计算理论
第一部分 计算模型

[S]

第**1**章 有限自动机

第**3**章 图灵机

第1章 有限自动机

字符串与语言

字母表: 任意一个有限集. 常用记号Σ, Γ.

符号: 字母表中的元素

字符串: 字母表中符号组成的有限序列

如**asdf**, 通俗地说即单词

串的长度|·|, 例: |**abcde**|=**5**

串的连接*, 例: **(abc)*(de)=abcde**

串的反转**R**, 例: $\mathbf{(abcde)^R=edcba}$

空词: 记为ε, 长度为**0**

语言: 给定字母表上一些字符串的集合

Σ^*, 语言, 字典序

取字母表 $\Sigma = \{0,1\}$, Σ上的语言举例:

A={0,00,0000}, B={0,00,01,000,001,…}

- Σ上所有有限长串记为Σ^*.
- Σ上的任一语言都是Σ^*的子集.
- Σ^*(字典序): ε, 0, 1, 00, 01, 10, 11, 000, …
- Σ上所有无限长串记为Σ^N.
- Σ上的语言与Σ^N一一对应.

决定性问题与语言一一对应

决定性问题(**Dicision Prob**): 只需回答是与否的问题

“一数是否是偶数” **--------------{** 以**0**结尾的**01**串 **}**

“串**0,1**个数是否相等” **------{ 0,1**个数相等的**01**串 **}**

“图是否连通” **------------------{ <G> | G**是连通图 **}**

其中**<G>**是图**G**编码成的字符串.

给定有限字母表Σ,

- 每个输入是一个串, 任意串都可以是输入串
- 一个决定性问题是满足某性质的串的集合(语言)

确定型有限(穷)自动机的形式定义

定义: 有限自动机是一个5元组(Q,Σ,δ,s,F),

1) Q是有限集, 称为状态集;

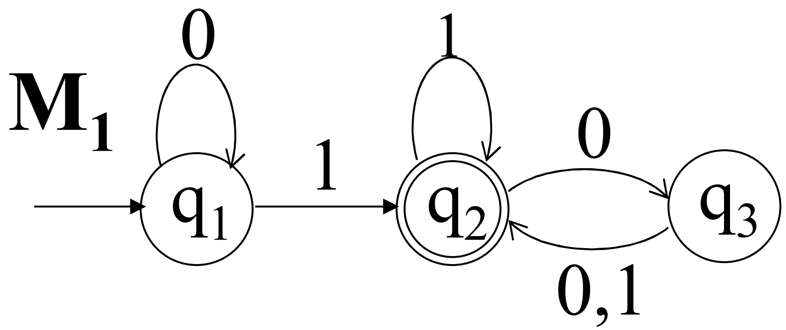
2) Σ是有限集, 称为字母表;

3) δ: Q× Σ→Q是转移函数;

4) s∈Q是起始状态;

5) F⊆Q是接受状态集;

读写头不能改写, 且只能右移



• 状态图等价于形式定义

δ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

$Q=\{q_1,q_2,q_3\}$, 状态集

$\Sigma=\{0,1\}$, 字母表

$s=q_1$, 起始状态

$F=\{q_2\}$接受状态集

有限自动机的语言:正则语言

对有限自动机**M**, 若 $\mathbf{A} = \{\, \mathbf{w} \in \Sigma^* \mid \mathbf{M}\text{接受}\mathbf{w} \,\}$,

即**A**是有限自动机**M**的语言, 记为**L(M)=A**, 也称**M**识别**A.**

若存在**DFA**识别语言**A**, 则称**A**是正则语言.

称两个有限自动机等价若它们语言相同.

有限自动机的设计

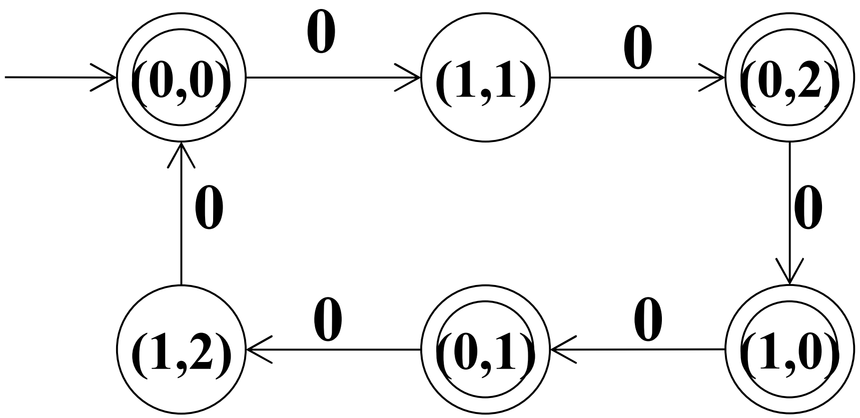
- 自己即自动机
- 寻找需要记录的关键信息

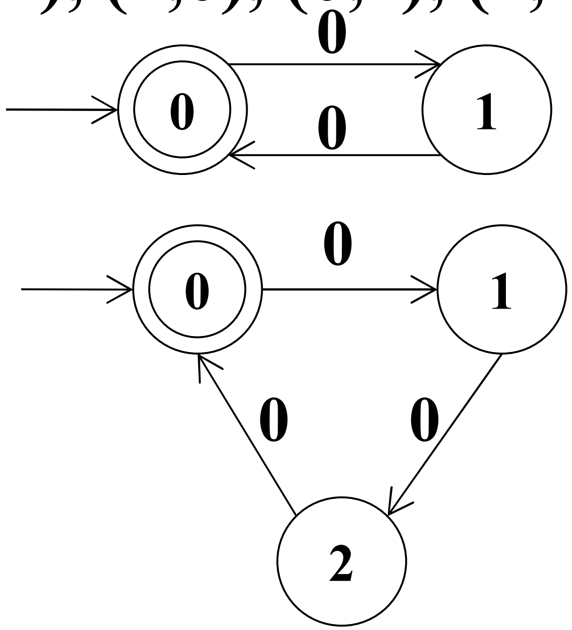
有限自动机的设计

{ 0^k | k是2或3的倍数 }

Σ={0}, 关键信息: $\varepsilon,0^1,0^2,0^3,0^4,0^5$,

记为: 0,1,2,3,4,5 或 (0,0), (1,1), (0,2), (1,0), (0,1), (1,2)





{0^k|k是2或3的倍数} = {0^k|k是2倍数} ∪ {0^k|k是3的倍数}

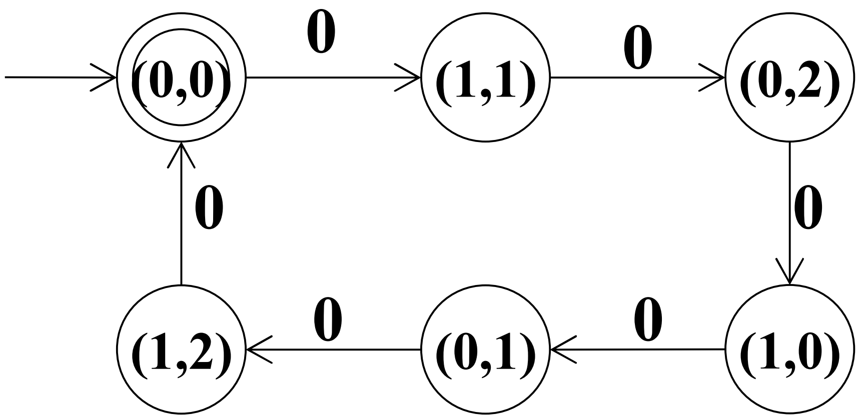
$$F = F_1 \times Q_2 \cup Q_1 \times F_2$$

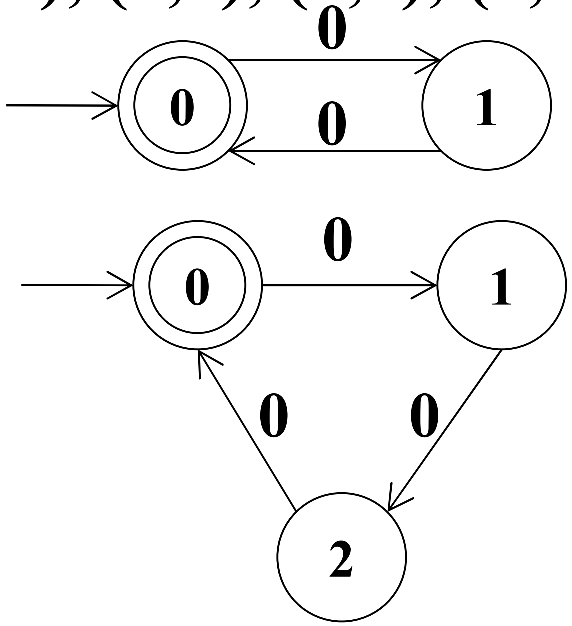
有限自动机的设计

$\{ 0^k \mid k$是2和3的倍数 $\}$

$\Sigma=\{0\}$, 关键信息: $\varepsilon, 0^1, 0^2, 0^3, 0^4, 0^5$,

记为: 0,1,2,3,4,5 或 (0,0), (1,1), (0,2), (1,0), (0,1), (1,2)





$\{0^k \mid k$是2和3的倍数$\} = \{0^k \mid k$是2倍数$\} \cap \{0^k \mid k$是3的倍数$\}$

$F=F_1 \times F_2$

正则语言与正则运算

如果语言A被一DFA识别,则称A是正则语言

定义: 设A和B是两个语言,定义正则运算

并,连接,星号如下:

- 并: $A \cup B = \{x | x \in A \text{或} x \in B\}$
- 连接: $A \circ B = \{xy | x \in A \text{且} y \in B\}$
- 星号: $A^* = \{x_1 x_2 \ldots x_k | k \geq 0 \text{且每个} x_i \in A\}$

正则语言的并是正则语言

定理: 设A,B都是Σ上的正则语言, 则$A\cup B$也是正则语言.

