Chapter 2

Methods of Proof

學會了簡單的邏輯後, 接下來便是學習如何證明. 在本章中我們將介紹一些證明的方法. 這裡我們只談有關證明的一些基本原則, 而不談證明的技巧, 所以給的例子將會挑選淺顯易懂的證明.

2.1. IF-Then 的證明

在數學中最常看到的就是這種 $P \Rightarrow Q$ 的 statement. 要證明這種 statement, 我們大致上有 **direct method**, **contrapositive method** 和 **contradiction method** 三種方法.

2.1.1. Direct Method. 所謂 direct method 指的就是直接證明, 也就是直接利用 P 成立的假設得到 Q 成立. (再次強調, 我們不必管 P 不成立時, Q 會如何). 當我們要證明 $P \Rightarrow Q$ 時, 若覺得 P 的條件已經足夠, 便可以考慮使用直接證明. 例如以下的例子.

Example 2.1.1. 令 p,a,b 為整數. 證明 if $p \mid a$ and $p \mid b$, then $p \mid a+b$.

Proof. 由假設 $p \mid a, p \mid b$, 知存在整數 m,n 使得 $a = pm, b = pn$. 故得

$$a+b = pm+pn = p(m+n).$$

因 $m+n$ 為整數, 得證 $p \mid a+b$. □

有時用直接證明的方法並不能一次到位, 需要借助其他的結果幫忙才能完成. 也就是說或許我們不能直接證出 $P \Rightarrow Q$, 但若證得 $P \Rightarrow R$ 又證得 $R \Rightarrow Q$, 此時便證得 $P \Rightarrow Q$ 了. 這是因為若證得 $P \Rightarrow R$, 表示 P 對的話 R 一定對, 再由 $R \Rightarrow Q$ 知 R 對的話 Q 一定對, 故連結而知 P 對則 Q 一定對, 得證 $P \Rightarrow Q$. 這種利用遞移性的證明通常有一部分是一些常用的性質或是一些 (輔助) 定理。例如以下的例子.

Example 2.1.2. 設 a 為正實數且 $a \neq 1$. 已知若 $a^z = 1$, 則 $z = 0$. 證明若 x,y 為實數滿足 $a^x = a^y$, 則 $x = y$.

Proof. 由於 $a \neq 0$, 對於任何實數 y, 我們知 $a^y \neq 0$, 故由 $a^x = a^y$, 等號兩邊除以 a^y 得 $a^{x-y} = 1$. 又因 $a \neq 1$, 我們知道若 $a^z = 1$, 則 $z = 0$. 故由 $a^{x-y} = 1$ 可得 $x - y = 0$, 得證 $x = y$. □

這個證明的例子中, 我們其實是先證得 $(a^x = a^y) \Rightarrow (a^{x-y} = 1)$, 再由 $(a^{x-y} = 1) \Rightarrow (x = y)$ 得證 $(a^x = a^y) \Rightarrow (x = y)$. 其中我們用了一個大家都知道的事實, 即當 a 為正實數且 $a \neq 1$, 若 $a^z = 1$, 則 $z = 0$. 這個事實是需要證明的，不過不容易用 direct method 證明，等一下我們會利用 contradiction method 來證明.

有時在 direct method 中我們可以分成好幾種情況, 看看哪些情況符合 P 的條件, 然後證得 Q. 這樣的證明方法有時稱為 *proof in cases.* 例如以下的例子.

Example 2.1.3. 假設 x 為實數. 證明 if $x^2 - 3x + 2 < 0$, then $1 < x < 2$.

Proof. 由 $x^2 - 3x + 2 = (x-1)(x-2) < 0$, 我們知可分成 2 種情況, 即

(1) $(x-1) < 0$ and $(x-2) > 0$;

(2) $(x-1) > 0$ and $(x-2) < 0$.

