Growth of Function

Θ

Definition

假定 f 是一個 asymptotically nonnegative, 也就是：

$$\exists k. \forall n > k.\ f(n) \geq 0$$

(直覺的理解：最後會為正的函數)。

$$\begin{aligned}\Theta(g(n)) = \{f(n) | \exists c_1, c_2, n_0 > 0. \forall n \geq n_0. \\ 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \}\end{aligned}$$

Remark

1. 表示某個函數 (f) 可以被某個函數 (g) 用常數「夾起來」。概念上來說增長速度跟某個函數大致相同是這個意思。
2. 這跟 $\lim_{n\to\infty} \frac{g(n)}{f(n)}$ 存在的定義不一樣。因為這邊只要求 g 可以把 f 夾住，沒有規定需要收斂到一個特定值。如果用 $\epsilon - \delta$ 定義看的話也很明顯發現兩者並不充分或必要。
3. 接下來會介紹兩個表示「上限」與「下限」的集合。可以看成是 Θ 定義中的一半：

$$\underbrace{c_1 g(n) \leq f(n)}_{\Omega(g(n))} \leq c_2 g(n)$$

$$c_1 g(n) \leq \underbrace{f(n) \leq c_2 g(n)}_{O(g(n))}$$

4. 後面會另外有兩個 qualifier 跟現在不同的定義。比如：

$$\begin{aligned}o(g(n)) = \{f(n) | \forall c_1. \exists n_0. \forall n \geq n_0 \\ 0 \leq c_1 g(n) \leq f(n)\}\end{aligned}$$

> *本來是「對於每一個裡面的函數，都存在 $c_1, n_0, \dots$ 使得等號 ...」，這裡是「對於每一個裡面的函數，對於任意 c_1，都存在 n_0，使得等號...」*

5. $\Theta(g(n))$ 是個函數的集合，因此照理說應該用「$\in$」 說明某個函數在 $\Theta(g(n))$ 裡面。但是傳統上用 「$f(n) = O(g(n))$ 」表示「$f(n) \in O(g(n))$」。

O

Definition

$$O(g(n)) = \{f(n) | \exists c_2, n_0 > 0. \forall n \geq n_0. \\ f(n) \leq c_2 g(n)\}$$

Remark

1. $\Theta(g(n)) \subset O(g(n))$
2. 可以當作複雜度的某種「上限」

Ω

Definition

$$\Omega(g(n)) = \{f(n) | \exists c_1, n_0 > 0. \forall n \geq n_0. \\ c_1 g(n) \leq f(n)\}$$

Remark

1. $\Theta(g(n)) \subset \Omega(g(n))$
2. 可以看作是演算法的「下介」。
3. 結合大 O 的第一條可以發現：

$$\Theta(g(n)) \subset \Omega(g(n)) \cap O(g(n))$$

Q: 反過來有沒有對?

Thm ：Θ 的充要條件

$$f \in \Theta(g(n)) \iff f \in O(g(n)) \cap \Omega(g(n))$$

白話文是「 f 的上下限都是 g」跟「 f 是 $\Theta(g(n))$」是同一回事

「pf」:

$$\begin{aligned} f \in \Theta(g(n)) &\iff \exists c_1, c_2, n_0 > 0. \forall n > n_0 \,.\, c_1 g(n) \leq f(n) \leq c_2 g(n) \\ &\iff \begin{cases} \exists c_1, n_0, \forall n > n_0, c_1 g(n) \leq f(n), \\ \exists c_2, n_0, \forall n > n_0, f(n) \leq c_2 g(n) \end{cases} \\ &\iff f(n) \in \Omega(g(n)), f(n) \in O(g(n)) \\ &\iff f(n) \in \Omega(g(n)) \cap O(g(n)) \end{aligned}$$

o

Definition

$$\begin{gathered} o(g(n)) = \{f(n) | \forall c_2, \exists n_0 > 0. \forall n \geq n_0 \,. \\ f(n) < c_2 g(n)\} \end{gathered}$$

Remark

1. 這個定義等價於:

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

   證明方式很簡單，$\forall c_2 = \epsilon > 0$，有:

$$f(n) < c_2 g(n) \iff \frac{f(n)}{g(n)} < c_2 = \epsilon$$

   因此:

$$
\begin{aligned}
&f(n) = o(g(n)) \\
&\iff \forall c_2, \exists n_0 > 0. \forall n \geq n_0.\, f(n) < c_2 g(n) \\
&\iff \forall c_2, \exists n_0 > 0. \forall n \geq n_0.\, \frac{f(n)}{g(n)} < c_2 := \epsilon \\
&\iff \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0
\end{aligned}
$$

ω

Definition

$$
\omega(g(n)) = \{f(n) | \forall c_1, \exists n_0 > 0. \forall n \geq n_0. \\
c_1 g(n) < f(n)\}
$$

Remark

類似的，這個定義跟：

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

等價。

證明方法也一樣，對於任意大的值 $M = c_1$，都可以取 $\omega(g(n))$ 定義保證的 n_0，就可以保證 $n > n_0$ 之後，再移項 ω 定義得到這個 n_0 保證 $\frac{f(n)}{g(n)} > c_1 = M$ ，由極限定義知極限趨近無窮。

反之，對於任意 $M = c_1$，取極限定義保證的 n_0，就可以保證任意的$n > n_0$ 下，$\frac{f(n)}{g(n)} > M = c_1$ ，因此推得 $\forall c_1. \exists n_0 > 0. \forall n > n_0.\, c_1 g(n) \leq f(n)$

性質

1. Transitivity：全部都有。
2. Reflexivity：大寫的都有，小寫的因為定義是嚴格的大於/小於所以不會對
3. Symmetry：只有 Θ 會對。(因為 $\forall c > 0. f(n) \geq cg(n) \Rightarrow g(n) \leq \frac{1}{c} f(n)$，對 c_1 跟 c_2 各自做一樣的事就可以了)而另外 4 個則是互有 Transpose Symmetriy
4. Transpose Symmetriy：

   $f(n) = O(g(n)) \iff g(n) = \Omega(f(n)), f(n) = o(g(n)) \iff g(n) = \omega(f(n))$

一些用語

1. It takes O(n) time to solve a problem 意思是：

   $$\text{若輸入大小為}n\text{時，需要花的時間是 } f(n)\text{，則} f(n) \in O(n)$$

2. $O(f(n)) = O(g(n))$ 意思是：

   $$\forall h(n) \in O(f(n)). h(n) \in O(g(n))$$

   所以更其實是：

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

3. $O(g_1(n)) + O(g_2(n)) = O(h(n))$ 意思是：

   $$\forall f_1(n) \in O(g_1(n)). \forall f_2(n) \in O(g_2(n)). f_1(n) + f_2(n) \in O(h(n))$$

   這邊要注意的是，跟剛剛提過 $f(n) = O(g(n))$ 類似，「=」並不是集合相等的意思，而是「⊆」的意思。（所以才要特別提醒這個）。

   $O(g_1(n)) \cdot O(g_2(n)) = O(h(n))$、$O(g_1(n)) - O(g_2(n)) = O(h(n))$ 等等都是一樣的量詞，只是後面的命題換成相應的運算。

4. $O(O(g(n))) = O(h(n))$

   $$\forall f(n) \in O(g(n)). O(f(n)) = O(h(n))$$

   可以把 $O(f(n)) = O(g(n))$ 的定義展開來，這整句話變成：

   $$\forall f(n) \in O(g(n)). \forall f' \in O(f(n)). f'(n) \in O(h(n))$$

白話文是：隨便挑一個 $O(g(n))$ 中的函數 $f(n)$，並且找出這個函數的 $O(f(n))$。則對於每一個這樣的集合中的任何一個函數 $f'(n)$，都滿足 $f'(n) \in O(h(n))$