证明: 设$M_1=(Q_1,\Sigma,\delta_1,s_1,F_1)$和$M_2=(Q_2,\Sigma,\delta_2,s_2,F_2)$是DFA,

且 $L(M_1)=A$, $L(M_2)=B$,

令$Q=Q_1\times Q_2$, $s=(s_1,s_2)$, $F = F_1\times Q_2 \cup Q_1\times F_2$,

δ: $Q\times\Sigma\rightarrow Q$, $\forall a\in\Sigma$, $r_1\in Q_1$, $r_2\in Q_2$,

$$\delta((r_1,r_2) , a) = (\delta_1(r_1,a), \delta(r_2,a)),$$

即对$i=1,2$, 第i个分量按M_i的转移函数变化.

令$M=(Q,\Sigma,\delta,s,F)$, 则 $\forall x (x\in L(M) \leftrightarrow x\in A \cup B)$

即 $L(M) = A \cup B$. 证毕

正则语言的交是正则语言

定理: 设A,B都是Σ上的正则语言, 则$A\cap B$也是正则语言.

证明: 设$M_1=(Q_1,\Sigma,\delta_1,s_1,F_1)$和$M_2=(Q_2,\Sigma,\delta_2,s_2,F_2)$是DFA,

且 $L(M_1)=A$, $L(M_2)=B$,

令$Q=Q_1\times Q_2$, $s=(s_1,s_2)$, $F=F_1\times F_2$,

δ: $Q\times\Sigma\to Q$, $\forall a\in\Sigma$, $r_1\in Q_1$, $r_2\in Q_2$,

$\delta(\,(r_1,r_2)\,,\,a\,)=(\,\delta_1(r_1,a),\,\delta(r_2,a)\,)$,

即对$i=1,2$, 第i个分量按M_i的转移函数变化.

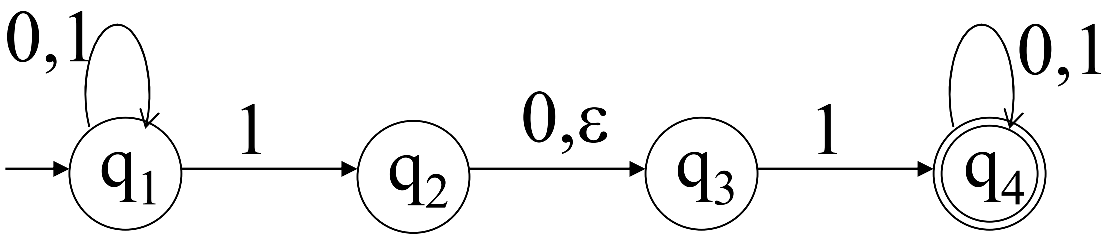
令$M=(Q,\Sigma,\delta,s,F)$, 则 $\forall x\,(\,x\in L(M)\leftrightarrow x\in A\cap B\,)$

即 $L(M)=A\cap B$. 证毕

证明特点: 构造性证明

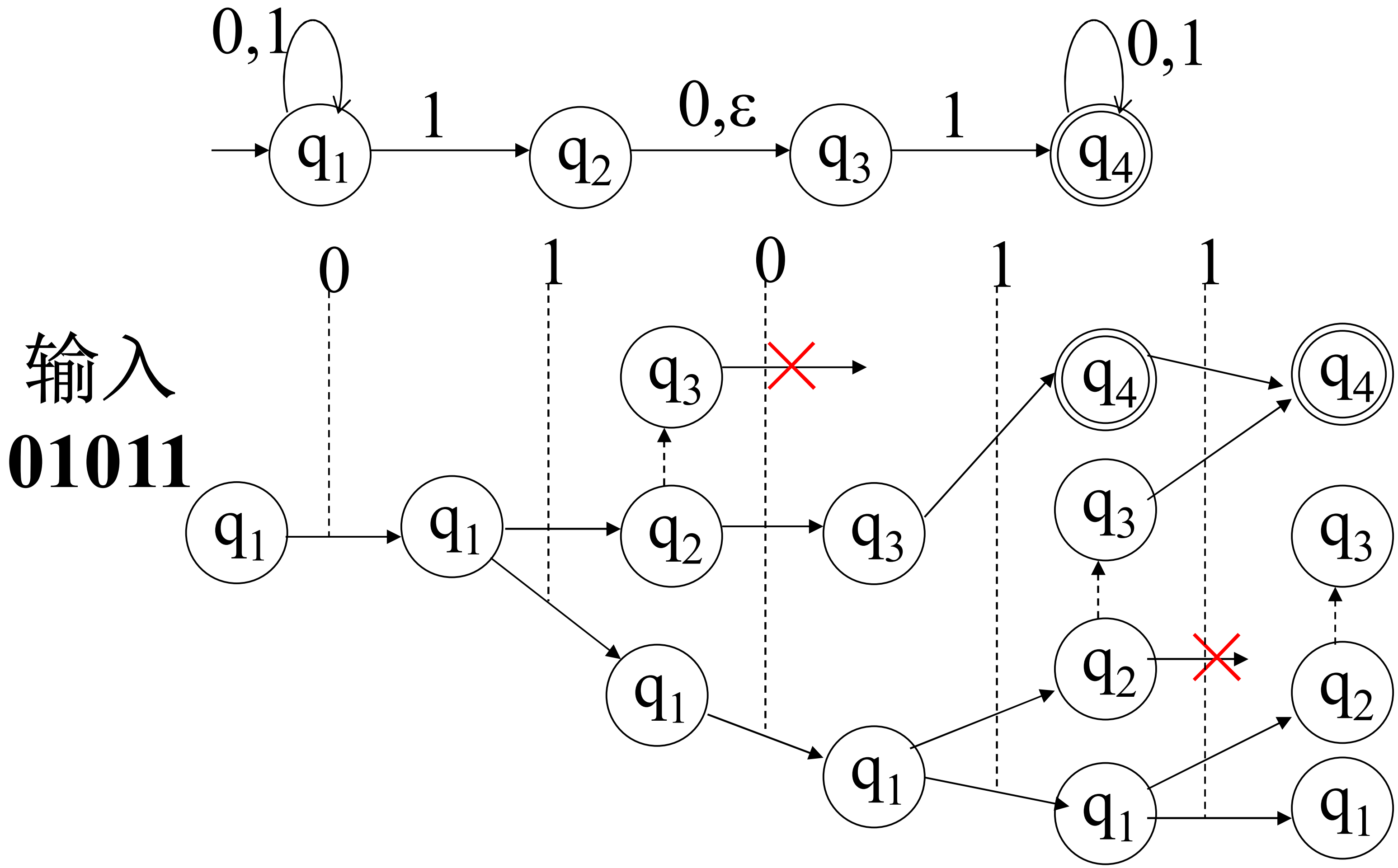
非确定型机器

现在引入非确定型有限自动机(**NFA**)

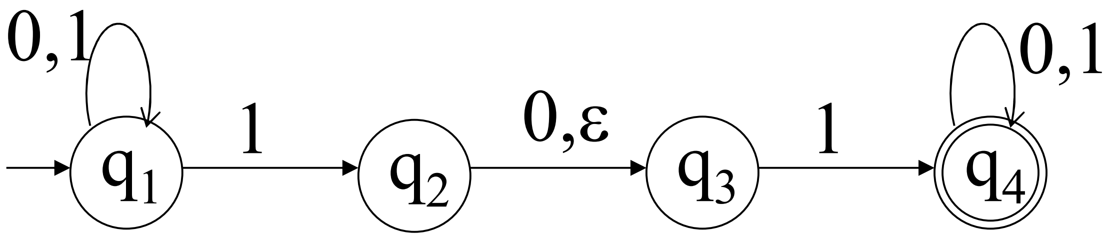


- 每步可以**0**至多种方式进入下一步
- 转移箭头上的符号可以是空串**ε,**
  表示不读任何输入就可以转移过去

非确定型计算



NFA的形式定义



定义: **NFA**是一个**5**元组**(Q,Σ,δ,s,F)**,

1) **Q**是状态集;

2) Σ是字母表;

3) δ: **Q**× $\Sigma_\varepsilon \rightarrow$ **P(Q)**是转移函数;

4) s∈**Q**是起始状态;

5) **F**⊆**Q**是接受状态集;

