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.

问题与决定性问题

- 决定性问题(**Dicision Prob**): 只需回答是与否的问题
  “一数是否是偶数”, “串长度是否是**2**的幂次”
  “图是否连通”, “图是否有**k**团”
- 一般问题: 排序, 最大团问题
- 本书只研究决定性问题
  1. 决定性问题能统一描述
  2. 一般问题总能转化为决定性问题
- 例: 最大团问题如何转化为决定性问题?

“最大团”与“图是否有k团”

- 团: 完全子图, 即所有节点对都有边相连的子图.
- 两个问题目前都没有快速算法
- 若“最大团”有快速算法, 则“图是否有k团”也有:

  对图G运行最大团算法, 得最大团的节点数m

  若m≥k, 则有k团; 否则没有k团.
- 若“图是否有k团”有快速算法, 则“最大团”也有:

利用“图是否有k团”二分搜索最大团节点数m.

决定性问题与字符串集合

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 }

“串长度是否是2的幂次” ---{ 0^{2^n} : $n\geq 0$ }

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

其中<G>是图G编码成的字符串.

“图是否有k团”------------------{<G> | 图G有k团}

给定有限字母表Σ, 例如{0,1}

- 每个输入是一个01串, 任意01串都可以是输入
- 决定性问题一一对应字符串集合

Σ^*的字典序与Σ^N

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.

Σ^*	ε	0	1	00	01	10	11	000	001	…
B	×	**0**	×	**00**	**01**	×	×	**000**	**001**	…
f(B)	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	…

等势, 可数, 不可数

- 等势: 若两集合间存在一一对应, 则称它们等势
- 可数: 若集合与有限集或与自然数集等势

  或者说集合元素可以按次序列出
- 不可数: 若集合不是可数的

  或者说集合元素不能按次序列出
- 自然数集可数, 正偶数集可数, $\{0,1\}^*$可数

可数集合举例

- 正有理数集可数

n	1	2	3	4		5	6	7	8	…
f(n)=p/q	1/1	2/1	1/2	3/1	2/2	1/3	4/1	3/2	2/3	…
p+q	2	3	3	4	4	4	5	5	5	…

正有理数集={p/q}, 其中p,q是互素的自然数.

- 给定字母表{0,1}, $\{0,1\}^*$可数.

{0,1}上所有有限长字符串的字典序排列:

ε,0,1,00,01,10,11,000,…

定理 $\{0,1\}^N$不可数

$\{0,1\}^N$是全体无限长的01串

证明: 假设$\{0,1\}^N$可数, 即可以排成一列($f(i)$)

按下面方法在$\{0,1\}^N$中取一点 x ,

x的第i位与$f(i)$的第i位相反

n	f(n)
1	1 1 1 0 1 …
2	0 0 0 0 0 …
3	0 1 1 1 1 …
4	1 1 1 0 0 …
…	…
x	0 1 0 1 …

x与列表每个数不同

x不在列表中

所以$\{0,1\}^N$不可数.

计算理论研究对象:语言

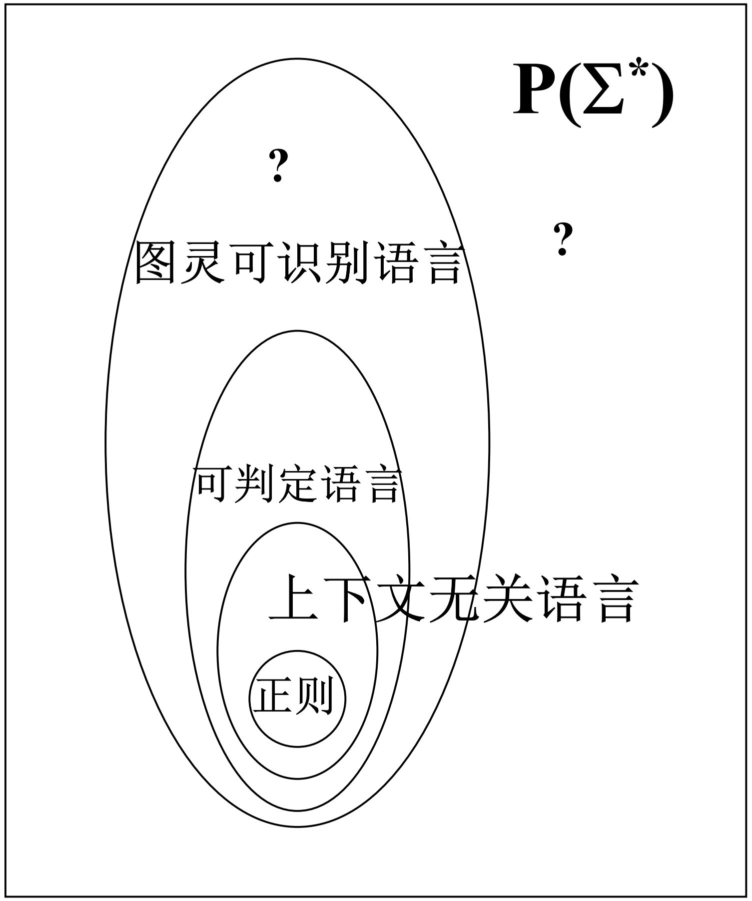
- 等势: 若两集合间存在一一对应, 则称它们等势
- 可数: 若集合与有限集或与自然数集等势
- 不可数: 若集合不是可数的
- 全体程序是$\{0,1\}^*$ 的子集, 至多可数
- 全体决定性问题与$\{0,1\}^N$ 等势, 不可数
- 程序可数, 问题不可数
- 数学的研究对象有数, 函数, 函数空间等
- 计算理论的研究对象: 问题 即 语言 即 字符串集合

