HIT
CS&E

第二章 数学基础

张炜
计算机科学与技术学院

提纲

2.1 计算复杂性函数的阶

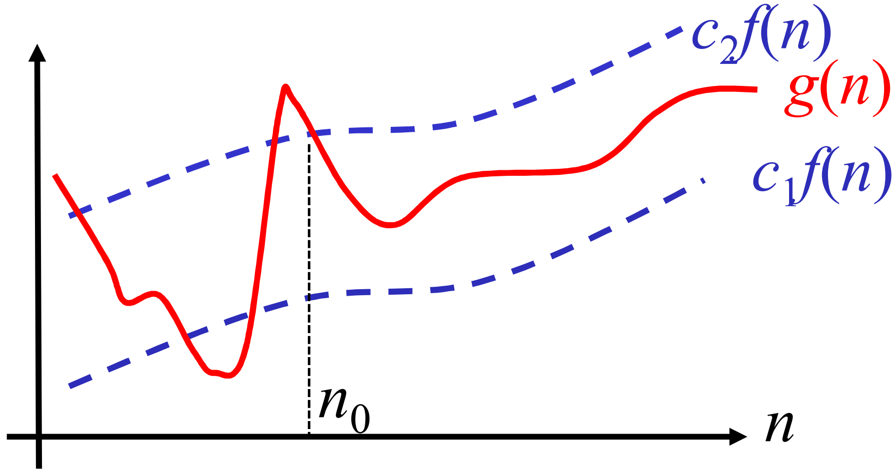
2.2 递归方程

2.1 计算复杂性函数的阶

2.1.1 同阶函数集合
2.1.2 低阶函数集合
2.1.3 高阶函数集合
2.1.4 严格低阶函数集合
2.1.5 严格高阶函数集合
2.1.6 函数阶的性质

2.1.1 同阶函数集合

定义2.1.1 (同阶函数集合) $\theta(f(n))=\{g(n) \mid \exists c_1, c_2>0,\ n_0,\ \forall n>n_0,\ c_1 f(n)\le g(n) \le c_2 f(n)\}$ 称为与$f(n)$同阶的函数集合。



- 若$g(n)\in\Theta(f(n))$, 则称$g(n)$与$f(n)$同阶
- $g(n)\in\Theta(f(n))$常记为$g(n)=\Theta(f(n))$
- $f(n)$是极限非负的，否则$\Theta(f(n))$定义为空集
  即：$f(n)$在n充分大之后必取非负值

