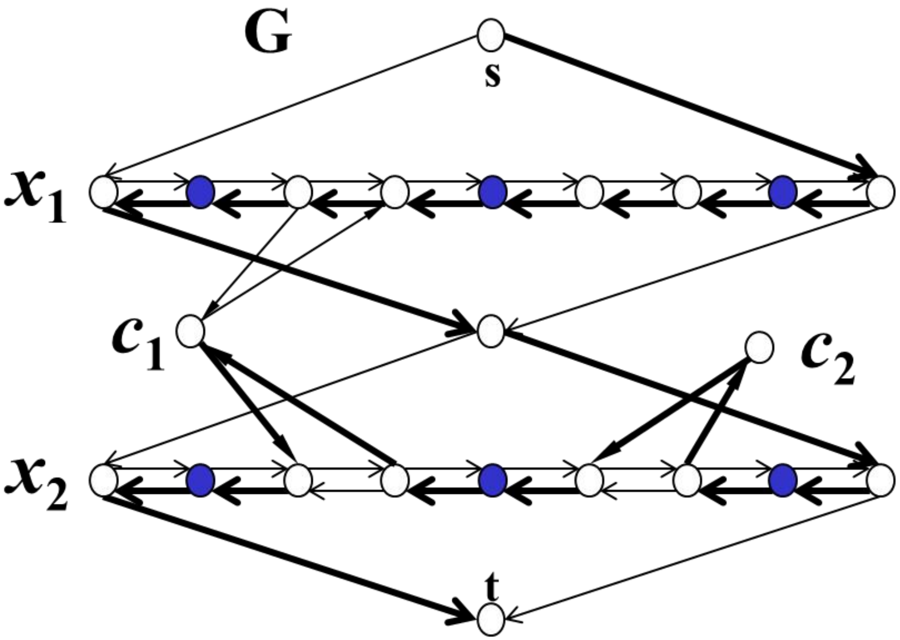
ϕ可满足 $\Rightarrow$ G有如下从s到t哈密顿路径

• 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

x_1=0, x_2=0,



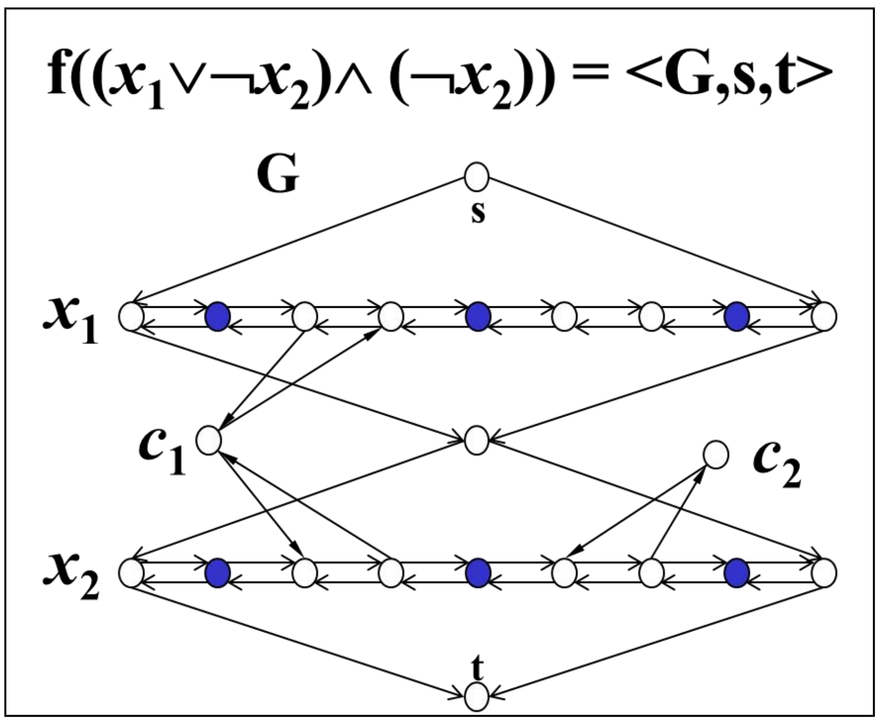


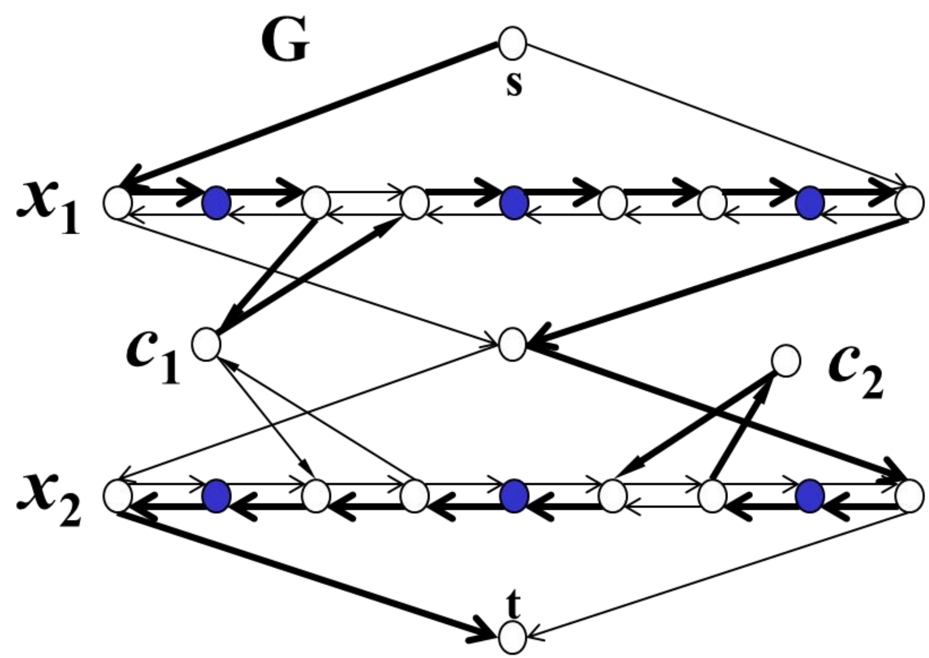
可满足赋值对应正规路径

ϕ可满足 ⇒ 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