其中 $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

$\delta(q_1,1) = \{q_1,q_2\}$

$\delta(q_2,\varepsilon) = \{q_3\}$

$\delta(q_2,1) = \varnothing$

$\delta(q_1,\varepsilon) = \varnothing$

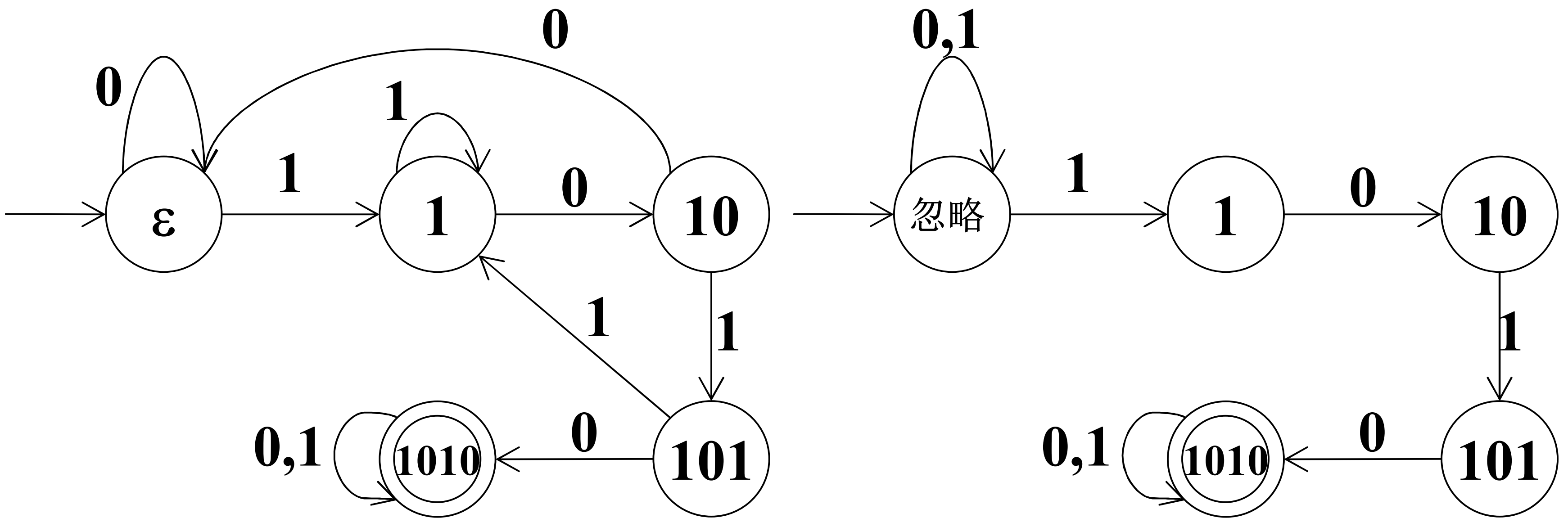
NFA的设计

- 自己即自动机
- 寻找需要记录的关键信息

NFA的设计

{ w∈{0,1}* | w含有子串1010 }

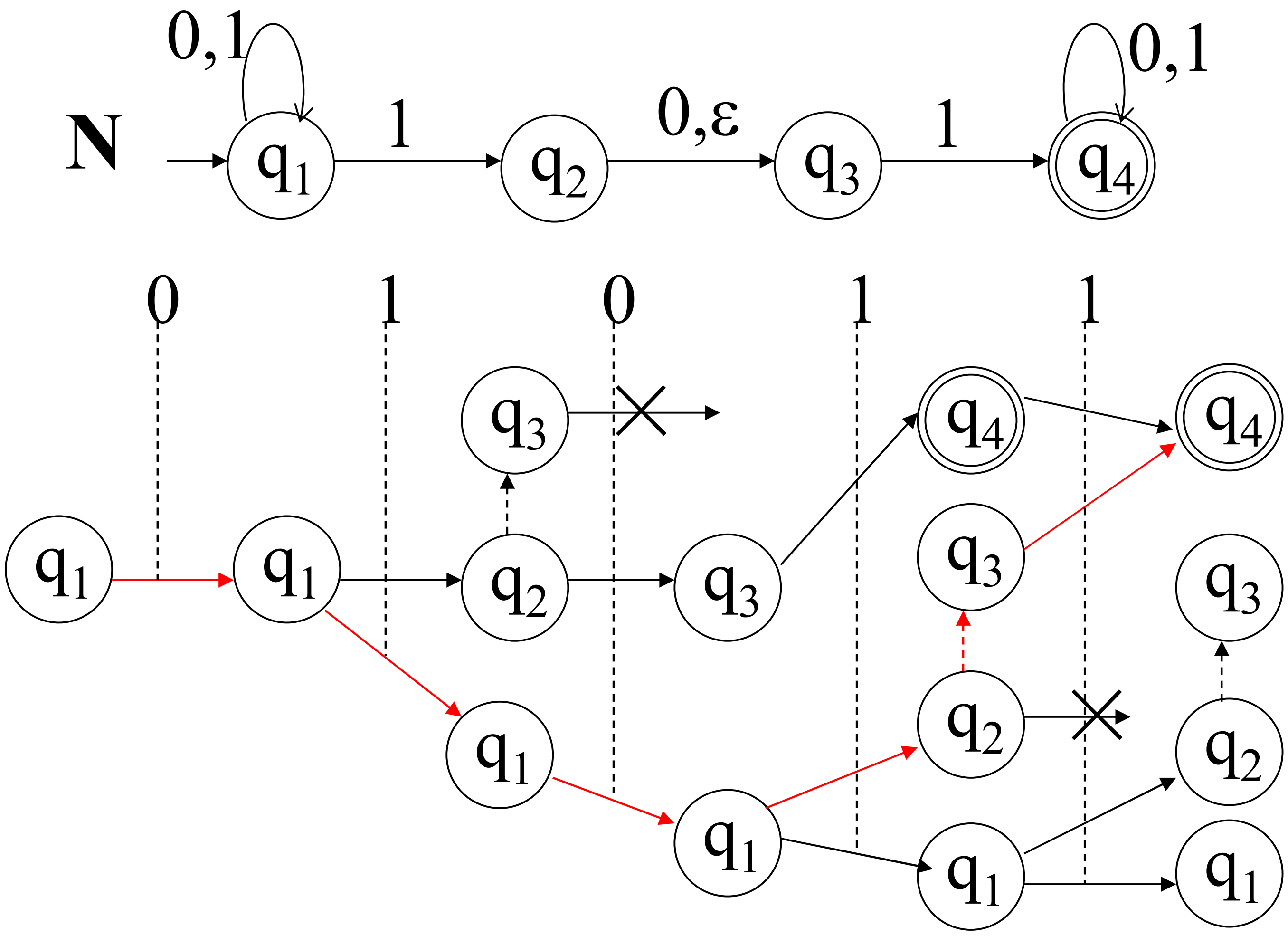
Σ={0,1}，关键信息：忽略(ε), 1, 10, 101, 1010



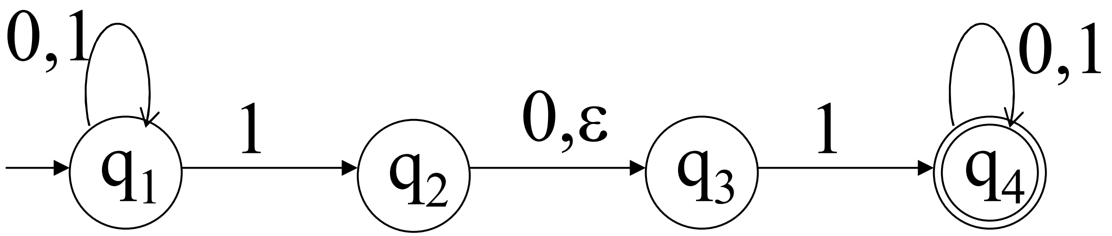
DFA

NFA

NFA的计算


N
0,1
q1
1
q2
0,ε
q3
1
q4
0,1
0
1
0
1
1
q1
q1
q2
q3
q1
q3
q1
q4
q3
q2
q1
q4
q3
q2
q1

每个NFA都有等价的DFA



以原状态的子集为新机器的状态

编号	δ	0	1
1	$\{q_1\}$ 1	$\{q_1\}$	$\{q_1, q_2, q_3\}$2
2	$\{q_1, q_2, q_3\}$	$\{q_1, q_3\}$3	$\{q_1, q_2, q_3, q_4\}$4
3	$\{q_1, q_3\}$	$\{q_1\}$	$\{q_1, q_2, q_3, q_4\}$
4*	$\{q_1, q_2, q_3, q_4\}$	$\{q_1, q_3, q_4\}$5	$\{q_1, q_2, q_3, q_4\}$
5*	$\{q_1, q_3, q_4\}$	$\{q_1, q_4\}$6	$\{q_1, q_2, q_3, q_4\}$
6*	$\{q_1, q_4\}$	$\{q_1, q_4\}$	$\{q_1, q_2, q_3, q_4\}$

正则运算的封闭性

定理: 正则语言对并运算封闭.

定理: 正则语言对连接运算封闭.

定理: 正则语言对星号运算封闭.

正则表达式

定义: 称**R**是一个正则表达式, 若**R**是

1) a, a$\in\Sigma$;

2) ε ;

3) $\varnothing$;

4) $(R_1 \cup R_2)$, R_1和R_2是正则表达式;

5) $(R_1 \circ R_2)$, R_1和R_2是正则表达式;

6) (R_1^*), R_1是正则表达式;

每个正则表达式**R**表示一个语言,记为**L(R)**.

例: 0^*10^*, $01 \cup 10$, $(\Sigma\Sigma)^*$, $1^*\varnothing$, $\varnothing^*$.

正则表达式与DFA等价

定理**2.3.1:** 语言**A**正则⇔**A**可用正则表达式描述.

A正则⇒A有正则表达式

构造广义非确定有限自动机(**GNFA**)

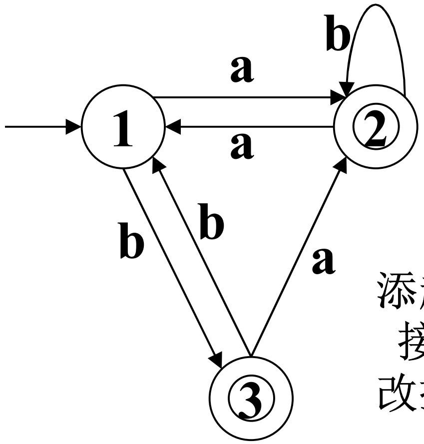
- 非确定有限自动机
- 转移箭头可以用任何正则表达式作标号

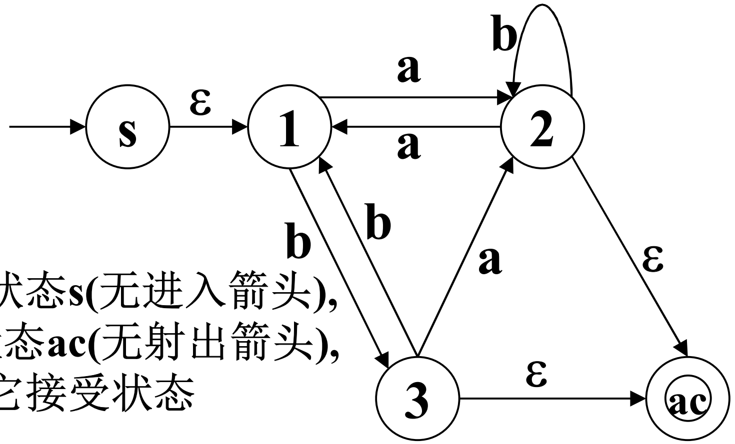
证明中的特殊要求:

- 起始状态无射入箭头.
- 唯一接受状态(无射出箭头).