大 O 公式

假定 $f(n)$, $g(n)$ 是 assymtopic nonnegative 的函數，則：

1. f(n) = O(f(n))

$$f(n) = O(f(n))$$

實際上就是上面的 Reflexive。

2. ∀c > 0. cO(f(n)) = O(f(n))

$$\forall c > 0.\ \ c \cdot O(f(n)) = O(f(n))$$

這句話的意思是：

$$\forall c > 0. \forall f_c(n) \in \{cf'(n) | f'(n) \in O(f(n))\}. \ f_c(n) \in O(f(n))$$

3. f(n) = O(g(n)) ⇒ O(f(n)) = O(g(n))

$$f(n) \in O(g(n)) \Rightarrow O(f(n)) = O(g(n))$$

關於 $O(f(n)) = O(g(n))$ 的意思可以參考上面提過的東西。

4. O(f(n)) · O(g(n)) = O(f(n) · g(n))

$$O(f(n)) \cdot O(g(n)) = O(f(n) \cdot g(n))$$

這句話的意思是:

$$\forall f'(n) \in O(f(n)), g'(n) \in O(g(n)).\ f'(n) \cdot g'(n) \in O(f(n) \cdot g(n))$$

5. O(f(n) · O(g(n))) = f(n) · O(g(n))

$$O(f(n) \cdot g(n)) = f(n) \cdot O(g(n))$$

這句話的意思也是類似:

$$\forall h(n) \in O(f(n) \cdot g(n)). h(n) \in \{f(n) \cdot g'(n) | g'(n) \in O(g(n))\}$$

這邊的 $f(n) \cdot O(g(n))$ 意思是「把所有 $O(g(n))$ 中的元素，乘以 $f(n)$ 之後，形成的集合」。

例子

1. f(n) + g(n) = Θ(max{f(n) + g(n)})

$$f(n) + g(n) = \Theta(max\{f(n) + g(n)\})$$

因為 Θ 有 Symmetry，所以可以證：

$$max\{f(n), g(n)\} = \Theta(f(n) + g(n))$$

再用 Symmetry 就可以知道原命題成立。

首先，依照 assumptopic nonnegative 可以知道：

$$\exists n_0 > 0. \forall n > n_0. f(n) > 0, g(n) > 0$$

因：

$$\forall n > n_0. f(n) \leq g(n) + f(n)$$
$$\forall n > n_0. g(n) \leq g(n) + f(n)$$

故：

$$\forall n > n_0. max\{f(n), g(n)\} \leq g(n) + f(n)$$

又，顯然：

$$\forall n > n_0.\, f(n) + g(n) \leq max\{f(n), g(n)\} + max\{f(n), g(n)\}$$

因此：

$$\forall n > n_0.\, \frac{1}{2}(f(n) + g(n)) \leq max\{f(n), g(n)\}$$

故選擇：

$$\begin{cases} n_0 = A.P.\text{保證的}n_0 \\ c_1 = \frac{1}{2} \\ c_2 = 1 \end{cases}$$

即有：

$$\forall n > n_0.\, \frac{1}{2}(f(n) + g(n)) \leq max\{f(n), g(n)\} \leq f(n) + g(n)$$

2. 多項式 = Θ(最高次項)

$$\forall f(n) \in \{p(n) | p(n) \in \mathbf{P}_k(n), \exists n_0 > 0. \forall n > n_0.\ f(n) > 0\}. \\ f(n) \in \Theta(n^k)$$

使用歸納法：

1. k = 1 時，假定：

$$f_1(n) = a_1 n + a_0$$

並且 $a_1 > 0$。則：

$$\forall n > 1.\ 2a_1 + a_0 > a_1 n + a_0 > \frac{1}{2} a_1 n + a_0$$

因此：

$$f_1(n) \in \Theta(n^1)$$

2. n = k > 1時

   假定:

$$f_k(n) = a_k n^k + f_{k-1}(n)$$

   其中:

$$f_{k-1}(n) \in \Theta(n^{k-1})$$

   可知:

$$\exists c_1', c_2', n_0' > 0. \forall n > n_0'.\, c_1' n^{k-1} \le f_{k-1}(n) \le c_2' n^{k-1}$$

   則:

   1. $a_k > 0$: 假定 $a_k < 0$, 顯見 $\exists n' > 0. \forall n > n'.\, a_k n + c_2' < 0$ ，故 $\exists n' > 0. \forall n > n'.\, f_k(n) \le (a_k n + c_2') n^{k-1} < 0$。與 asymtopic nonnegative 的假定矛盾。
   2. 又因為:

      $\exists c_1', c_2', n_0'. \forall n > n_0'.$
      $c_1' n^{k-1} \le f_{k-1}(n) < c_2' n^{k-1} \Rightarrow$
      $a_k n^k + c_1' n^{k-1} \le a_k n^k + f_{k-1}(n) \le a_k n^k c_2' n^{k-1} \Rightarrow$
      $a_k n^k \le a_k n^k + c_1' n^{k-1} \le a_k n^k + f_{k-1}(n) \le a_k n^k + c_2' n^{k-1} \le (a_k + c_2')$

      故取:

$$\begin{cases} c_1 = a_k \\ c_2 = a_k + c_2' \\ n_0 = n_0' \end{cases}$$

      則可使:

$$\forall n > n_0.\, c_1 n^k \le f(n) \le c_2 n^k$$

3. 利用極限輔助證明

雖然極限跟大 O 並不是充分必要條件，但有時候如果有極限，可以透過極限來證明大 O（然後用 Transpose Symmetriy 告訴我們可以用同一招證大 Ω）。比如：

$$\forall \epsilon > 0. \lg n \in O(n^{\epsilon})$$

乍看之下不這麼顯然。不過可以發現：

$$\begin{aligned}\lim_{x\to\infty} \frac{x^{\epsilon}}{\lg x} &= \lim_{x\to\infty} \frac{\epsilon x^{\epsilon-1}}{\left(\frac{1}{\ln 2}\cdot\frac{1}{x}\right)} \\ &= \lim_{x\to\infty} (\ln 2)\epsilon x^{\epsilon} \\ &= \infty\end{aligned}$$

所以，根據極限的定義可以知道：

$$\forall M > 0.\exists x_0. \forall x > x_0. \frac{\lg x}{x^{\epsilon}} > M$$

如果把 M 換成 c，$n_0 = \lceil x_0 \rceil \geq x_0$，並且作一點移項，上面這個命題變成：

$$\forall c > 0.\exists n_0. \forall n > n_0. n^{\epsilon} > \lg n$$

可以發現上面這個東西的結論比大 O 的定義還要更強。因為大 O 是：

$$\exists c > 0, n_0 > 0. \forall n > n_0. n^{\epsilon} > \lg n$$

大 O 只有要求 $\exists c > 0$，但極限保證到 $\forall c > 0$ 都成立。所以用極限裡面的符號寫大 O 該取的常數的話，就是取：

$$\begin{cases} c = M & (\mathbb{R}^{+}\text{ 裡面隨便選一個數都可以}) \\ \\ n_0 = \lceil x_0 \rceil & (x_0\text{是給定 } M \text{ 之後，極限定義保證的}x_0)\end{cases}$$

由剛剛極限保證的結果，得證：

$$\forall n > n_0. \lg x < cx^{\epsilon}$$

4. 利用極限吃屎

「極限存在」跟「在 Θ 裡面」並不充分也不必要。舉例來說:

$$f(n) = g(n) = (-1)^n$$

則:

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = 1$$

但:

$$f(n) \neq \Theta(g(n))$$
$$g(n) \neq \Theta(f(n))$$

類似地,假設:

$$\begin{cases} f(n) = 2 + (-1)^n \\ g(n) = 2 - (-1)^n \end{cases}$$

明顯發現:

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

不存在。然而隨便取一個超大的 c_2(比如說 10000) 跟 c_2(比如說 0.000001) 就可以證明:

$$f(n) = \Theta(g(n))$$

所以顯然兩者並不充分也不必要。

在這個例子中關鍵是 assymptopic nonnegative。

參考資料

1. CLRS
2. 呂學一老師 2010 年投影片
3.