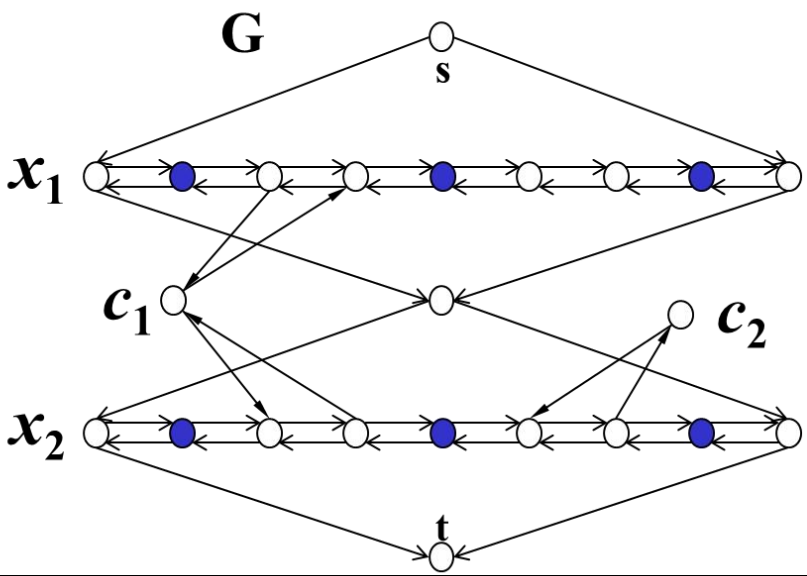
x_1=1, x_2=0,

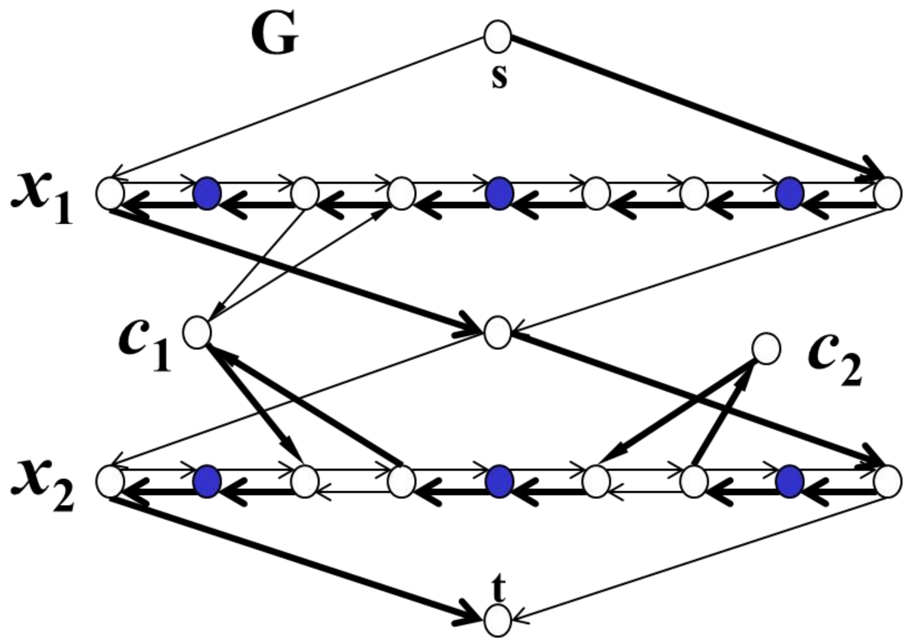




正规路径对应可满足赋值

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



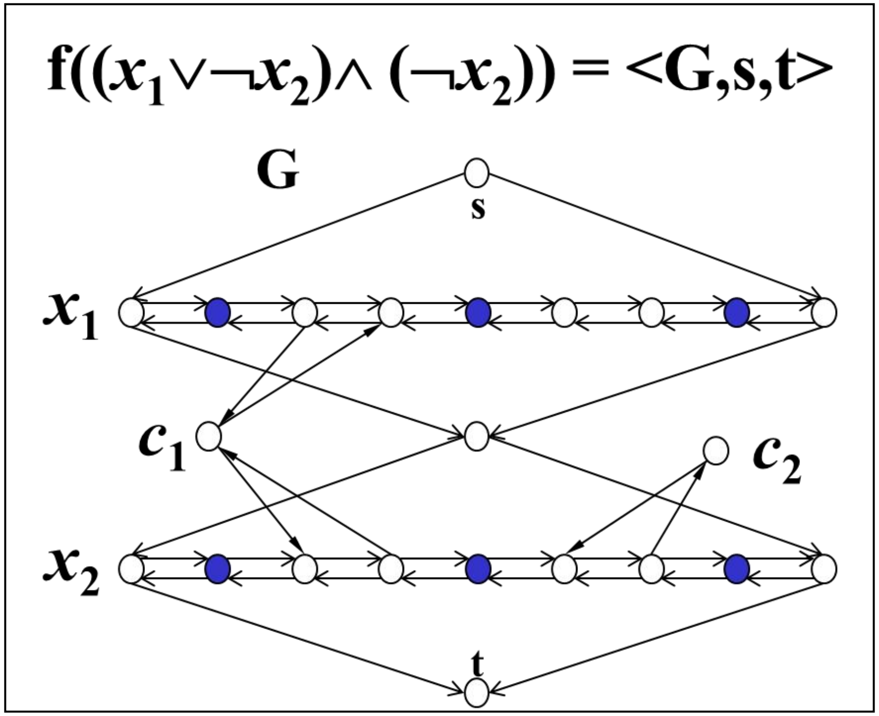


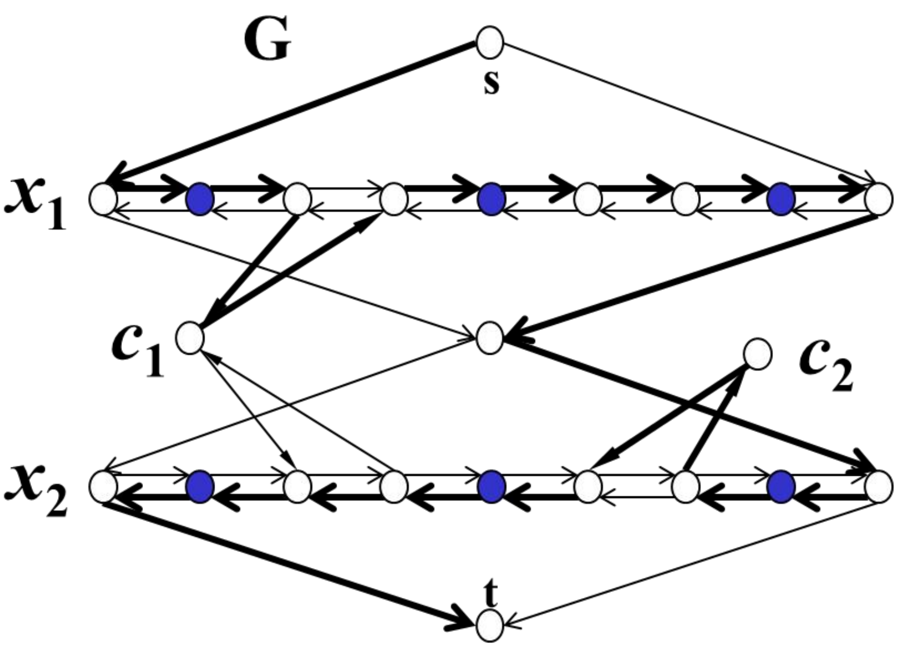
- 由左-右 或 右-左 式穿过钻石可确定变量赋值,