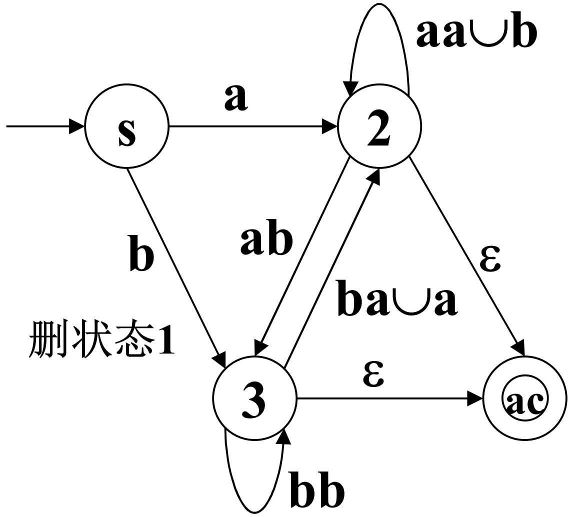
手段: 一个一个地去掉中间状态.

举例: A正则⇒A有正则表达式

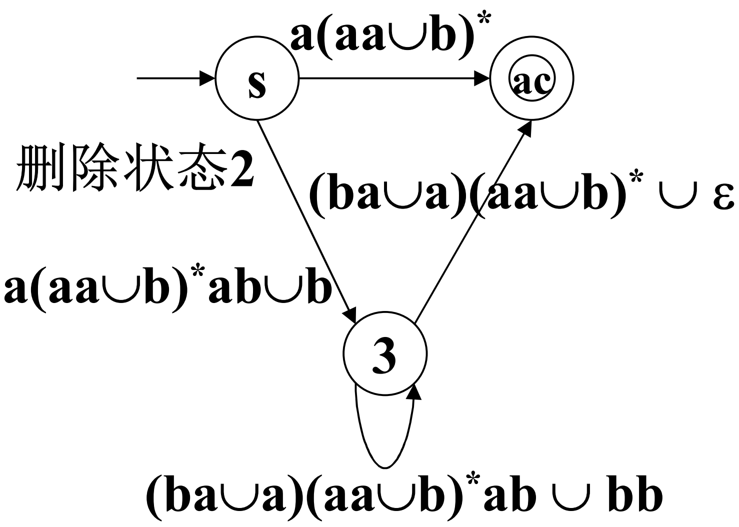




添起始状态s(无进入箭头),
接受状态ac(无射出箭头),
改掉其它接受状态

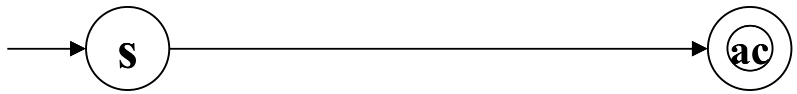


删状态1



删除状态2

$((a(aa\cup b)^*ab\cup b)((ba\cup a)(aa\cup b)^*ab \cup bb)^*$
$((ba\cup a)(aa\cup b)^*\cup\varepsilon)) \cup (a(aa\cup b)^*)$



删除状态3

非正则语言：泵引理的等价描述

定理(泵引理): 设A是正则语言,则存在$p>0$使得

对任意$w \in A$, $|w| \geq p$, 存在分割$w=xyz$满足

1) 对任意 $i \geq 0$, $xy^iz \in A$;

2) $|y|>0$;

3) $|xy| \leq p$.

若A是正则语言,	若$\forall p>0$
则$\exists p>0$	$\exists w \in A(|w| \geq p)$
$\forall w \in A(|w| \geq p)$	$\forall x,y,z(|y|>0, |xy| \leq p, w=xyz)$
$\exists x,y,z(|y|>0, |xy| \leq p, w=xyz)$	$\exists i \geq 0$,
$\forall i \geq 0$,	$xy^iz \notin A$.
$xy^iz \in A$.	则A非正则语言

非正则语言：$B = \{ 0^n 1^n \mid n \geq 0 \}$ 非正则

∵ $\forall p>0$,

令 $w=0^p1^p$,

$\forall x,y,z(|y|>0, |xy|\leq p, w=xyz)$

令 $i=0$,

$xz = 0^{p-|y|}1^p \notin B$

∴ B非正则语言

若 $\forall p>0$

$\exists w \in A(|w| \geq p)$

$\forall x,y,z(|y|>0, |xy|\leq p, w=xyz)$

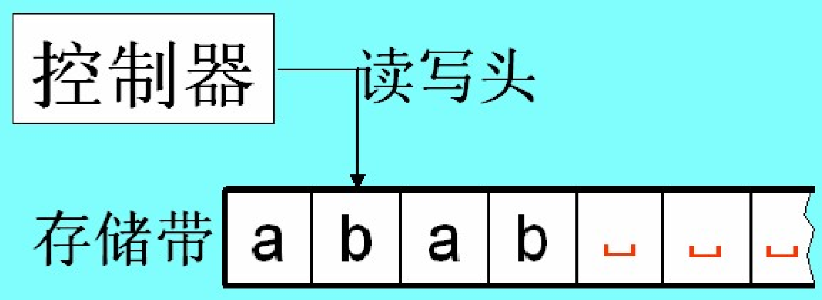
$\exists i \geq 0$,

$xy^iz \notin A$.

则A非正则语言

第3章 图灵机

图灵机(TM)的形式化定义



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␣∈Γ, Σ⊂Γ.

4) $\delta: Q\times\Gamma\to Q\times\Gamma\times\{L,R\}$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, $q_a \neq q_r$.

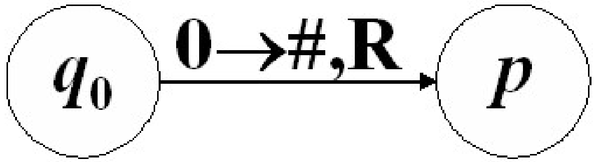
图灵机的运行

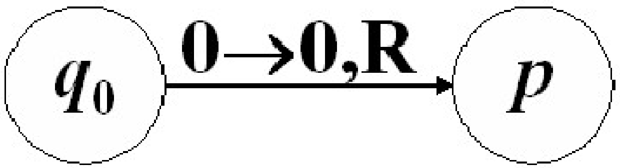
- 图灵机根据转移函数运行.

  例:设输入串为**0101**, 且$\delta(q_0,0)=(p,\#,R)$, 则有

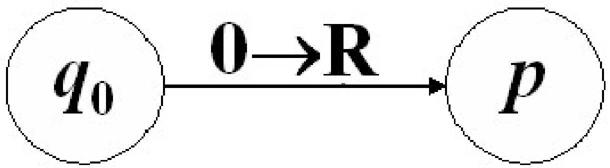
- 注: 若要在最左端左移, 读写头保持不动.

$\delta(q_0,0)=(p,\#,R)$的状态图表示:





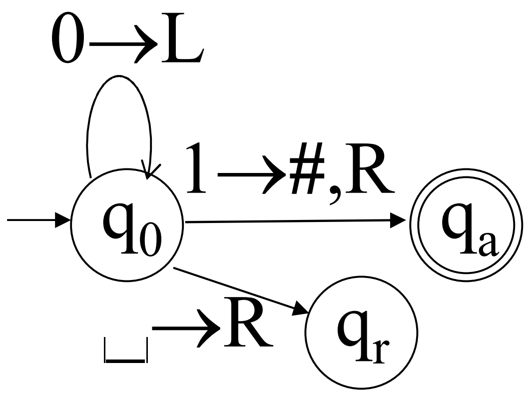
简记为



判定器与语言分类

- 图灵机运行的三种结果

  1. 若**TM**进入接受状态,则停机且接受输入,
  2. 若**TM**进入拒绝状态,则停机且拒绝输入,
  3. 否则**TM**一直运行,不停机.



- 定义: 称图灵机**M**为判定器,

  若**M**对所有输入都停机.

- 定义不同语言类:

  图灵可判定语言: 某个判定器的语言

  图灵可识别语言: 某个图灵机的语言,

图灵机的描述

(1) 形式水平的描述**(**状态图或转移函数**)**

(2) 实现水平的描述**(**读写头的移动**,**改写**)**

(3) 高水平描述**(**使用日常语言**)**

用带引号的文字段来表示图灵机. 例如:

M=“对于输入串**w,**

1) 若**w=**ε**,** 则拒绝**.**

2) 若只有**1**个**0,** 则接受**.**

3) 若**0**的个数为奇数**,** 则拒绝**.**

4) 从带左端隔一个**0,** 删一个**0.** 转**(2).”**

图灵机的变形

图灵机有多种变形:

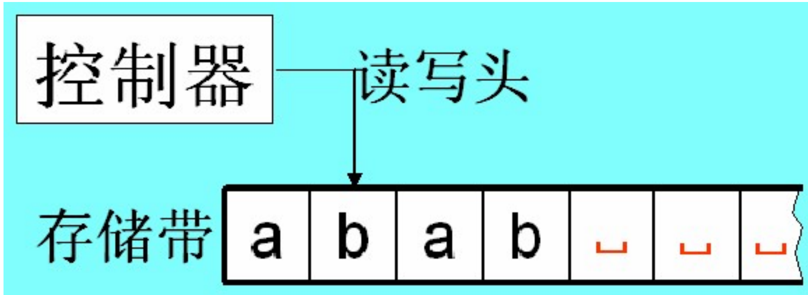
例如多带图灵机, 非确定图灵机

还有如枚举器, 带停留的图灵机等等

只要满足必要特征,

它们都与这里定义的图灵机等价.

非确定型图灵机(NTM)



- **NTM的转移函数**

  $\delta: Q\times\Gamma\rightarrow P(Q\times\Gamma\times\{L,R\})$

- **NTM转移函数举例**

  $\delta(q_3,0)=\{(q_2,x,R), (q_1,1,L), (q_3,\$,R)\}$

- 称**NTM M**接受**x**, 若在**x**上运行**M**时有接受分支.
- 称一**NTM**为判定的,

  若它对所有输入,所有分支都停机.

- 定理: 每个**NTM**都有等价的确定**TM**.
- 定理: 每个判定**NTM**都有等价的判定**TM**.