HIT
CS&E

Example

例1 证明: $f(n)=an^2+bn+c=\Theta(n^2)$ $(a>0)$

证明: 令$c_1=a/4$, $c_2=7a/4$, $n_0=2\cdot\max\{|b|/a, \sqrt{|c|/a}\}$

则， $n>n_0$后有 $c_1n^2 \le an^2+bn+c \le c_2n^2$

$an^2+bn+c \le c_2n^2$

$\Leftrightarrow an^2+bn+c \le an^2+(a/2)n^2 + (a/4)n^2$

$\Leftrightarrow 0 \le an/2[n-2b/a]+(a/4)(n^2-4c/a)$

$an^2+bn+c \ge c_1n^2$

$\Leftrightarrow an^2+bn+c \ge (a/4)n^2$

$\Leftrightarrow an/2[n+2b/a]+(a/4)(n^2+4c/a) \ge 0$

Example

例2 证明: $6n^3 \neq \Theta(n^2)$

反证. 如果存在$c_1,c_2>0$，n_0使得当$n \geq n_0$时，有

$$c_1 n^2 \leq 6n^3 \leq c_2 n^2$$

于是，当$n>c_2/6$时，必有

$$n \leq c_2/6$$

这与n的取值范围矛盾。

Example

例3 $p(n)=\sum_{i=0}^{d} a_i n^i = \Theta(n^d)$ $(a_d>0)$

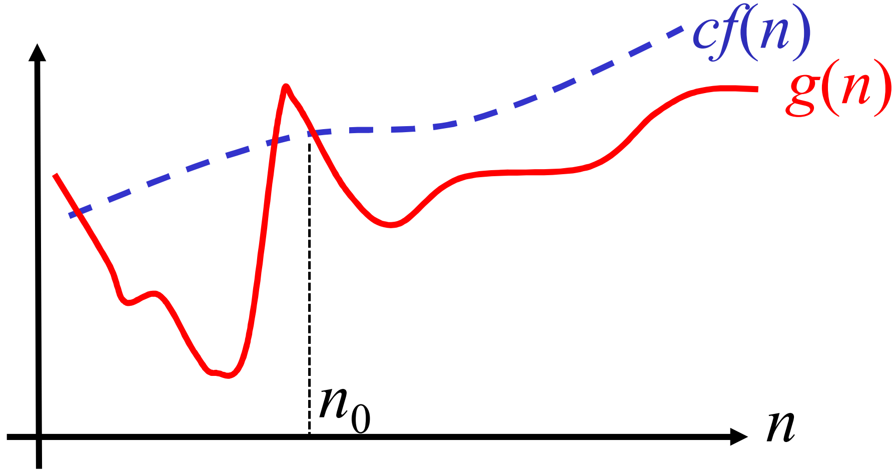
例4 对于任意常数$c>0$有 $c=\Theta(n^0)=\Theta(1)$

取$c_1=c/2$, $c_2=3c/2$, $n_0=1$, 则$n>n_0$后有

$$c_1 n^0 \le c \le c_2 n^0$$

2.1.2 低阶函数集合

定义2.1.2 (低阶函数集) $O(f(n))=\{g(n) \mid \exists c>0, n_0, \forall n>n_0 \text{有} 0 \le g(n) \le cf(n)\}$ 称为比$f(n)$低阶的函数集合



- 若$g(n) \in O(f(n))$, 则称$f(n)$是$g(n)$的上界
- $g(n) \in O(f(n))$常记为 $g(n) = O(f(n))$

Example

例1 $f(n)=\Theta(g(n)) \Rightarrow f(n)=O(g(n))$

$\Theta(g(n)) \subseteq O(g(n))$

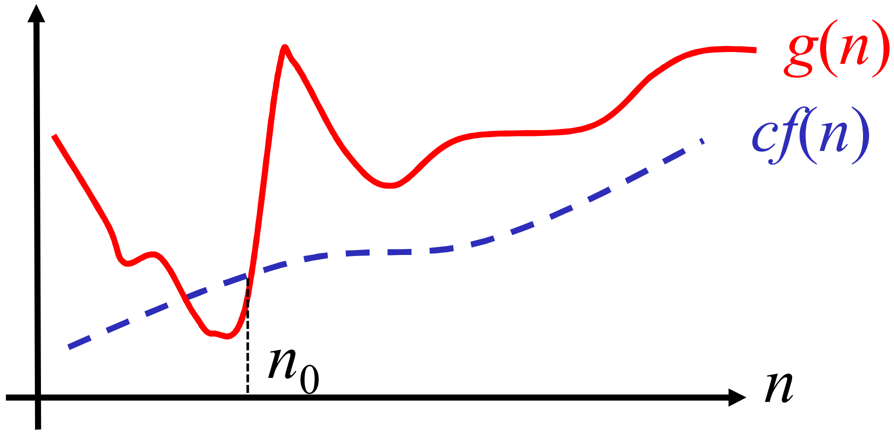
例 2 证明: $n=O(n^2)$

证明: 令$c=1, n_0=1$,

则$n \geq n_0$后, 恒有 $0 \leq n \leq cn^2$

2.1.3 高阶函数集合

定义2.1.3 (高阶函数集) $\Omega(f(n))=\{g(n) \mid \exists c>0, n_0, \forall n>n_0 \text{有} 0 \le cf(n) \le g(n)\}$ 称为比$f(n)$高阶的函数集合



- 若$g(n) \in \Omega(f(n))$, 则称$f(n)$是$g(n)$的下界
- $g(n) \in \Omega(f(n))$常记为 $g(n) = \Omega(f(n))$

