

# 1 極限

極限的嚴謹定義和運算在大學的微積分會詳細說明，在複雜度的證明上不會用到太多，因此這裡就只簡單介紹極限的概念以及一些性質。概念上來說，一個函數在  $x$  趨近於  $a$  的極限為  $L$ ，記做

$$\lim_{x \rightarrow a} f(x) = L$$

表示當  $x$  越來越接近  $a$  (但不等於  $a$ ) 時， $f(x)$  的值就會越來越靠近  $L$ 。要特別注意的是， $\lim_{x \rightarrow a} f(x)$  和  $f(a)$  是完全不同的東西，因為  $f(a)$  可能不存在，也可能  $\lim_{x \rightarrow a} f(x) \neq f(a)$ ，或者有各種奇怪的情況 (到了大學微積分就會看到各式各樣莫名其妙的函數了)。舉例來說，可以構造一個函數

$$f(x) = \begin{cases} 1 & , \text{if } x = 1 \\ 0 & , \text{if } x \neq 1 \end{cases}$$

則

$$f(1) = 1, \quad \lim_{x \rightarrow 1} f(x) = 0$$

以下是幾個極限的運算性值，當然這些性質實際上是要證明的，但還是老話一句，這些就留給大學的微積分課吧。

已知  $\lim_{x \rightarrow a} f(x) = L, \lim_{x \rightarrow a} g(x) = M, \lim_{x \rightarrow a} h(x) = \infty$ ，則

$$(1) \lim_{x \rightarrow a} (f(x) + g(x)) = L + M$$

$$(2) \lim_{x \rightarrow a} (f(x) \times g(x)) = L \times M$$

$$(3) \lim_{x \rightarrow a} \frac{f(x)}{g(x)} = \frac{L}{M} \quad (M \neq 0)$$

$$(4) \lim_{x \rightarrow a} \frac{1}{h(x)} = 0$$

在複雜度的證明中，只會用到  $\lim_{x \rightarrow \infty} f(x)$  這種極限的形式，以下給出一些常用的等式

$$(1) \lim_{x \rightarrow \infty} k = k \quad (\text{常數})$$

$$(2) \lim_{x \rightarrow \infty} x = \infty \quad (\text{此時可以說極限不存在})$$

$$(3) \lim_{x \rightarrow \infty} \frac{1}{x} = 0$$

$$(4) \lim_{x \rightarrow \infty} k^x = \begin{cases} 0 & , \text{if } k < 1 \\ 1 & , \text{if } k = 1 \\ \infty & , \text{if } k > 1 \end{cases}$$

**例題 1**請計算  $\lim_{n \rightarrow \infty} \frac{n}{n+1}$ 。**解答**

$$\lim_{n \rightarrow \infty} \frac{n}{n+1} = \lim_{n \rightarrow \infty} \frac{1}{1 + \frac{1}{n}} = \frac{\lim_{n \rightarrow \infty} 1}{\lim_{n \rightarrow \infty} (1 + \frac{1}{n})} = \frac{1}{1} = 1$$

## 2 複雜度

以下我們將更深入了解 Big-O 的定義，這部分也許有點難理解，可以跳過沒關係，只要知道可以用兩方程相除結果的極限值來描述複雜度（量級）的關係就好（**想忽略此段落的同學可以直接跳到例題 5**）。我們會用時間複雜度和空間複雜度的觀念講解 Big-O 的精神，但 Big-O 絕對不僅限於這兩者，而是可以拿來描述所有函數，所以請大家絕對不要有「Big-O 就是時間/空間複雜度」的錯誤觀念。