计算理论
第二部分 可计算理论

第**4**章 可判定性

定理:停机问题Halt是图灵可识别的

Halt={ <M,x> | 图灵机**M**在串**x**上会停机 **}**

证明**:** 构造识别**Halt**的图灵机**T,**

T = “ 对于输入**<M,x>, M**是图灵机**, x**是串

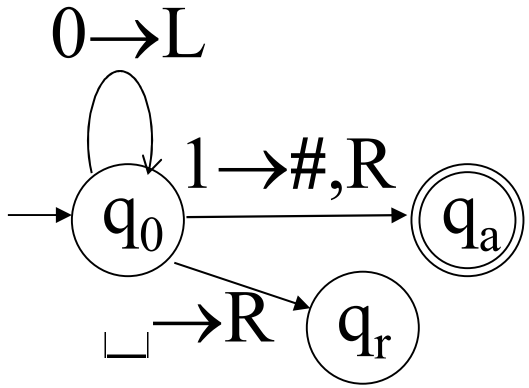
1. 在**x**上模拟**M,**

2. 若**M**停机**(**接受或拒绝**),** 则接受**.”**

T的语言是**Halt,** 证毕**.**

注：**T**不是判定器 **(?)**

例输入**<M,01>**



M

定理:停机问题Halt不可判定

Halt={ <M,x> | 图灵机M在x上会停机 }

证明: 假设Halt有判定器H, 构造图灵机D使用H:

Diagonal = “ 对于输入<M>, M是图灵机,

1. 在<M,<M>>上运行H,
2. 若H接受, 则返回1;
3. 若H拒绝, 则停机.”

- 在Diagonal上输入<Diagonal>”是否会停机?
- 若D停机, 即<D,<D>>∈HALT, H接受<D,<D>>, 则由2, D不停机
- 若D不停机, 即<D,<D>>∉HALT, H拒绝<D,<D>>, 则由3, D停机
- 矛盾, 所以H不存在.

定理: A_{TM}的补不是图灵可识别的

定理: 若A和A的补都是图灵可识别, 则A图灵可判定

证明: 设图灵机T和Q分别识别A和A的补, 构造R:

R= “ 对于输入x, x是串,

1. 在x上同步模拟T和Q, 直到有一个接受,
2. 若T接受x, 则接受; 若Q接受x, 则拒绝.”

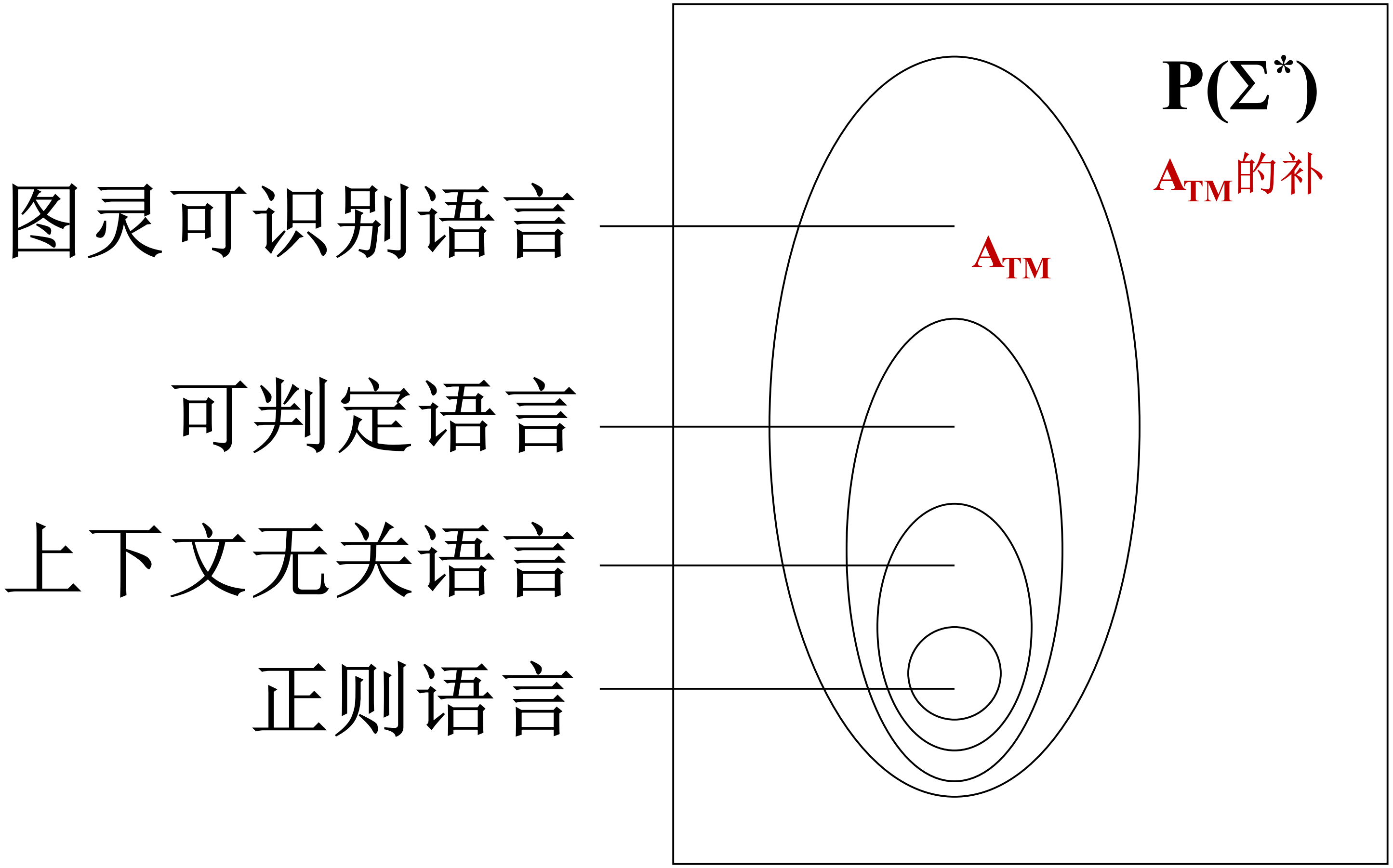
$x \in A \Rightarrow$ T接受x $\Rightarrow$ R接受x

$x \notin A \Rightarrow$ Q接受x $\Rightarrow$ R拒绝x

1. R是判定器 2. R的语言是A.

推论: A_{TM}的补不是图灵可识别的.

各语言类之间的关系



可判定性

成员测试:

A_{DFA}={<B,w>|B是DFA,w是串,B接受w} 可判定

A_{TM}={<M,w>|M是一个TM,且接受w} 不可判定

空性质测试: E_{DFA}={<A>|A是DFA,L(A)=∅} 可判定

等价性质测试:

EQ_{DFA}={<A,B>|A和B都是DFA,且L(A)=L(B)} 可判定

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

时间复杂性

- 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

分析算法

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: (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$O(n^2)$.

图灵机M_2

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.

×log n

5)若带上同时没有0和1,则接受. $O(n)$
否则拒绝."

总时间: O(nlogn)

单带与多带运行时间比较

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

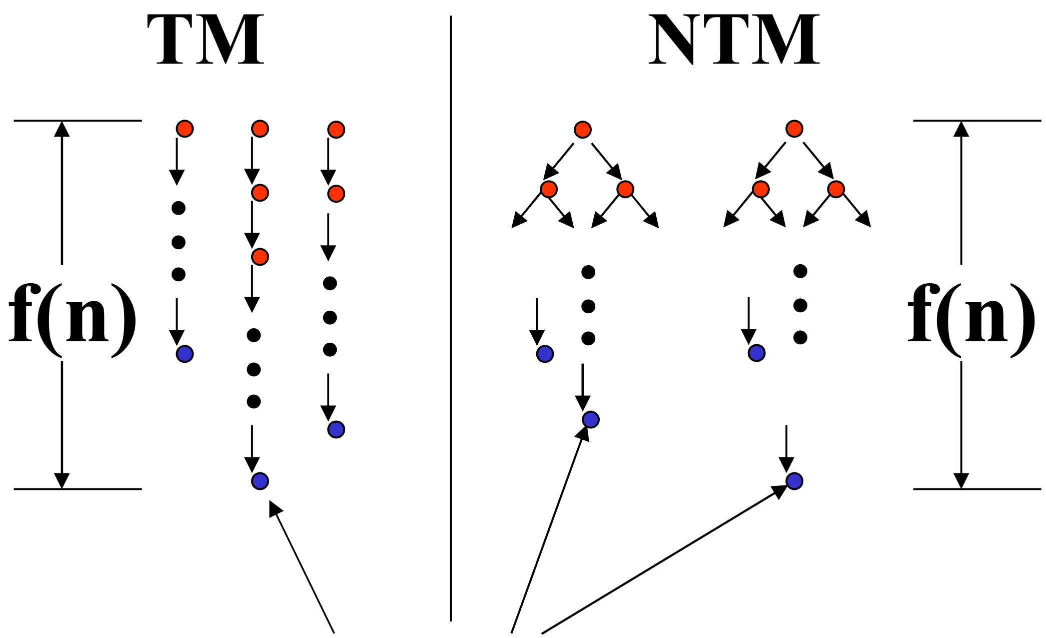
4) 如果两带上同时没有0和1,就接受."

定理:设函数t(n)≥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.

NTM的运行时间

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
时间t(n)NTM
都有一等价的
时间$2^{O(t(n))}$TM.

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

P类

定义:**P**是单带确定**TM**在

多项式时间内可判定的问题,即

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

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

NTIME(t(n))={L|L可被O(t(n))时间NTM判定.}

定义:**NP**是单带非确定**TM**在

多项式时间内可判定的问题,即

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

NP问题

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."

P与NP

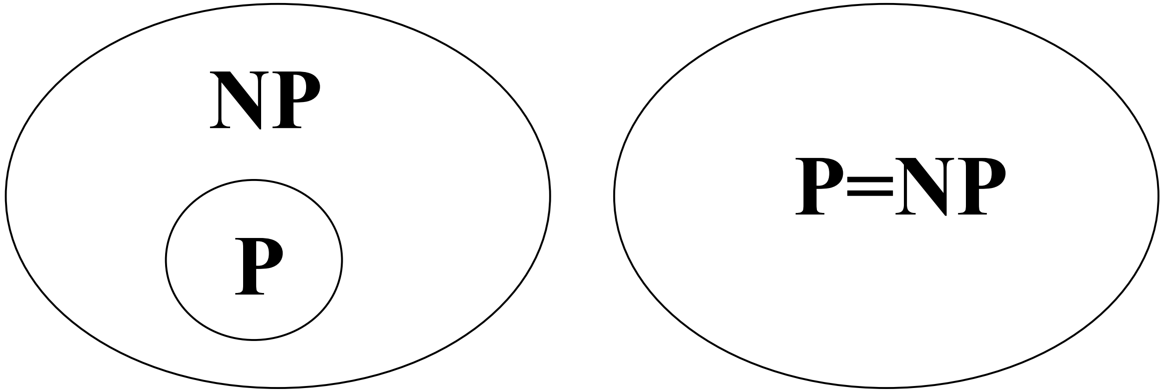
P=成员资格可以快速判定的语言类.