HIT
CS&E

定理2.1 $f(n)=\Theta(g(n)) \Leftrightarrow f(n)=O(g(n))$ 且 $f(n)=\Omega(g(n))$

证明：$\Rightarrow$. 由$f(n)=\Theta(g(n))$知，$\exists\, c_1,c_2>0, n_0>0$, 当$n\geq n_0$时

$$c_1 g(n) \leq f(n) \leq c_2 g(n)$$

易知 $f(n)=O(g(n))$ 且 $f(n)=\Omega(g(n))$

$\Leftarrow$. 由$f(n)=\Omega(g(n))$知，$\exists\, c_1>0, n_1>0$, 当$n\geq n_1$时

$$c_1 g(n) \leq f(n)$$

由$f(n)=O(g(n))$知，$\exists\, c_2>0, n_2>0$, 当$n\geq n_2$时

$$f(n) \leq c_2 g(n)$$

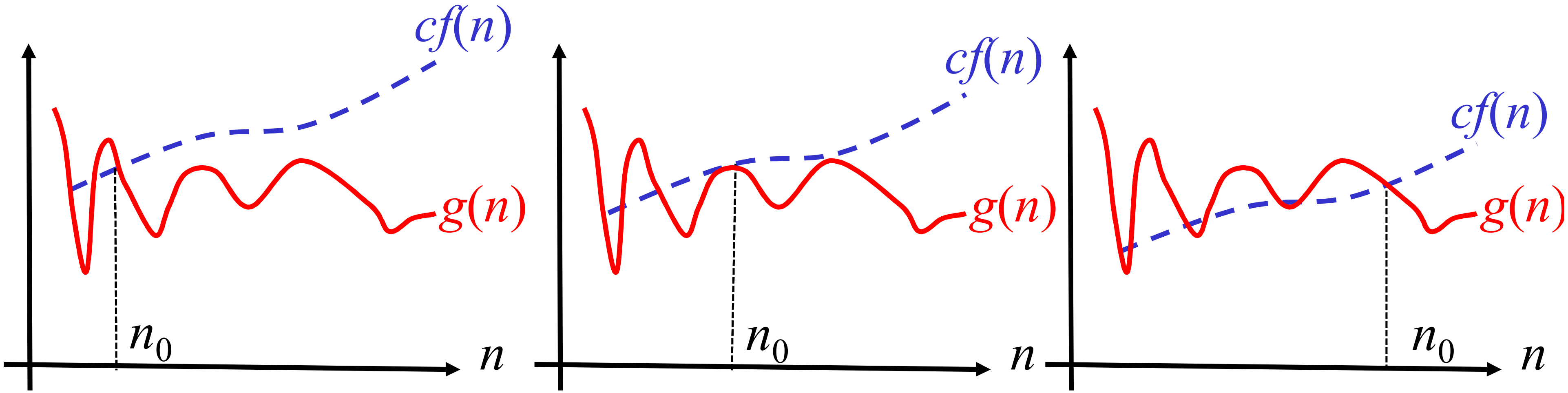
取$n_0=\max\{n_1,n_2\}>0$, 当$n\geq n_0$时

$$c_1 g(n) \leq f(n) \leq c_2 g(n)$$

即： $f(n)=\Theta(g(n))$

2.1.4 严格低阶函数集合

定义2.1.4 (严格低阶函数集) $o(f(n))=\{g(n) \mid \forall c>0, \exists n_0, 0 \le g(n) < cf(n)$ 对 $n \ge n_0$ 恒成立$\}$ 称为$f(n)$的严格低阶函数集合



c逐步变小时，n_0相应地变化

- 若$g(n) \in o(f(n))$, 则称$f(n)$是$g(n)$的严格上界
- $g(n) \in o(f(n))$常记为 $g(n)=o(f(n))$

HIT
CS&E

例1　证明：$2n = o(n^2)$

证 明：对$\forall c>0$, 欲使$2n<cn^2$必有$2/c<n$
于是，$\forall c>0$, 取$n_0=2/c$,当$n\geq n_0$必有$2n<n^2$