- c_j被穿过说明在对应变量赋值下c_j=1,

则公式φ可满足

右边正规路径对应x_1=0, x_2=0.

正规路径对应可满足赋值





- 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.

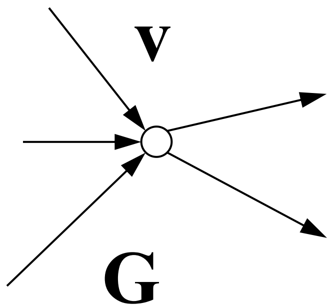
无向图哈密顿路径问题是NPC

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

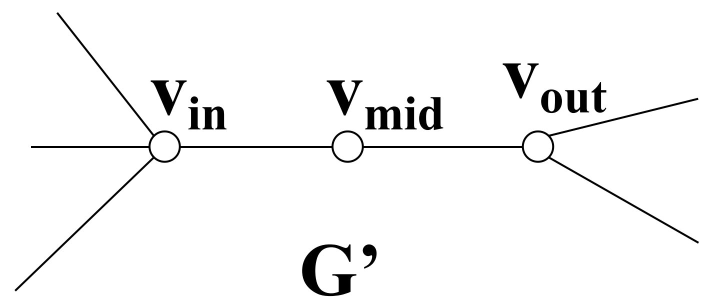