NP=成员资格可以快速验证的语言类.

显然有 $\mathbf{P} \subseteq \mathbf{NP}$

但是否有 $\mathbf{P} = \mathbf{NP}$?

看起来难以想象, 但是现在没有发现反例.



当代数学与理论计算机共同的难题.

可满足问题SAT

- 可满足性问题:

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

- 二元可满足性问题:

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

- 三元可满足性问题:

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

二元可满足问题2SAT∈P

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

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

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

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

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.

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

$x_3=0\to(x_4)\wedge(\neg x_4\vee\neg x_5)\to(\neg x_5)\to\varnothing$ 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.

3SAT$\in$NP

三元可满足性问题:

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

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.

多项式时间映射归约与C-L定理

- **Cook-Levin**定理: **SAT$\in$P $\Leftrightarrow$ P=NP.**
- 定义:多项式时间可计算函数$f:\Sigma^*\to\Sigma^*$.
- 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$A \le_P SAT$.
- 定理1: 若 $A \le_P B$, 且 $B\in P$, 则 $A\in P$.
- 注: 定理1说明, 若$SAT\in P$, 则 $NP = P$.

多项式时间映射归约的作用

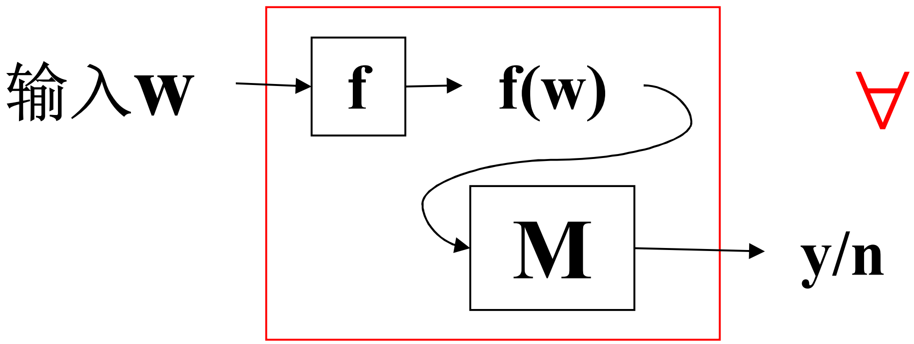
- 定理1: 若 $A \le_P B$, 且 $B \in P$, 则 $A \in P$.

  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.



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

利用f和B的判定器
构造A的判定器

定理: 3SAT $\leq_P$ CLIQUE

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

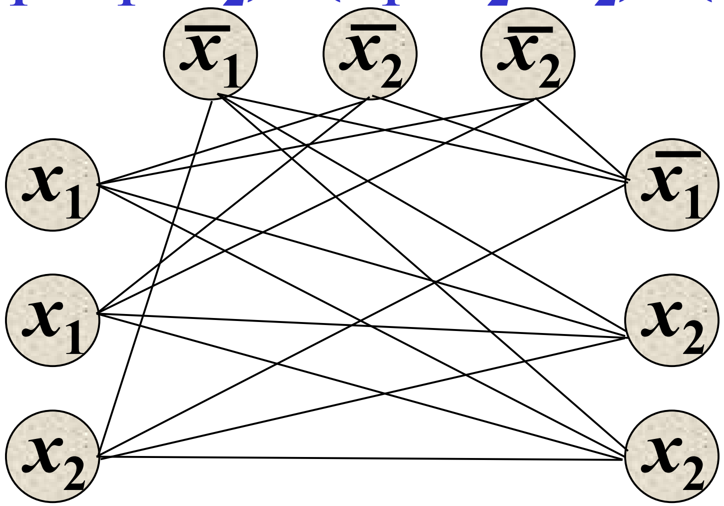
CLIQUE = $\{\langle G,k\rangle \mid G$是有k团的无向图$\}$.

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.

$f(\phi)=\langle G,k\rangle$, 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.

$f((x_1\vee x_1\vee x_2)\wedge(\overline{x_1}\vee\overline{x_2}\vee\overline{x_2})\wedge(\overline{x_1}\vee x_2\vee x_2))=\langle G,3\rangle$



需证:$\phi\in$**3SAT**

$\Leftrightarrow$

$(G,k)\in$CLIQUE

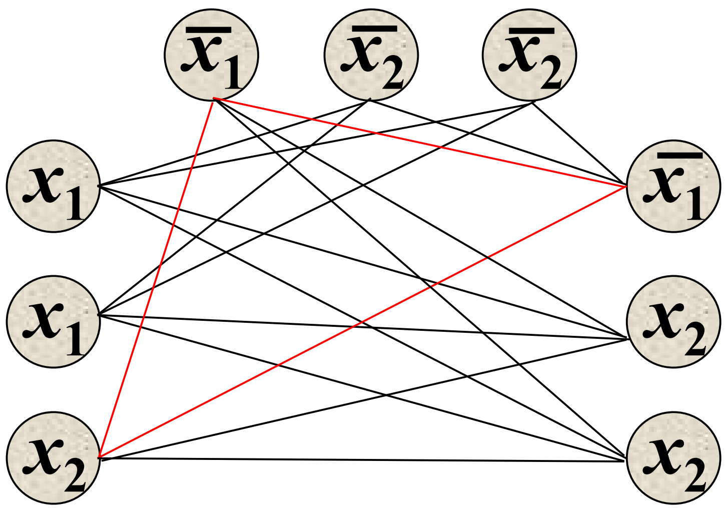
$\forall\phi,\ \phi\in 3SAT \Leftrightarrow f(\phi)\in CLIQUE$

$<\phi>(<(x_1\vee x_1\vee x_2)\wedge(\overline{x_1}\vee\overline{x_2}\vee\overline{x_2})\wedge(\overline{x_1}\vee x_2\vee x_2)>)\ \in$ **3SAT**

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

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

$\Leftrightarrow$ **f(ϕ) (<G,3>) ∈ CLIQUE.**



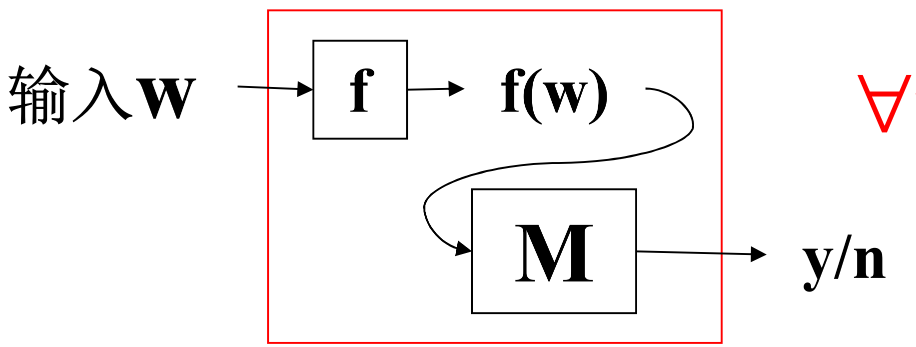
NP完全性

- 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$A \le_P B$.
- 定理1:若 $A \le_P B$, 且 $B \in P$, 则 $A \in P$.
- 定理2: 若B是NPC, 且$B \in P$, 则P=NP.
- 定理3: 若B是NPC, $B \le_P C$,且$C \in NP$, 则C是NPC.

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

- Cook-Levin定理: SAT是NP完全问题.
- 推论: 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

$\forall A \in NP$, 都有 $A \le_P SAT$

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

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

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

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

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

推论:3SAT是NP完全的

只需将前面的ϕ改造为**3cnf**公式.

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

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

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

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

降子句长度

其它NP完全问题

SAT

3SAT

HP(哈密顿路径) **KS(子集和, 01背包)**

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)

哈密顿路径(HP)是NPC($3SAT \le_p HP$)

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ϕ = $(a_1 \vee b_1 \vee d_1) \wedge \ldots \wedge (a_k \vee b_k \vee d_k)$,

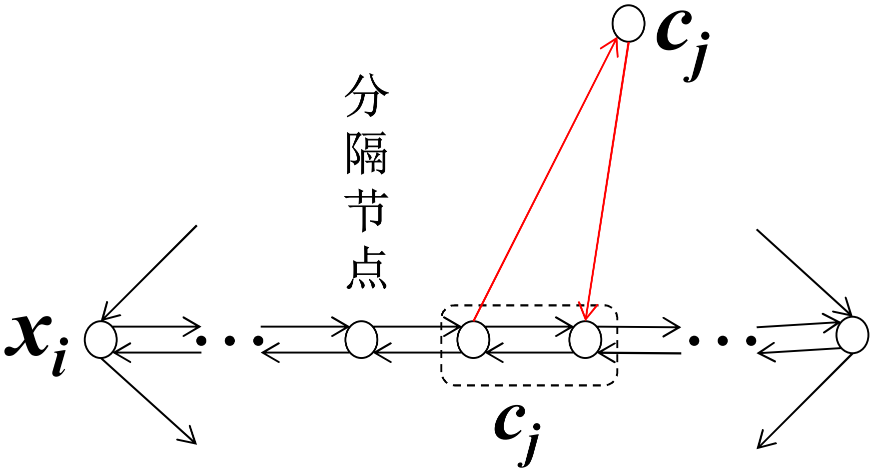
不妨设有k个子句$c_1, \ldots, c_k$, n个变量$x_1, \ldots, x_n$,