计算理论
第二部分 可计算理论

第4章 可判定性

可判定=有算法

- **Halt** 图灵可识别 非图灵可判定
- **Halt的补** 非图灵可识别
- 可判定问题举例
- 不可判定问题举例



Church-Turing 论题

1930's人们开始考虑算法的精确定义

- **1933, Kurt Gödel,** 递归函数
- **1936, Alonzo Church, λ-calculus**
- **1936, Alan Turing,** 判定图灵机**(**判定器**)**
- **Church** 和 **Turing** 证明这三种定义等价
- 计算机能力的极限
- 即使未来几年量子计算机制造成功,
  人们能解决的问题类并不会变大

一些自然构造的问题

停机问题:

Halt = { <M,x> | 图灵机M在串x上会停机 }

成员测试:

A_{DFA} = {<B,w>|B是DFA,w是串,B接受w}

A_{CFG} = {<B,w>|B是CFG,w是串,B派生w}

A_{TM} = {<M,w>|M是一个TM,且接受w}

空性质测试:

E_{DFA} = {<A>|A是DFA,L(A)=∅}

E_{CFG} = {<G>|G是CFG,L(G)=∅}

等价性质测试:

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

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

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

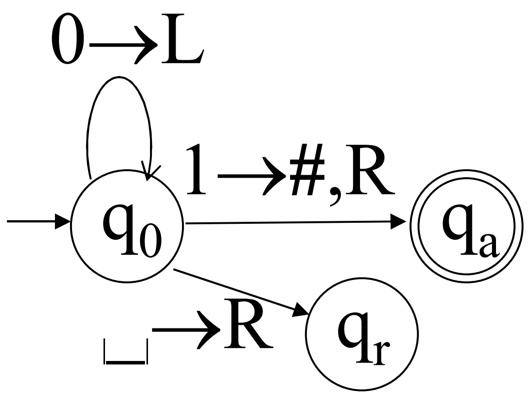
1. 在x上模拟M,

2. 若M停机(接受或拒绝), 则接受.”

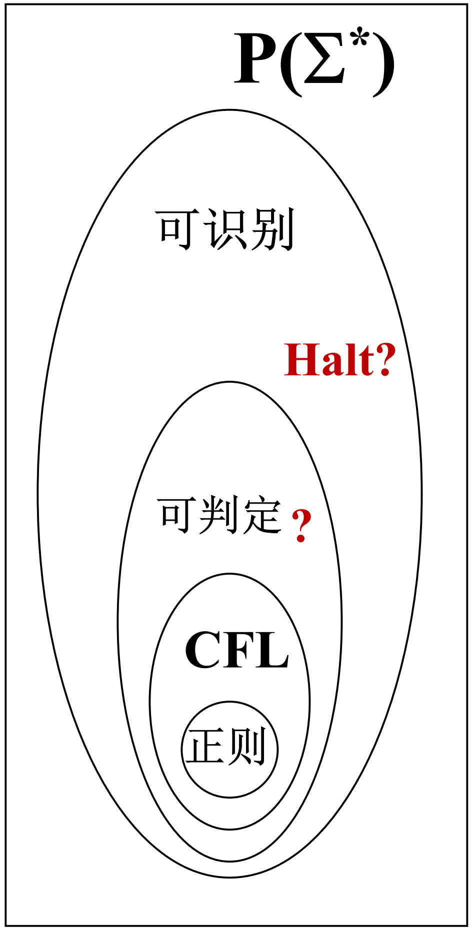
T的语言是Halt, 证毕.