(1) 的情況表示 $x < 1$ 且 $x > 2$. 由於沒有實數 x 會同時滿足 $x < 1$ 以及 $x > 2$, 我們知 (1) 不可能成立, 故推得 (2), 即 $x > 1$ 且 $x < 2$. 證得 $1 < x < 2$. □

注意, 在這個證明中, 有些同學或許會疑惑為什麼是排除 (1), 而不直接驗證 (2) 可得 $x^2 - 3x + 2 < 0$ 呢? 這是錯誤的. 在我們的證明中明明白白表示: 若 x 滿足 $x^2 - 3x + 2 < 0$, 那麼 x 一定會滿足 (1) 或是 (2). 排除 (1) 表示只有 (2) 會對, 所以確定若 x 滿足 $x^2 - 3x + 2 < 0$, 那麼 x 一定滿足 (2). 若僅說 x 滿足 (2) 可得 $x^2 - 3x + 2 < 0$, 而沒有排除 (1), 那麼這是證明 if $1 < x < 2$, then $x^2 - 3x + 2 < 0$, 而不是證 if $x^2 - 3x + 2 < 0$, then $1 < x < 2$. 千萬別搞錯.

Question 2.1. 假設 x 為實數.

(1) *"If* $x^2 - 3x + 2 < 0$, *then* $0 < x < 3$*."* 和 *"If* $x^2 - 3x + 2 < 0$, *then* $1.3 < x < 1.7$*."* 這兩個 *statements* 哪一個是對的*?*

(2) *"If* $0 < x < 3$, *then* $x^2 - 3x + 2 < 0$*."* 和 *"If* $1.3 < x < 1.7$, *then* $x^2 - 3x + 2 < 0$*."* 這兩個 *statements* 哪一個是對的*?*

有時在論證的問題有多種結論例如要論證 $P \Rightarrow Q \wedge R$ 或是 $P \Rightarrow Q \vee R$. 當要論證 $P \Rightarrow Q \wedge R$，我們當然就是證明 P 對則 Q 和 R 都會對。但要證明 $P \Rightarrow Q \vee R$, 就會有點麻煩。因為 $Q \vee R$ 會對表示有可能 Q 對 R 錯；Q 錯 R 對，甚至 Q,R 都對。也因此要推得 $Q \vee R$ 會對，感覺有點麻煩。接下來這個技巧就很好用。我們可以直接僅考慮已知 P 對且 Q 錯的情況，因為如果 Q 對那自然 $Q \vee R$ 是對的。因此，此時我們就只要證明由 P 對 Q 錯可推得 R 對即可。從邏輯來看，因為 $(P \Rightarrow (Q \vee R)) \sim (\neg P \vee (Q \vee R))$, 而 $((P \wedge \neg Q) \Rightarrow R) \sim (\neg(P \wedge \neg Q) \vee R) \sim ((\neg P \vee Q) \vee R))$ 所以 $(P \Rightarrow (Q \vee R)) \sim ((P \wedge \neg Q) \Rightarrow R)$. 我們看以下的例子。

Example 2.1.4. 證明若 $xy=0$, 則 $x=0$ 或 $y=0$.

Proof. 我們可假設 $xy=0$ 且 $x\neq 0$, 此時因 $x\neq 0$, 可將等式 $xy=0$ 的兩邊除以 x (或說乘以 $1/x$), 得 $y=0$. □

當然了，在證明 $P\Rightarrow(Q\vee R)$ 時，若覺得 $\neg R$ 比較好用，當然可改成證 $(P\wedge\neg R)\Rightarrow Q$.

2.1.2. Contrapositive Method. 我們稱 $(\neg Q)\Rightarrow(\neg P)$ 為 $P\Rightarrow Q$ 這個 statement 的 *contrapositive statement.* 回顧一下, 我們知道 $P\Rightarrow Q$ 和 $(\neg Q)\Rightarrow(\neg P)$ 是 logically equivalent (參見式子 (1.11)). 也就是說 $P\Rightarrow Q$ 和 $(\neg Q)\Rightarrow(\neg P)$ 的對錯是一致的. 因此, 若我們能證明 $(\neg Q)\Rightarrow(\neg P)$ 便證得 $P\Rightarrow Q$ 了.

當要證明 $P\Rightarrow Q$ 時, 若發現 P 的條件似乎不容易幫助我們證明, 而 $\neg Q$ 較容易處理時便可以考慮使用 contrapositive method. 也就是說證明 $(\neg Q)\Rightarrow(\neg P)$. 最常發生的情況就是有不等式的情形. 因為不等式不如等式使用方便, 很多不等式的使用規則其實都是用等式推導的, 所以如果一個 statement 牽涉到不等式, 而它的 contrapositive statement 是等式. 那麼自然用 contrapositive method 會比較容易證明. 我們有以下的例子.

Example 2.1.5. 設 x,y 為實數. 證明 if $x\neq y$, then $x^3\neq y^3$.