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

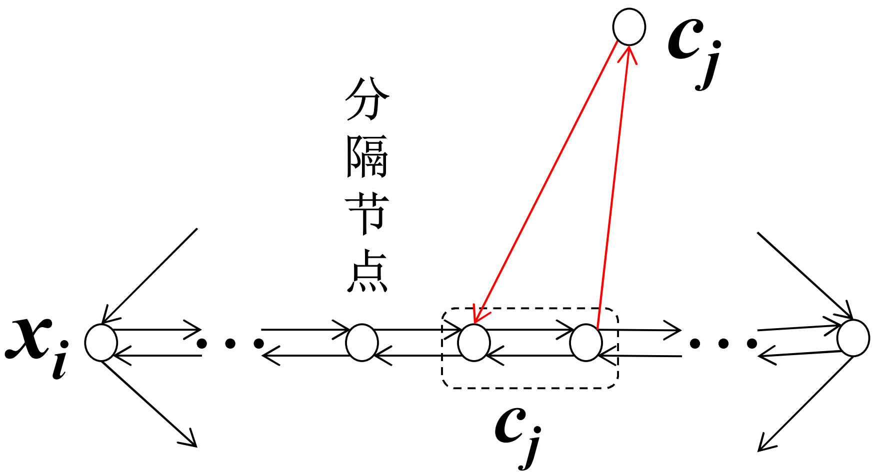
一般由**3cnf**公式构造图有

变量构件,子句构件, 联接构件

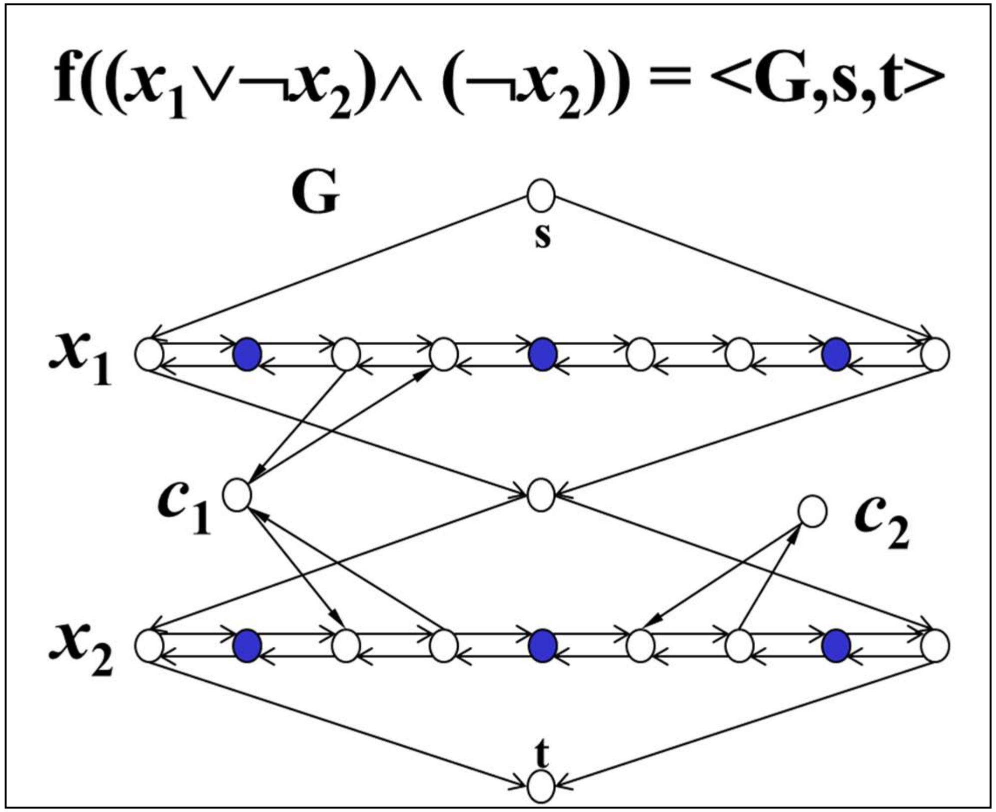
变量与子句构件的连接



当子句c_j含有文字x_i时添加的边



当子句c_j含有文字$\neg x_i$时添加的边



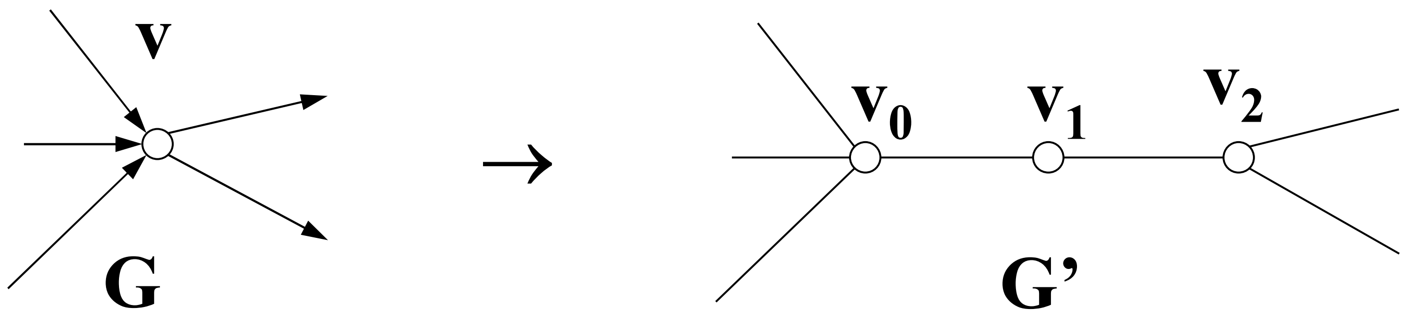
无向图的哈密顿路径

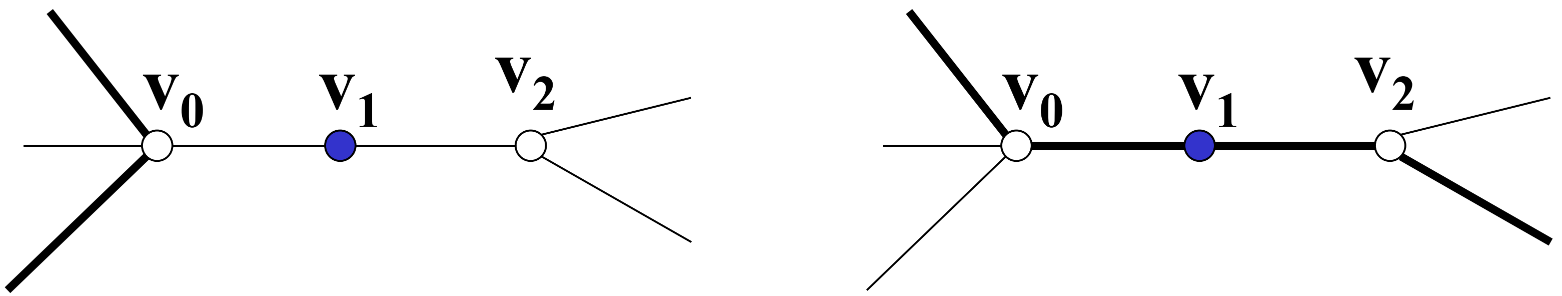
HP = {<G,s,t> | G是有从s到t哈密顿路径的有向图 }

UHP = {<G,s,t>| 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 **<G,s,t> $\rightarrow$ <G',s_2,t_0>**

s对应**s_2, t**对应**t_0,** 其它每个节点**v**对应**v_0,v_1,v_2,**





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** $\le_P$ **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.

$y_1,\ldots,y_n,z_1,\ldots,z_n,g_1,\ldots,g_k,h_1,\ldots,h_k,t$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≤**(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≤**3**个**1**

归约举例

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

	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

	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

单位阵

单位阵

单位阵

单位阵

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$\neg x_2$.

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, NPC**

计算理论基础第1章作业

1.1 下图给出了两台**DFA** M_1和M_2的状态图。

回答下述关于这两台机器的问题。

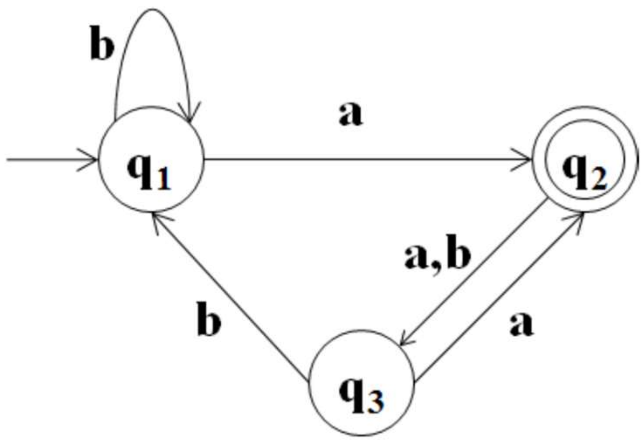
a. 它们的起始状态是什么?

b. 它们的接受状态集是什么?

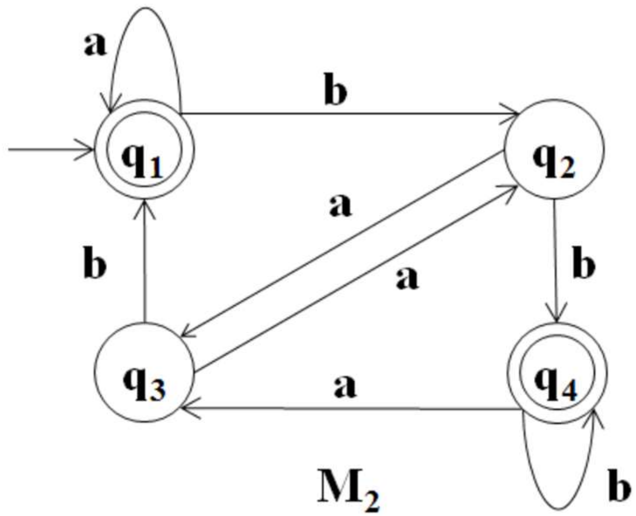
c. 对输入**aabb**，它们经过的状态序列是什么?

d. 它们接受字符串**aabb**吗?

e.它们接受字符串ε吗?