注：T不是判定器 (?)

例T上运行<R,01>



R



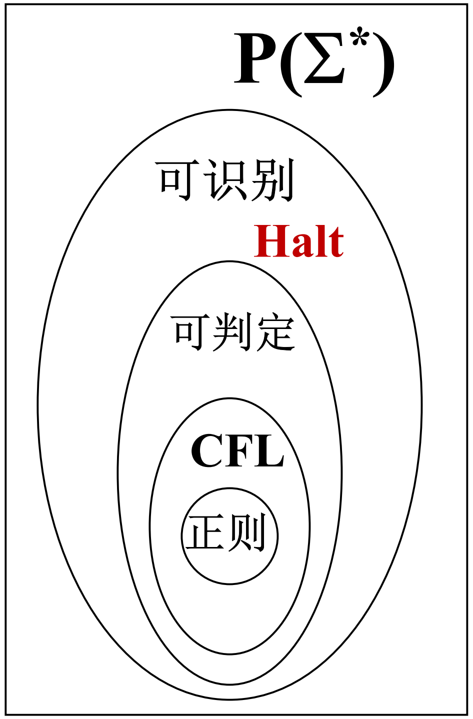
定理:停机问题Halt不可判定

Halt={ <M,x> | 图灵机M在串x上会停机 }

证明: 假设Halt有判定器H, 构造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上输入串<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.

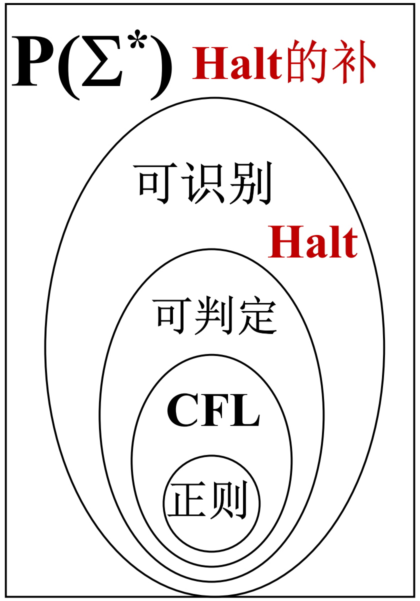
定理: Halt的补不是图灵可识别的

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, 直到有一个停机,
2. 若T接受x, 则接受x;
3. 若Q接受x, 则拒绝x.”



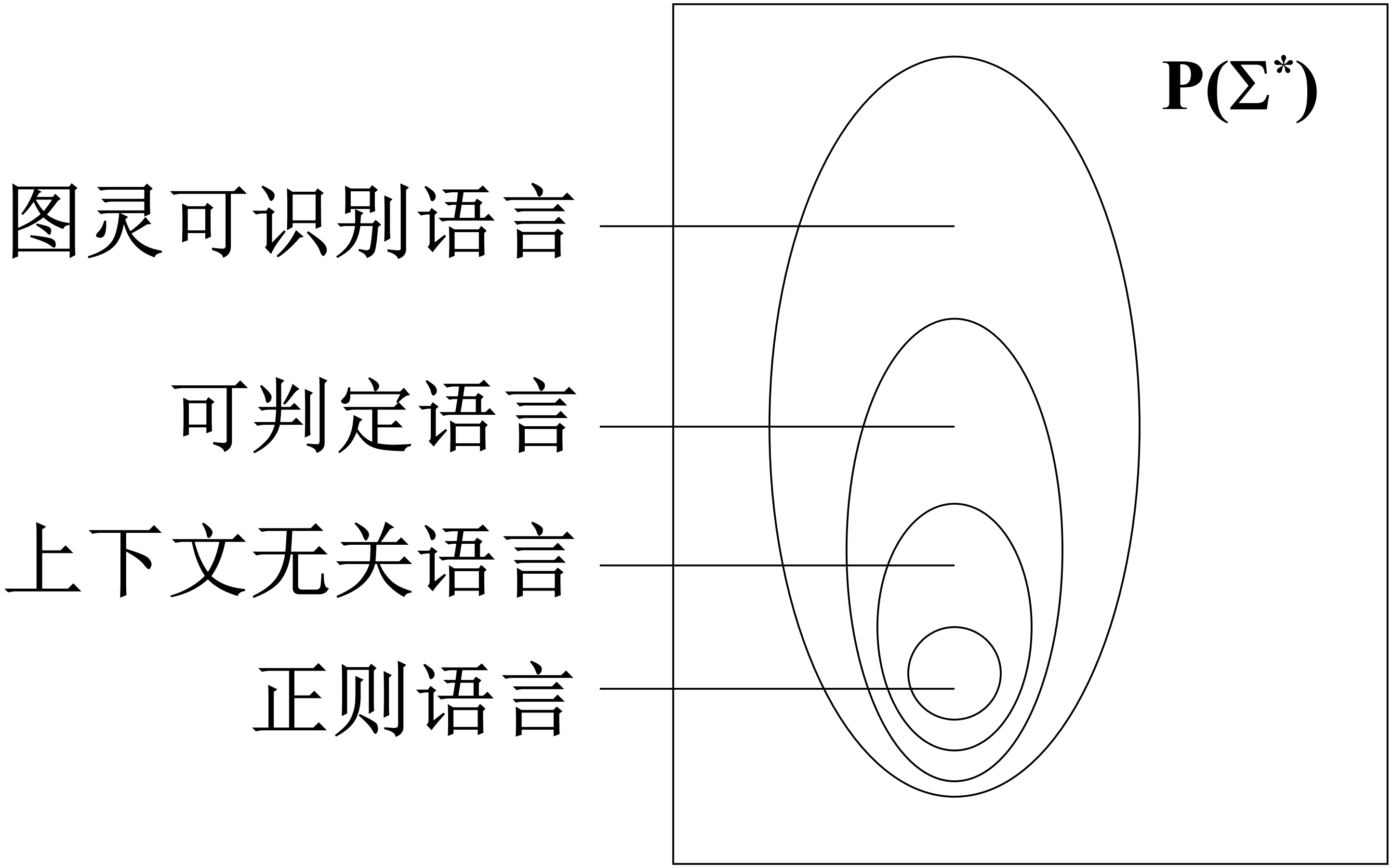
R是判定器, R的语言是A.