UHP = {$\langle G,s,t\rangle$| G是有从s到t哈密顿路径的无向图 }

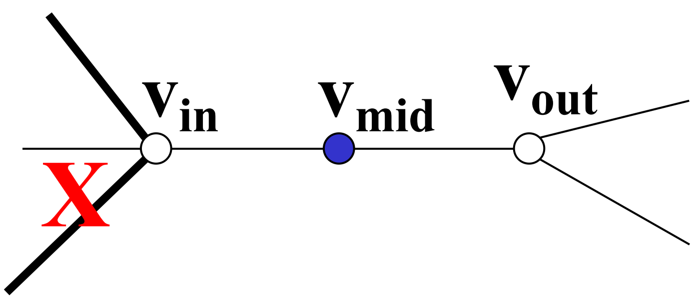
证明: **HP$\leq_P$UHP**, 映射归约如下 $\langle G,s,t\rangle \rightarrow \langle G',s_{out},t_{in}\rangle$

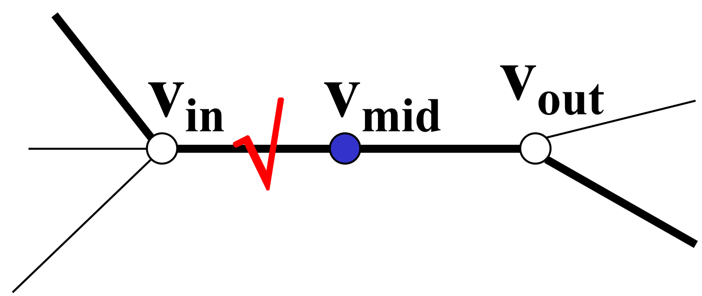
s对应s_{out}, t对应t_{in}, 其它每个节点v对应v_{in},v_{mid},v_{out},