例2　证明：$2n^2 \neq o(n^2)$

证 明：当$c=1$时,对$\forall n_0$, $2n^2<cn^2$在$n\geq n_0$都不成立

HIT
CS&E

命题2.1. $f(n)=o(g(n)) \Leftrightarrow \lim_{n\to\infty}\frac{f(n)}{g(n)}=0$

证 明：$f(n)=o(g(n))$

$\Leftrightarrow \forall c>0, \exists n_0$,使得 $0\le f(n)<cg(n)$对$n\ge n_0$恒成立

$\Leftrightarrow \forall c>0, \exists n_0$,使得 $0\le \frac{f(n)}{g(n)}<c$对$n\ge n_0$恒成立

$\Leftrightarrow \lim_{n\to\infty}\frac{f(n)}{g(n)}=0$ 且$f(n)\ge 0, g(n)>0$

2.1.5 严格高阶函数集合

定义2.1.4 (严格高阶函数集) $\omega(f(n))=\{g(n)|\forall c>0, \exists n_0, 0\le cf(n)<g(n)$ 对 $n\ge n_0$ 恒成立$\}$ 称为 $f(n)$ 的严格高阶函数集合

$\forall c>0, \exists n_0, 0\le cf(n)<g(n)$ 对 $n\ge n_0$ 恒成立

$\Leftrightarrow \forall c>0, \exists n_0, 0\le f(n)<(1/c)g(n)$ 对 $n\ge n_0$ 恒成立

$\Leftrightarrow \forall c>0, \exists n_0, 0\le f(n)<cg(n)$ 对 $n\ge n_0$ 恒成立

命题2.2 $g(n)=\omega(f(n)) \Leftrightarrow f(n)=o(g(n))$

命题2.3. $g(n)=\omega(f(n)) \Leftrightarrow \lim\limits_{n\to\infty}\frac{g(n)}{f(n)}=\infty$

Example

例1 证明：$n^2/2 = \omega(n)$

证明：$\lim\limits_{n\to\infty} \dfrac{n^2/2}{n} = \infty$

例2 证明：$n^2/2 \neq \omega(n^2)$

证明：$\lim\limits_{n\to\infty} \dfrac{n^2/2}{n^2} \neq \infty$

2.1.6 函数阶的性质

- 传递性
  - $f(n)=\Theta(g(n)) \wedge g(n)=\Theta(h(n)) \Rightarrow f(n)=\Theta(h(n))$
  - $f(n)=O(g(n)) \wedge g(n)=O(h(n)) \Rightarrow f(n)=O(h(n))$
  - $f(n)=\Omega(g(n)) \wedge g(n)=\Omega(h(n)) \Rightarrow f(n)=\Omega(h(n))$
  - $f(n)=o(g(n)) \wedge g(n)=o(h(n)) \Rightarrow f(n)=o(h(n))$
  - $f(n)=\omega(g(n)) \wedge g(n)=\omega(h(n)) \Rightarrow f(n)=\omega(h(n))$

HIT
CS&E

2.1.6 函数阶的性质(续)

- 自反性
  - $f(n) = \Theta(f(n))$
  - $f(n) = O(f(n))$
  - $f(n) = \Omega(f(n))$
- 对称性
  - $f(n) = \Theta(g(n)) \Leftrightarrow g(n) = \Theta(f(n))$
- 反对称性
  - $f(n) = O(g(n)) \Leftrightarrow g(n) = \Omega(f(n))$
  - $f(n) = o(g(n)) \Leftrightarrow g(n) = \omega(f(n))$

注意

!

* 并非所有函数都是可比的，即对于函数 $f(n)$ 和 $g(n)$，可能 $f(n) \neq O(g(n)), f(n) \neq \Omega(g(n))$. 例如，$n$ 和 $n^{1+\sin n}$.

2.2 标准符号和通用函数

- Flour 和 ceiling
- 多项式

2.2.1 Flour和ceiling

定义2.2.1(Flours和ceiling). $\lfloor x \rfloor$表示小于或等于x的最大整数. $\lceil x \rceil$表示大于等于x的最小整数.