M_1



M_2

计算理论基础第1章作业

1.6 画出识别下述语言的**DFA**状态图. 字母表为**{0,1}**

d. { **w** | **w**的长度不小于**3**, 并且第**3**个符号为**0**};

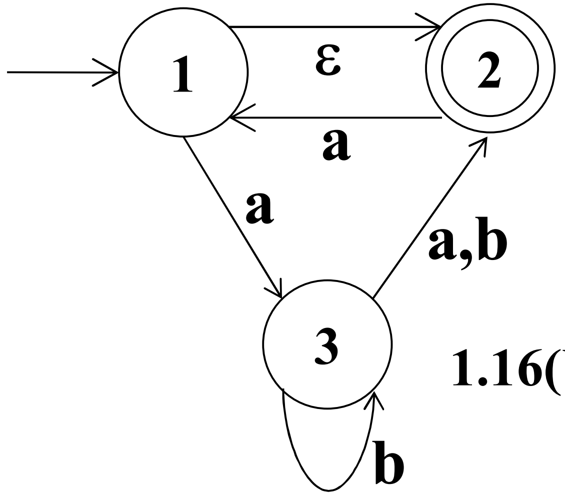
1.7. 给出下述语言的**NFA**, 并且符合规定的状态数.

字母表为**{0,1}**

e. 语言$0^*1^*0^*0$, **3**个状态.

计算理论基础第1章作业

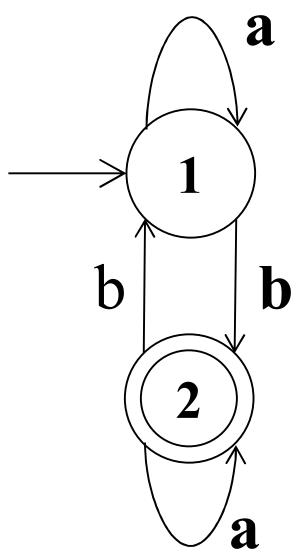
1.16(b) 将如右图的非确定有限自动机转换成等价的确定有限自动机.



1.16(b)题图

计算理论基础第1章作业

1.21(a) 将如右图的有限自动机转换成等价的正则表达式.



1.21(a)题图

1.29 使用泵引理证明下述语言不是正则的。

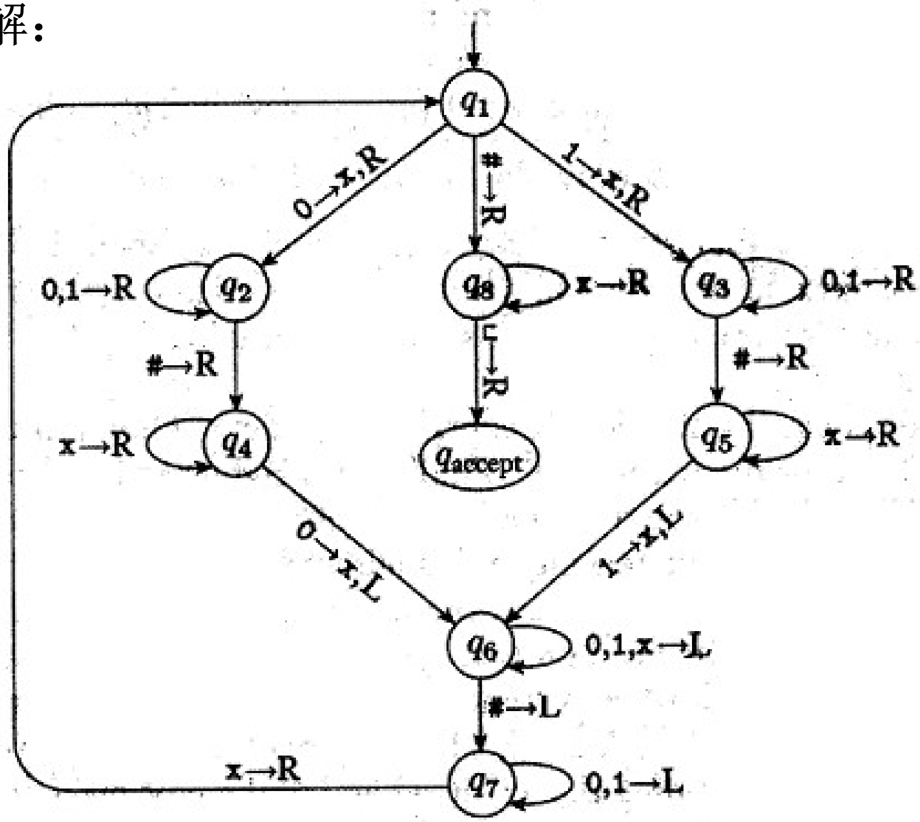
b. $A = \{ www \mid w \in \{a,b\}^* \}$

计算理论第3章作业

3.2 对于识别{w|w=u#u, u∈{0,1}*}的图灵机M_1 (见左图)，在下列输入串上，给出M所进入的格局序列.

c. 1##1， d. 10#11， e. 10#10

解：



补充说明: 没有画出的箭头指向拒绝状态, 假设这些箭头都不改写右移且q_r是拒绝状态.

计算理论第3章作业

3.8 下面的语言都是字母表{**0,1**}上的语言, 以实现水平的描述给出判定这些语言的图灵机:

b. B = {w|w所包含的**0**的个数是**1**的个数的两倍**}**

3.15b 证明图灵可判定语言类在连接运算下封闭.

3.16d证明图灵可识别语言类在交运算下封闭.

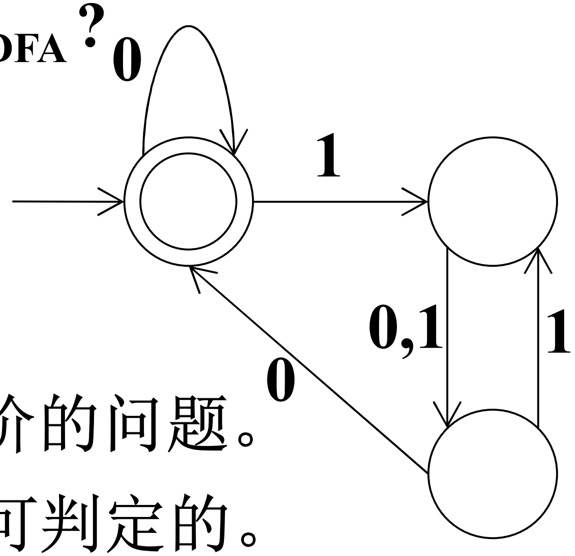
计算理论第4章作业

4.1 对于右图所示的**DFA M**, 回答下列问题，并说明理由

a. $\langle M,0100\rangle \in A_{DFA}$? **b.** $\langle M,011\rangle \in A_{DFA}$?

c. $\langle M\rangle \in A_{DFA}$?

e. $\langle M\rangle \in E_{DFA}$? **f.** $\langle M,M\rangle \in EQ_{DFA}$?



4.2 考虑一个**DFA**和一个正则表达式是否等价的问题。

将这个问题描述为一个语言并证明它是可判定的。

4.3 设 $ALL_{DFA} = \{\langle A\rangle \mid A$是一个识别$\Sigma^*$的DFA$\}$. 证明$ALL_{DFA}$可判定.

计算理论第7章作业

7.9 无向图中的三角形是一个**3**团。证明**TRIANGLE**$\in$**P**，其中**TRIANGLE={ <G> | G**包含一个三角形 **}**。

计算理论第7章作业

7.11 若图G的节点重新排序后，G可以变得与H完全相同，则称G与H是同构的。令ISO = {<G,H> | G和H是同构的图}。 证明ISO∈NP。

证明:构造如下非确定图灵机

N= "对于输入<G,H>, G和H都是图,

1) 若G和H顶点数不同, 则拒绝.
2) 设G的顶点为$x_1, x_2, \ldots, x_n$, H的顶点为$y_1, y_2, \ldots, y_n$.
3) 非确定的选择1到n的排列p.
4) 对 i = 1 到 n-1
5) 　对 j = i+1 到 n
6) 　　若 $(x_i,x_j) \in E(G)$ 异或 $(y_{p(i)},y_{p(j)}) \in E(H)$ 为真, 则拒绝
7) 接受."。

若G, H同构,则 N一定有分支接受; 否则, N所有分支拒绝.

N的所有分支都在都在$O(n^2)$时间内运行.

所以, N是ISO的多项式时间非确定判定器, ISO∈NP.

计算理论第7章作业

7.21 令**Double-SAT** = { <ϕ> | ϕ至少有两个满足赋值 }。证明**Double-SAT**是**NP**完全的。

证明：

(1) Double-SAT$\in$NP

构造如下非确定图灵机

N= "对于输入<ϕ>, ϕ是布尔公式,

 (a) 非确定地产生两组不同赋值**s,t**

 (b) 若既有在赋值**s**下ϕ=**1**, 又有在赋值**t**下ϕ=**1**, 则接受；否则，拒绝"

因为**N**的语言是**Double-SAT**, 且**N**在多项式时间内运行, 所以**Double-SAT$\in$NP.**

(2) 证明**SAT**可以多项式时间映射归约到**Double-SAT.**

对任意布尔公式ϕ, 添加一个新变量**a**, 构造函数**f**(ϕ) = $\phi \wedge (a \vee \neg a)$。

首先, **f**可在多项式时间内计算完成。

其次, **f**是**SAT**到**Double-SAT**的映射归约, 即ϕ可满足$\Leftrightarrow$**f**(ϕ)有两个满足赋值:

若ϕ有可满足赋值**s**, 则在赋值**s**和**a=1**下**f**(ϕ)=**1**, 在赋值**s**和**a=0**下**f**(ϕ)=**1**, 从而有两个不等赋值; 若**f**(ϕ)有可满足赋值**s**, 则从**s**中去掉**a**的赋值, 必然也是ϕ的可满足赋值. 所以**f**是从**SAT**到**Double-SAT**的多项式时间映射归约。

由**(1)**和**(2)** 及**SAT**是**NP**完全问题，**Double-SAT**是**NP**完全问题。

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$\in$**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**$\in$**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。