→

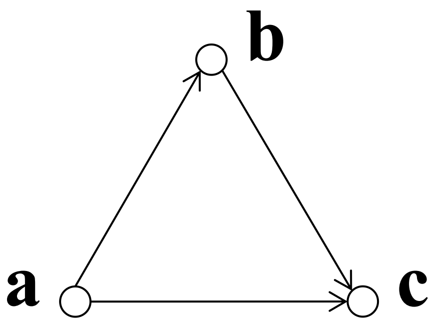


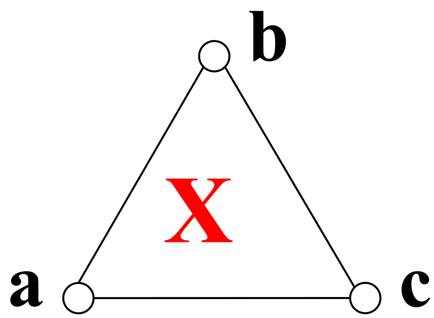


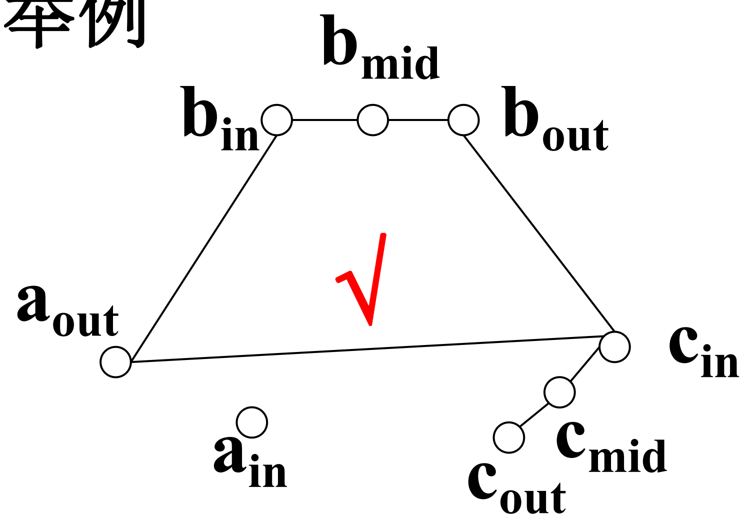


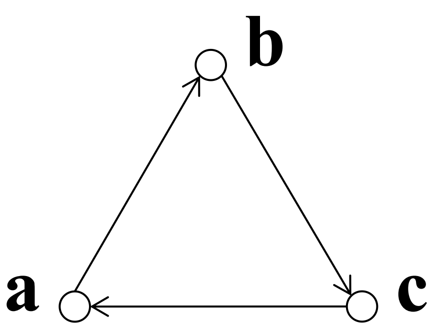
$HP \le_p UHP$

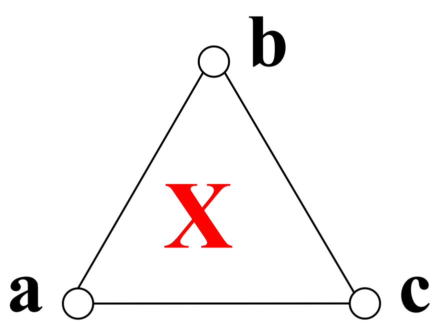
映射归约 $\langle G,a,a \rangle \rightarrow \langle G',a_{out},a_{in} \rangle$ 举例

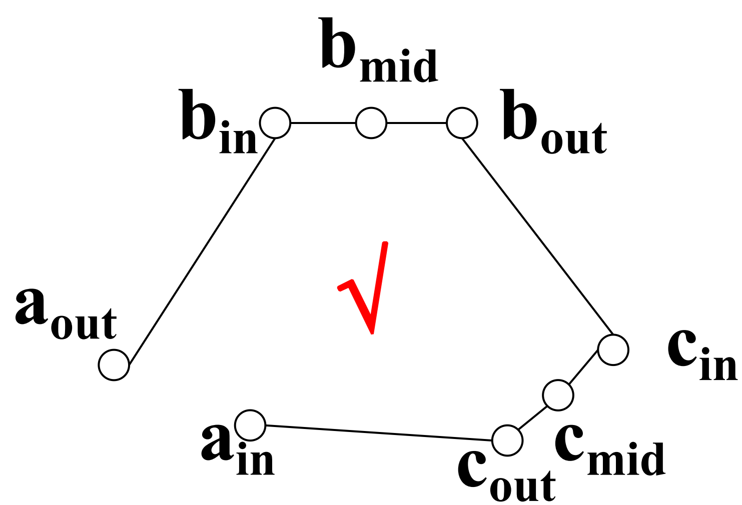












UHC是NP完全的(补充)