命题 2.2.1 $x-1<\lfloor x \rfloor \le x \le \lceil x \rceil < x+1$

命题 2.2.2 对于任意整数 n, $\lceil n/2 \rceil + \lfloor n/2 \rfloor = n$

证． 若 $n = 2k$ ，则 $\left\lceil \frac{n}{2} \right\rceil = k$ ， $\lfloor n/2 \rfloor = k$ ． 于是 $\left\lceil \frac{n}{2} \right\rceil + \lfloor n/2 \rfloor = 2k = n$

若 n=2k+1， 则 $\left\lceil \frac{n}{2} \right\rceil + \lfloor n/2 \rfloor = k+1+k = 2k+1 = n$ ．

命题 2.2.3 设 *n*、*a*、*b* 是任意整数， $a \neq 0, b \neq 0$ ， 则

(1) $\lceil \lceil n/a \rceil /b \rceil = \lceil n/ab \rceil$ 。

(2) $\lfloor \lfloor n/a \rfloor /b \rfloor = \lfloor n/ab \rfloor$

证． (1) 若 $n = kab$ ，则 $\left\lceil \frac{n/a}{b} \right\rceil = \left\lceil \frac{kb}{b} \right\rceil = k = \left\lceil \frac{kab}{ab} \right\rceil = \left\lceil \frac{n}{ab} \right\rceil$ 。

若 $n = kab + \alpha$ ， 0< $\alpha < ab$ ， 则

$$\left\lceil \left\lceil \frac{n}{a} \right\rceil \Big/ b \right\rceil = \left\lceil \frac{kb+1}{b} \right\rceil = k+1 = \left\lceil \frac{n}{ab} \right\rceil = \left\lceil \frac{kab+\alpha}{ab} \right\rceil = k+1$$

(2) 类似于(1)的证法。

HIT CS&E

2.2.2 和式的估计与界限

1. 线性和

命题 2.4.5 $\sum_{k=1}^{n}(ca_k + b_k) = c\sum_{k=1}^{n} a_k + \sum_{k=1}^{n} b_k$

命题 2.4.6 $\sum_{k=1}^{n}\theta(f(k)) = \theta\left(\sum_{k=1}^{n} f(k)\right)$

证. 对 n 用数学归纳法证明。

当 $n = 1$ 时， $\theta(f(1)) = \theta(f(1))$ 显然成立。假设 $n \le m$ 时成立。

令 $n = m + 1$， 则 $\sum_{k=1}^{m+1}\theta(f(k)) = \sum_{k=1}^{m}\theta(f(k)) + \theta(f(m+1))$

$$= \theta\left(\sum_{k=1}^{m} f(k)\right) + \theta(f(m+1))$$

$$= \theta\left(\sum_{k=1}^{m} f(k) + f(m+1)\right)$$

$$= \theta\left(\sum_{k=1}^{m+1} f(k)\right)。$$

2.2 递归方程

- 递归方程：根据其在较小的输入值上的取值递归地描述一个函数的方程或不等式

- 递归方程例：**Merge-sort**算法的复杂性方程

$$T(n)=\theta(1) \quad \text{if } n = 1$$

$$T(n)=2T(n/2)+\theta(n) \quad \text{if } n > 1.$$

T(n)的解是$\theta(n\log n)$

求解递归方程的三个主要方法

- 迭代方法:
  - 把方程转化为一个和式
  - 然后用估计和的方法来求解
- 替换方法:
  - 先猜测方程的解,
  - 然后用数学归纳法证明.
- **Master**方法:
  - 求解型为$\mathbf{T}(\boldsymbol{n}) = \boldsymbol{a}\mathbf{T}(\boldsymbol{n}/\boldsymbol{b})+\boldsymbol{f}(\boldsymbol{n})$的递归方程