$\forall x \in A \Rightarrow$ T一定停机接受 $\Rightarrow$ R停机接受

$\forall x \notin A \Rightarrow$ Q一定停机接受 $\Rightarrow$ R停机拒绝

推论: 停机问题Halt的补不是图灵可识别的.

各语言类之间的关系



可判定性

停机问题:

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

成员测试:

A_{DFA} = {<**B**,**w**>|**B**是**DFA**,**w**是串,**B**接受**w**} 可判定

A_{CFG} = {<**B**,**w**>|**B**是**CFG**,**w**是串,**B**派生**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

E_{CFG} = {<**G**>|**G**是**CFG**,**L**(**G**)=∅} 可判定

E_{TM} = {<**M**>|**M**是**TM**,**L**(**M**)=∅} 不可判定

等价性质测试:

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

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

A_{DFA}={<B,w>|DFA B接受串w}可判定

证明:如下构造A_{DFA}的判定器:

M="对于输入<B,w>,其中B是DFA,w是串:

1)在输入w上模拟B.

2)如果模拟以接受状态结束,则接受;
如果以非接受状态结束,则拒绝."

$L(M) = A_{DFA}$. 将B视为子程序或实现细节:

- 检查输入. ((p,q,…)(a,…)((p,a,q),…)(q_0)(F), w)
- 模拟. 初始,B的状态是 q_0,读写头位于w的最左端,
  状态的更新由B的转移函数决定.

A_{NFA}={<B,w>|NFA B接受串w}可判定

思路1: 直接模拟NFA?

思路2: 先将NFA转换成DFA.

证明:如下构造A_{NFA}的判定器:

N="在输入<B,w>上,其中B是NFA,w是串:

1)将NFA B转换成一个等价的DFA C.

2)在输入<C,w>上运行A_{DFA}的判定器M.

3)如果M接受,则接受,否则拒绝."

运行TM M: M作为子程序加进N的设计中.

$L(N) = A_{NFA}$.

空性质测试

定理: E_{DFA}={<A>|A是DFA,L(A)=∅}可判定.

证明: 若A为一个DFA, 则

L(A)≠∅ ⇔ 存在从起始状态到某接受状态的路径.

T="对于输入<A>, 其中A是一个DFA:

1)标记起始状态.

2)重复下列步骤,直到没有新标记出现.

3) 对任一未标记状态, 若有从已标记状态到它的转移, 则将它标记.

4)如果无接受状态被标记,则接受;否则拒绝."

L(T) = E_{DFA} .

TM成员测试A_{TM}

A_{TM}={<M,w>|M是一个TM,且接受w}

定理 A_{TM}是不可判定的.

命题 A_{TM}是图灵可识别的.

U="对于输入<M,w>,其中M是TM,w是串:

1) 在输入w上模拟M;

2) 若M进入接受状态,则接受;

   若M进入拒绝状态,则拒绝."

L(U) = A_{TM} .

注:若M在w上不停机,则U在<M,w>上不停机.