UHC = {<G>| G是有哈密顿回路的无向图 }

(1) UHC∈NP

构造多项式时间内判定UHC的非确定图灵机:

N="对于输入<G>, G是无向图,

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

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

(2) UHP$\leq_P$UHC

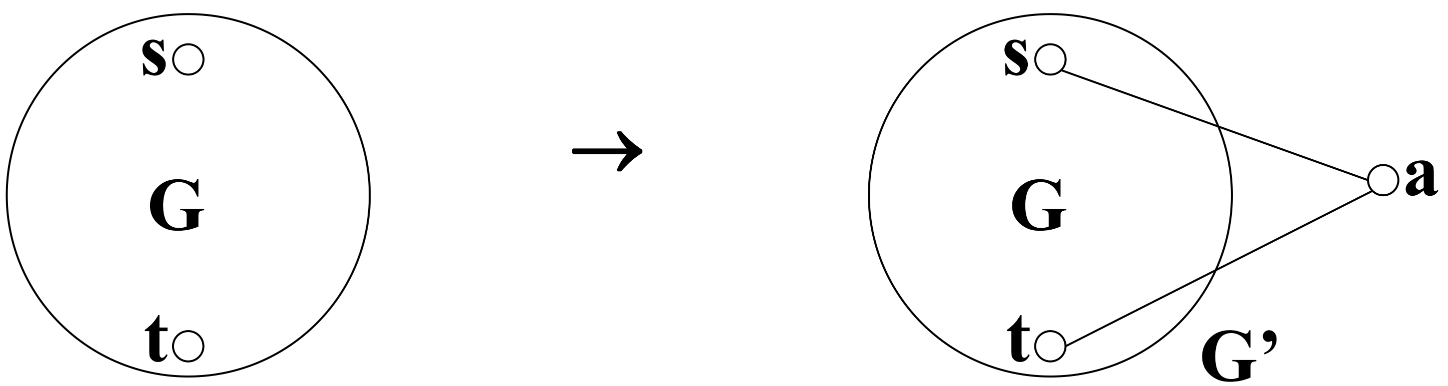
由(1), (2)和UHP是NP完全的, 得UHC是NP完全的

UHP$\leq_p$UHC

UHP = {<G,s,t>| G是有从s到t哈密顿路径的无向图 }

UHC = {<G>| G是有哈密顿回路的无向图 }

证明: 映射归约如下 <G,s,t> → <G'>



<G,s,t> → <G'> 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 ⇔ G'有哈密顿回路

TSP是NP完全的(补充)

TSP={<G,s,w,b> | 无向图G有 从s出发回到s, 权和≤b 的哈密顿回路 } //将TSP修改成决定性问题

(1) TSP∈NP. 构造多项式时间内判定TSP的NTM:

N="对于输入<G,s,w,b>, G是无向图,s是节点, w是权, b≥0,

1)非确定地选择G所有节点的排列$s,v_2,\ldots, v_n$.

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

3)否则拒绝."