2.2.1 迭代方法

方法：

循环地展开递归方程，
把递归方程转化为和式，
然后可使用求和技术解之

HIT CS&E

例1. $T(n)=2T(n/2)+cn$

$$=2^2T(n/2^2)+cn+cn$$

$$=2^3T(n/2^3)+cn+cn+cn$$

$$=\ldots$$

$$=2^kT(n/2^k)+knc$$

$$=2^kT(1)+knc \qquad n=2^k$$

$$=nT(1)+cn\log n$$

$$=\Theta(n\log n)$$

例2

$$T(n) = n + 3T\left(\left\lfloor n/4 \right\rfloor\right)$$

$$= n + 3\left(\left\lfloor n/4 \right\rfloor + 3T\left(\left\lfloor n/16 \right\rfloor\right)\right)$$

$$= n + 3\left(\left\lfloor n/4 \right\rfloor + 3\left(\left\lfloor \frac{n}{16} \right\rfloor + 3T\left(\left\lfloor n/64 \right\rfloor\right)\right)\right)$$

$$= n + 3\left\lfloor n/4 \right\rfloor + 9\left\lfloor \frac{n}{16} \right\rfloor + 27T\left(\left\lfloor n/64 \right\rfloor\right)$$

$$= n + 3\left\lfloor n/4 \right\rfloor + 3^2\left\lfloor \frac{n}{4^2} \right\rfloor + 3^3\left(\left\lfloor \frac{n}{4^3} \right\rfloor\right) + \ldots\ldots + 3^i T\left(\left\lfloor n/4^i \right\rfloor\right)$$

$$\boxed{\text{令}\ \frac{n}{4^i} = 1 \Rightarrow 4^i = n \Rightarrow i = \log_4 n}$$

$$= n + 3\left\lfloor n/4 \right\rfloor + 3^2\left\lfloor \frac{n}{4^2} \right\rfloor + 3^3\left(\left\lfloor \frac{n}{4^3} \right\rfloor\right) + \ldots\ldots + 3^{\log_4 n} T(\lfloor 1 \rfloor)$$

$$\le \sum_{i=0}^{\log_4 n} 3^i \, n/4^i + \mathrm{O}(n) \le n\sum_{i=0}^{\infty}\left(\frac{3}{4}\right)^i = n \times \frac{1}{1-\frac{3}{4}} = 4n = O(n)$$

2.2.2 替换法

方法：

1. 变量代换, 将方程转换成已知方程
2. 先根据方程的形式猜测解
   然后用数学归纳法证明

变量代换

例1. $T(n)=2T(n/2+17)+n$

令 $n=m+34$,则

$T(m+34)=2T(m/2+34)+m+34$

令 $T(m+34)=S(m)$,则

$S(m)=2S(m/2)+m+34$

$S(m)=\Theta(m\log m)$

$T(n)=\Theta(n\log n)$

HIT
CS&E

例2. $T(n)=2T(n^{1/2})+\log n$

令 $n=2^m$,则

$$T(2^m)=2T(2^{m/2})+m$$

令 $T(2^m)=S(m)$,则

$$S(m)=2S(m/2)+m$$

$$S(m)=\Theta(m\log m)$$

$$T(n)=\Theta(\log n \log\log n)$$

先猜后证

例. $T(n)=2T(n/2+17)+n$

由于 $n/2$ 与 $n/2+17$ 在 n 充分大之后相近

故猜 $T(n/2) \approx T(n/2+17)$ 在 n 充分大后成立

故

$$T(n) \approx 2T(n/2)+n$$

故原始方程的解 $T(n)=\Theta(n\log n)$

再用数学归纳法证明

猜测方法 I：
猜测上下界，减少不确定性范围

例 3．求解 $T(n)=2T\left(\frac{n}{2}\right)+n$.