定理 A_{TM}不可判定

A_{TM}={<M,w>|M是一个TM,且接受w}

证明:假设A_{TM}可判定, 且设H是其判定器, 构造

D="对于输入<M>,其中M是TM:

1)在串<M, <M>>上运行H.

2)若H接受(<M, <M>>), 则D拒绝(<M>);

若H拒绝(<M, <M>>), 则D接受(<M>)."

$<D> \in L(D)$ (?)

$\Leftrightarrow \quad <D,<D>> \in A_{TM}$

$\Leftrightarrow$ H接受<D,<D>>

$\Leftrightarrow$ D拒绝 <D>

$\Leftrightarrow \quad <D> \notin L(D)$

矛盾, 所以H不存在.

定理 A_{TM}不可判定
A_{TM}={<M,w>|M是一个TM,且接受w}

证明:假设A_{TM}可判定, 且设H是其判定器, 构造

D=“对于输入<M>,其中M是TM:

1)在串<M, <M>>上运行H.

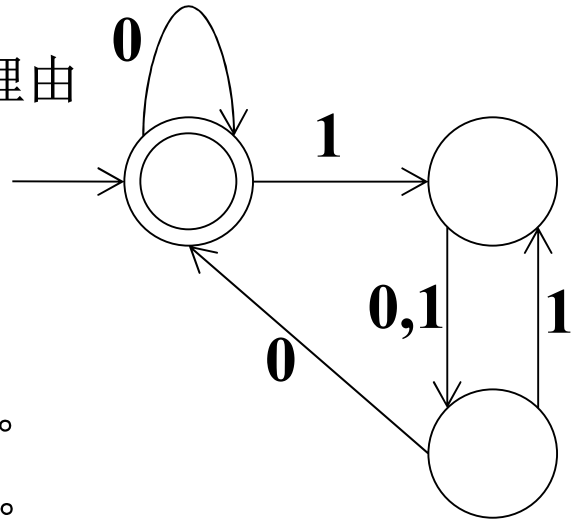
2)若H接受(<M, <M>>), 则D拒绝(<M>);

若H拒绝(<M, <M>>), 则D接受(<M>).”

	<M_1>	<M_2>	<M_3>	<M_4>	…	<D>	…
M_1	accept	reject	accept	reject	accept		
M_2	accept	accept	accept	accept	…	accept	…
M_3	reject	reject	reject	reject		reject	
M_4	accept	accept	reject	reject		accept	
⋮		⋮			⋱		
D	reject	reject	accept	accept		?	
⋮		⋮					⋱

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.

4.15 设$A = \{\langle R\rangle \mid R$是一个正则表达式，

其所描述的语言中至少有一个串w以111为子串$\}$.

证明A是可判定的。

不可判定问题举例

Hilbert第十问题**: "**多项式是否有整数根**"**有没有算法**?**

1970's 被证明不可判定 **(**没有判定器**,** 即没有算法**)**

M = "对于输入 **"p", p**是**k**元多项式**,**

1. 取**k**个整数的向量**x (** 绝对值和从小到大 **)**

2. 若**p(x) = 0,** 则停机接受**.**

3. 否则转**1. "**

这个图灵机对输入 **p(x,y) =** x^2+y^2-3不停机

对比: 一个可判定问题

一元多项式是否有整数根?

M = “对于输入 “**p**”, **k**次**1**元多项式**p(x)**,

1. 计算解的绝对值上界**N**

2. 对所有|**x**|≤**N**

3. 若**p(x) = 0**, 则停机接受.

4. 停机拒绝. ”

3.21 设多项式 $c_1x^n + c_2x^{n-1} + \ldots + c_nx + c_{n+1}$ 有根 $x = x_0$, c_{max} 是 c_i 的最大绝对值. 证明 $|x_0| < (n+1)\, c_{max} / |c_1|$

解: 不妨设 $c_1 \neq 0$.

若 $|x_0| \le 1$, 则 $|x_0| \le 1 \le c_{max} / |c_1| \le (n+1)\, c_{max} / |c_1|$, 性质成立

若 $|x_0| > 1$, 则由 $c_1x_0^n + c_2x_0^{n-1} + \ldots + c_nx_0 + c_{n+1} = 0$, 得

$$c_1x_0^n = -(c_2x_0^{n-1} + \ldots + c_nx_0 + c_{n+1}),$$

$$|c_1|\,|x_0|^n < (n+1)c_{max}|x_0|^{n-1},$$

$$|x_0| < (n+1)\, c_{max} / |c_1|.$$

例如: $2x^3 + 3x^2 - 7x + 11 = 0$