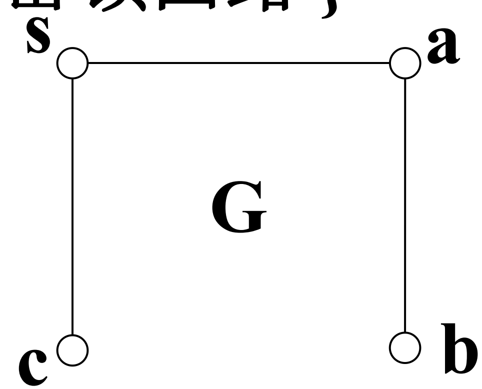
(2) $UHC \le_P TSP$

由(1), (2)和UHC是NP完全的, 得TSP是NP完全的

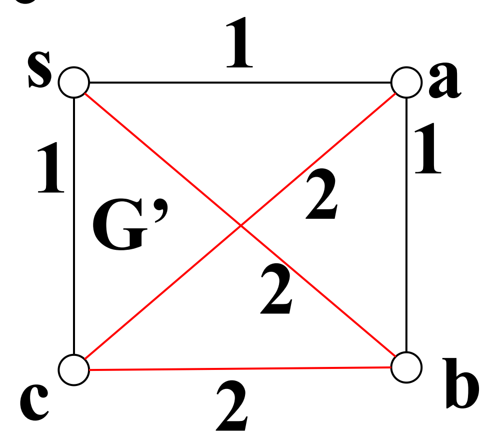
UHC $\le_p$ TSP

UHC = { <G> | G是有哈密顿回路(HC)的无向图 }

TSP = { <G,s,w,b> | G有s出发费用≤b的哈密顿回路 }



- 设G=(V,E), $s\in V=\{v_1,\ldots,v_n\}$//n个节点
- 令G’=(V,V×V), f(<G>) = <G′,s,w,n>,
- 定义权w: $$w[v_i,v_j]=\begin{cases}0 & 若 i=j \\ 1 & 若 (v_i,v_j)\in E \\ 2 & 其它\end{cases}$$



- f(<G>)增加边数≤n^2, 多项式时间可计算
- G有HC ⇒ G’有s出发费用≤n的HC
- G’有s出发费用≤n的HC

 ⇒ 该回路上的边都在G中 ⇒ G有HC

0-1背包(knapsack)问题是NPC

[S]中称为子集和问题.

KS = { $< A,t >$ | t 等于A中一些数的和 }

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

设ϕ是3cnf公式, 构造 f(<ϕ>) = $< A,t >$

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

构造数集 $A = \{ y_1,\ldots,y_n, z_1,\ldots,z_n, g_1,\ldots,g_k, h_1,\ldots,h_k \}$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$\le(n+k+1)^2$.

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

归约举例1

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

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

	x_1	x_2	c_1	c_2
y_1	1 单位阵	0	1	0
y_2	0	1	0	0
z_1	1 单位阵	0	0	0
z_2	0	1	1	1
g_1	0	0	1 单位阵	0
g_2	0	0	0	1
h_1	0	0	1 单位阵	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 (<$(x_1 \vee \neg x_2)$>) = <{101,10,100,11,1,1},113 >

	x_1 x_2 … x_n	c_1 c_2 … c_k
y_1 … y_n	$yx_{ij}=\begin{cases}1 & i=j\\0 & else\end{cases}$	$yc_{ij}=\begin{cases}1 & \text{若}c_j\text{中有}x_i\\0 & else\end{cases}$
z_1 … z_n	$zx_{ij}=\begin{cases}1 & i=j\\0 & else\end{cases}$	$zc_{ij}=\begin{cases}1 & \text{若}c_j\text{中有}\neg x_i\\0 & else\end{cases}$
g_1 … g_k	0	$gc_{ij}=\begin{cases}1 & i=j\\0 & else\end{cases}$
h_1 … h_k	0	$hc_{ij}=\begin{cases}1 & i=j\\0 & else\end{cases}$
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(<ϕ>)$\in$KS(knapsack)

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

	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$\Rightarrow f(\langle\phi\rangle)\in KS$

$f(\langle\phi\rangle) = \langle A, t\rangle$,

$A = \{ y_1,\ldots,y_n,z_1,\ldots,z_n,$

$g_1,\ldots,g_k,h_1,\ldots,h_k\}$

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

若ϕ有满足赋值($\langle\phi\rangle\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≥1, ≤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 $f(\langle\phi\rangle)\in KS$

ϕ可满足$\Leftarrow$ f(<ϕ>)$\in$KS

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

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

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

若f(<ϕ>)$\in$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≥ 1, ≤ 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HALF-CLIQUE = { <G> | G是无向图, 包含结点数至少为m/2的完全子图, m是G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) HALF-CLIQUE∈NP

构造如下非确定图灵机

N=“对于输入<G>, 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HALF-CLIQUE∈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<G,k>，其中G是一个无向图，k是一个正整数。构造函数f(<G,k>) = 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⇔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