解：首先证明 $T(n)=\Omega(n)$，$T(n)=O(n^2)$

然后逐阶地降低上界、提高下界。

$\Omega(n)$的上一个阶是$\Omega(n\log n)$，
$O(n^2)$的下一个阶是 $O(n\log n)$。

HIT
CS&E

细微差别的处理

- 问题：猜测正确，数学归纳法的归纳步
  似乎证不出来
- 解决方法:从猜测结论中减去一个低阶项，
  可能方法就能用了

例 4. 求解 $T(n)=T(\lfloor n/2 \rfloor)+T\left(\left\lceil \frac{n}{2} \right\rceil\right)+1$

解：(1) 我们猜 $T(n)=\mathrm{O}(n)$

证： $T(n)\le c\lfloor n/2 \rfloor+c\lceil n/2 \rceil+1=cn+1\ne cn$

证不出 $T(n)=\mathrm{O}(cn)$

(2) 减去一个低阶项，猜 $T(n)\le cn-b$， $b\ge 0$ 是常数

证：设当 $\le n-1$ 时成立

$$T(n)=T(\lfloor n/2 \rfloor)+T\left(\left\lceil \frac{n}{2} \right\rceil\right)+1\le c\left\lfloor \frac{n}{2} \right\rfloor-b+c\left\lceil \frac{n}{2} \right\rceil-b+1$$

$$=cn-2b+1=cn-b-b+1\le cn-b$$ （只要 $b\ge 1$ ）。

* c 必须充分大，以满足边界条件。

避免陷阱

例 5. 求解 $T(n)=2T(\lfloor n/2 \rfloor)+n$ 。

解： 猜 $T(n)=\mathrm{O}(n)$

证： 用数学归纳法证明 $T(n)\le cn$ 。

$$T(n)\le 2(c\lfloor n/2 \rfloor)+n\le cn+n=\mathrm{O}(n)$$

--错！！

错在那里：过早地使用了 $\mathrm{O}(n)$ 而陷入了陷阱应该在证明了 $T(n)\le cn$ 才可用。从 $T(n)\le cn+n$ 不可能得到 $T(n)\le cn$ 因为对于任何 c>0，我们都得不到 $cn+n\le cn$.

2.2.3 Master method

目的：求解 $T(n) = aT\left(n/b\right) + f(n)$ 型方程， $a \geq 1, b > 0$ 是常数， $f(n)$ 是正函数

方法：记住三种情况，则不用笔纸即可求解上述方程.

Master 定理

定理 2.4.1 设$a \geq 1$和$b > 1$是常数，$f(n)$是一个函数，$T(n)$是定义在非负整数集上的函数$T(n) = aT\left(n/b\right) + f(n)$．$T(n)$可以如下求解：

(1)．若$f(n) = O\left(n^{\log_b a - \varepsilon}\right)$, $\varepsilon > 0$是常数，则$T(n) = \theta\left(n^{\log_b a}\right)$.

(2)．若$f(n) = \theta\left(n^{\log_b a}\right)$，则$T(n) = \theta\left(n^{\log_b a} \lg n\right)$.

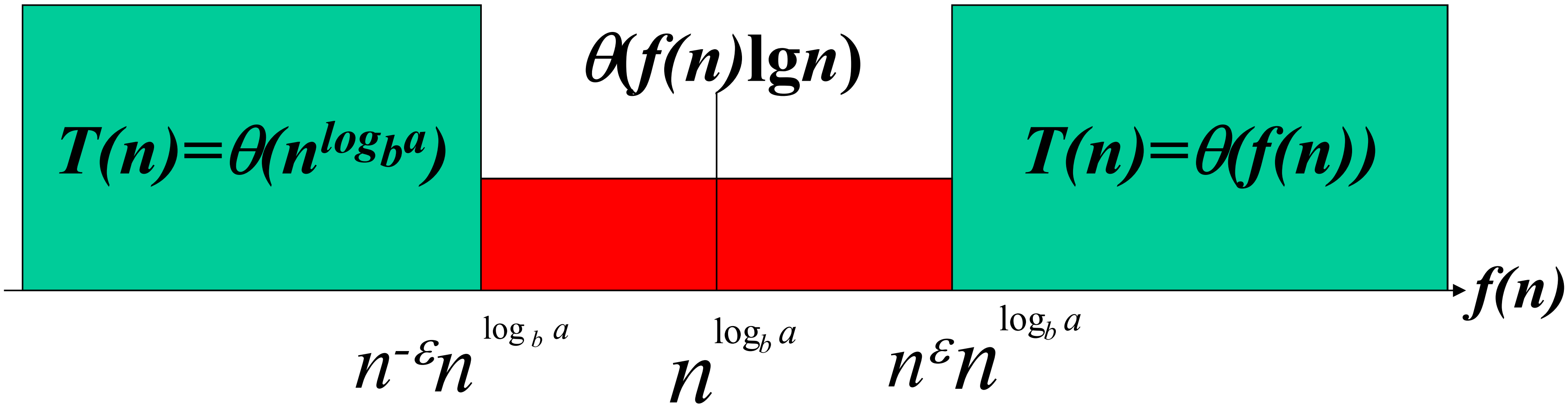
(3)．若$f(n) = \Omega\left(n^{\log_b a + \varepsilon}\right)$, $\varepsilon > 0$是常数，且对于所有充分大的n $af\left(n/b\right) \leq cf(n)$, C<1 是常数，则$T(n) = \theta(f(n))$.