在演算法的分析中，我們往往希望知道一個演算法在最差的情況下會跑多久，或是當資料量增加時，演算法所需時間的成長幅度。此時可以經過計算得到一個運算次數對資料量大小的函數  $f(n)$ ，表示該算法的特性，然而這個函數可能長得很醜，運算過程也大都很複雜，因此一般狀況下不會有人想做這種苦工。換個想法，如果我們不求那麼精準的計算  $f(n)$ ，而是找到另外一個函數  $g(n)$ ，並滿足

$$\forall n > 0, f(n) \leq g(n)$$

則我們就可以知道「不論  $f(n)$  多大，他最多就是和  $g(n)$  一樣大」，或是「 $f(n)$  的成長速度最快就是和  $g(n)$  一樣快」。然而，實際上我們很難知道  $f(n)$  所帶的常數倍數是多少，或者在不同的情況下， $f(n)$  所帶的常數倍數也可能會變動。舉例來說，如果現在討論的是空間複雜度， $f(n)$  表示一個演算法所需要的空間大小，則在不同單位下 (byte, KB, MB, ...)， $f(n)$  就會被乘上不同的常數，這時如果  $g(n)$  也要跟著變動就是一件很麻煩的事。因此，我們可以把一個常數加進不等式中，只要  $g(n)$  乘上一個常數之後可以滿足不等式就好了，如此一來在不同單位下就可以使用相同的  $g(n)$  了！更改後的式子如下

$$\exists k > 0, \forall n > 0, f(n) \leq k \cdot g(n)$$

另外，我們在意的是一個函數的「趨勢」，也就是當  $n$  越來越大時， $f(n)$  的成長速度有多快。反過來說，我們其實不太在意  $f(n)$  在  $n$  很小時的特性，因此可以再把不等式的條件放寬，只要在  $n$  超過一個門檻值  $N$  之後滿足不等式就好，也就是

$$\exists k > 0, \exists N, \forall n > N, f(n) \leq k \cdot g(n)$$

而這個就是 Big-O 的正式定義！

$$f(n) \in O(g(n)) \iff \exists k > 0, \exists N, \forall n > N, f(n) \leq k \cdot g(n)$$

大家可以發現  $O(g(n))$  其實是一個函數的集合，包含所有滿足以上式子的函數，所以正確的寫法是  $f(n) \in O(g(n))$  而不是  $f(n) = O(g(n))$ ，但現在似乎都可以混用了。

以下用一些例子讓大家更了解這個定義：

### 例題 2

請證明  $3n^2 \in O(n^2)$ 。

#### 證明

取  $k = 3, N = 0$ ，則  $\forall n > 0, 3n^2 \leq 3 \cdot n^2$  成立，得證。  
(當然也可以取  $N = 1, N = 2, \dots$ ，或是取  $k = 4, k = 5, \dots$ )

### 例題 3

請證明  $3n^2 \in O(n^3)$ 。

#### 證明

取  $k = 1, N = 3$ ，則  $\forall n > 3, 3n^2 \leq 1 \cdot n^3$  成立，得證。

### 例題 4

請證明  $3n^2 \notin O(n)$ 。

#### 證明

不論  $k$  取多少，只要  $n > k$ ， $3n^2 \leq k \cdot n$  就不會成立，因此  $N$  不存在，也就是  $3n^2 \notin O(n)$ ，得證。

以上可以發現，要說  $n^2 \in O(n^3)$  是可以的，因為  $n^3$  的確是  $n^2$  的上界，只是他是個比較鬆的上界。另外， $O(n+1), O(2n)$  等等的寫法也都是可以的，只是一般來說都會寫成等價的  $O(n)$ 。那麼，影片中說的取極限又是怎麼一回事呢？其實只要改寫一下式子就可以了：

$$\begin{aligned} f(n) \in O(g(n)) &\iff \exists k > 0, \exists N, \forall n > N, f(n) \leq k \cdot g(n) \\ &\iff \exists k > 0, \exists N, \forall n > N, \frac{f(n)}{g(n)} \leq k \end{aligned}$$

我們不知道  $N$  是多少，但是一定是個常數，那麼把  $n$  趨近到無限大就一定會超過  $N$  了！於是就得到

$$f(n) \in O(g(n)) \iff \exists k > 0, \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \leq k$$

也就是說，只要  $\frac{f(n)}{g(n)}$  的極限值不是無限大（例如是一個常數），則  $f(n) \in O(g(n))$ ，而這就是影片中講解的判斷方法了。