*直观地：我们用 $f(n)$ 与 $n^{\log_b a}$ 比较

(1). 若 $n^{\log_b a}$ 大，则 $T(n)=\theta\left(n^{\log_b a}\right)$

(2). 若 $f(n)$ 大，则 $T(n)=\theta(f(n))$

(3). 若 $f(n)$ 与 $n^{\log_b a}$ 同阶，则 $T(n)=\theta\left(n^{\log_b a}\lg n\right)=\theta(f(n)\lg n)$.



对于红色部分，Master定理无能为力

HIT
CS&E

更进一步:

(1). 在第一种情况, $f(n)$ 不仅小于 $n^{\log_b a}$，必须多项式地小于, 即对于一个常数 $\varepsilon > 0$， $f(n) = O(\frac{n^{\log_b a}}{n^{\varepsilon}})$.

(2). 在第三种情况, $f(n)$ 不仅大于 $n^{\log_b a}$，必须多项式地大于，即对一个常数 $\varepsilon > 0$， $f(n) = \Omega(n^{\log_b a} \cdot n^{\varepsilon})$.

Master定理的使用

例 1. 求解 $T(n) = 9T\left(n/3\right) + n$.

解： $a = 9$， $b = 3$， $f(n) = n$ ， $n^{\log_b a} = \theta\left(n^2\right)$

$\because f(n) = n = O\left(n^{\log_b a - \varepsilon}\right)$ ， $\varepsilon = 1$

$\therefore T(n) = \theta\left(n^{\log_b a}\right) = \theta\left(n^2\right)$

例 2. 求解 $T(n) = T\left(2n/3\right) + 1$.

解： $a = 1$， $b = \left(3/2\right)$， $f(n) = 1$， $n^{\log_b a} = n^{\log_{3/2} 1} = n^0 = 1$ ，

$f(n) = 1 = \theta(1) = \theta\left(n^{\log_b a}\right)$ ， $T(n) = \theta\left(n^{\log_b a} \lg n\right) = \theta(\lg n)$

Master定理的使用（续）

例 3． 求解 $T(n)=3T\left(n/4\right)+n\lg n$

解： $a=3$， $b=4$， $f(n)=n\lg n$， $n^{\log_b a}=n^{\log_4 3}=O\left(n^{0.793}\right)$

(1) $f(n)=n\lg n\geq n=n^{\log_b a+\varepsilon}$ ，$\varepsilon\approx 0.2$

(2) 对所有 n, $af\left(n/b\right)=3\times\frac{n}{4}\lg\frac{n}{4}=\frac{3}{4}n\lg\frac{n}{4}\leq\frac{3}{4}n\lg n=cf(n)$, $c=\frac{3}{4}$.

于是， $T(n)=\theta(f(n))=\theta(n\lg n)$

例 4． 求解 $T(n)=2T\left(n/2\right)+n\lg n$.

解： $a=2$， $b=2$， $f(n)=n\lg n$， $n^{\log_b a}=n$. $f(n)=n\lg n$ 大于 $n^{\log_b a}=n$ ，但不是多项式地大于，Master 定理不适用于该 $T(n)$.