## 例題 5

請證明或否證  $f(n) \in O(n^3) \iff f(n) \in O((n+1)^3)$ 。

## 證明

對於這類型的命題，我們只須依據定義證明  $A \implies B$  與  $A \impliedby B$  即可。

(1) 先證明  $f(n) \in O(n^3) \implies f(n) \in O((n+1)^3)$ ：

$$\begin{aligned} &\because f(n) \in O(n^3) \\ &\therefore \exists k \geq 0, \lim_{n \rightarrow \infty} \frac{f(n)}{n^3} = k \\ &\therefore \lim_{n \rightarrow \infty} \frac{f(n)}{(n+1)^3} = \lim_{n \rightarrow \infty} \frac{f(n)}{n^3} \cdot \frac{n^3}{(n+1)^3} = \lim_{n \rightarrow \infty} \frac{f(n)}{n^3} \cdot \lim_{n \rightarrow \infty} \frac{n^3}{(n+1)^3} \\ &\quad = k \cdot 1 = k \quad (\text{仍然為一常數}) \\ &\therefore f(n) \in O((n+1)^3) \end{aligned}$$

(2) 接著證明  $f(n) \in O(n^3) \impliedby f(n) \in O((n+1)^3)$ ：

$$\begin{aligned} &\because f(n) \in O((n+1)^3) \\ &\therefore \exists k \geq 0, \lim_{n \rightarrow \infty} \frac{f(n)}{(n+1)^3} = k \\ &\therefore \lim_{n \rightarrow \infty} \frac{f(n)}{n^3} = \lim_{n \rightarrow \infty} \frac{f(n)}{(n+1)^3} \cdot \frac{(n+1)^3}{n^3} = \lim_{n \rightarrow \infty} \frac{f(n)}{(n+1)^3} \cdot \lim_{n \rightarrow \infty} \frac{(n+1)^3}{n^3} \\ &\quad = k \cdot 1 = k \quad (\text{仍然為一常數}) \\ &\therefore f(n) \in O(n^3) \end{aligned}$$

由 (1),(2)，得證  $f(n) \in O(n^3) \iff f(n) \in O((n+1)^3)$ 。

## 例題 6

請證明或否證  $f(n) \in O(2^n) \iff f(n) \in O(3^n)$ 。

## 證明

如果我們直接拿上一個例題的方式來解這個問題，會發現證明途中遇到阻礙， $\lim_{n \rightarrow \infty} \frac{3^n}{2^n}$  並不會收斂成一個常數。此時我們可以猜想，這個命題是錯誤的，我們所須要的僅僅是一個反例，再用反證法來證明否證原命題。

假設命題成立，那麼我們令  $f(n) = 3^n$ ，則

$$\lim_{n \rightarrow \infty} \frac{f(n)}{2^n} = \lim_{n \rightarrow \infty} \frac{3^n}{2^n} = \infty$$

並非收斂為一個常數，矛盾。於是  $f(n) \in O(2^n) \impliedby f(n) \in O(3^n)$  不成立，從而

原命題不成立。

## 習題

1. 請回答以下問題：

(a) (10 pts) 請計算  $\lim_{n \rightarrow \infty} \frac{3n+1}{n-1}$

(b) (10 pts) 請計算  $\lim_{n \rightarrow \infty} \frac{n}{n^2+1}$

(c) (20 pts) 請證明或否證  $f(n) \in O(2^n) \iff f(n) \in O(2^{n+1})$

(d) (20 pts) 請證明或否證  $f(n) \in O(n!) \iff f(n) \in O((n+1)!)$

(e) (20 pts) 請證明或否證  $f(n) \in O(n) \implies 2^{f(n)} \in O(2^n)$

2. (20 pts) 已知一遞迴函數

$$f(n) = \begin{cases} 1 & , n = 1 \\ 2f(\lfloor \frac{n}{2} \rfloor) + 2n & , n > 1 \end{cases}$$

證明對所有滿足  $n = 2^m$ ，其中  $m$  為正整數的  $n$ ， $f(n) \leq 3n \log_2 n$ 。

3. (20 pts) 已知一遞迴函數

$$f(n) = \begin{cases} 1 & , n = 1 \\ f(\lfloor \frac{n}{2} \rfloor) + f(\lceil \frac{n}{2} \rceil) + n^2 & , n > 1 \end{cases}$$

證明對所有正整數  $n$ ， $f(n) \leq 2n^2 - 1$ 。(此時代表  $f(n) \in O(n^2)$ )