當然了, 若你了解 $f(x)=x^3$ 的圖形或利用微積分, 可以知道這一定對的. 不過我們想要用比較基礎的方法處理. 若用 contrapositive method 來證明, 就是先假設 $\neg(x^3\neq y^3)$ (即 $x^3=y^3$), 要證得 $\neg(x\neq y)$ (即 $x=y$).

Proof. 利用 contrapositive method, 首先假設 $x^3=y^3$, 即 $0=x^3-y^3=(x-y)(x^2+xy+y^2)$. 由 Example 2.1.4 可得 $x-y=0$ 或 $x^2+xy+y^2=0$. 因此可用 proof in cases 處理。

(1) $x-y=0$：此時即 $x=y$.

(2) $x^2+xy+y^2=0$: 此時由

$$x^2+xy+y^2=(x+\frac{1}{2}y)^2+\frac{3}{4}y^2.$$

可得 $x+\frac{1}{2}y=0$ 且 $y=0$. 因此得 $x=y=0$.

由於所有情況皆得 $x=y$, 故得證 if $x^3=y^3$ then $x=y$, 也因此得證 if $x\neq y$, then $x^3\neq y^3$. □

當我們利用 contrapositive method 把要證明的 $P\Rightarrow Q$ 改為證明 $\neg Q\Rightarrow\neg P$ 後，就可以用前面所提的 direct method 的方法證明 $\neg Q\Rightarrow\neg P$. 在上例中，我們就是利用遞移性以及 proof in case 處理。

若一個 statement 是由包含 x 的式子的性質, 來推得 x 本身的性質, 由於通常是反過來將 x 代入式子比較容易，此時也是用 contrapositive method 的好時機. 例如以下的簡單例子.

Example 2.1.6. 設 x 為整數. 證明 if x^2 is even (偶數), then x is even.

Proof. 用 contrapositive method, 即證明若 x 為奇數, 則 x^2 為奇數. 然而 x 為奇數, 表示 x 可以寫成 $x=2n+1$, 其中 n 為整數. 故得證 $x^2=(2n+1)^2=4n^2+4n+1=2(2n^2+2n)+1$ 為奇數. □

Question 2.2. 假設 x,y 為整數. 試用 *contrapositive method* 證明 *if $x+y$ is even, then x and y are both even or odd.*

有時用 proof in cases 處理的情況, 也可用 contrapositive method 來證明. 例如 Example 2.1.3 就可以用 contrapositive method 證明. 也就是說, 先假設 $\neg(1<x<2)$ (即 $x\geq 2$ or $x\leq 1$) 求得 $\neg(x^2-3x+2<0)$ (即 $x^2-3x+2\geq 0$). 我們看另外的例子.

Example 2.1.7. 設 x,y,a 為正實數. 證明 if $xy=a$, then $x\leq\sqrt{a}$ or $y\leq\sqrt{a}$.

用 proof in cases 證明，需考慮 x,y 所有可能與 $\sqrt{a}$ 的大小關係，要分成好幾種情況。而用 contrapositive method, 我們只要考慮一種情況。

Proof. 用 contrapositive method, 先假設 $\neg((x\leq\sqrt{a})\vee(y\leq\sqrt{a}))$ (即 $x>\sqrt{a}$ and $y>\sqrt{a}$) 要證得 $\neg(xy=a)$ (即 $xy\neq a$). 然而 x,y,a 為正, 故由 $x>\sqrt{a}$ and $y>\sqrt{a}$ 可得 $xy>(\sqrt{a})^2=a$, 得證 $xy\neq a$. □

Question 2.3. 利用類似 *Example* 2.1.4 中證明 $P\Rightarrow(Q\vee R)$ 的方法，證明 *Example* 2.1.7.

Question 2.4. 設 x,y,a 為正實數. 請問 *if $xy=a$, then $x\geq\sqrt{a}$ or $y\geq\sqrt{a}$* 是否為對*?* 若為對, 此和 *Example 2.1.7,* 是否有矛盾*?*

2.1.3. Contradiction Method. 回顧式子 (1.10) 告訴我們 $P\Rightarrow Q$ 和 $Q\vee\neg P$ 是 logically equivalent. 也就是說, 若我們能證明 $Q\vee\neg P$ 就等同證明了 $P\Rightarrow Q$. 然而 $Q\vee\neg P$ 是“或”的情況, 處理起來有點像前面提過的 proof in cases, 沒有太多的優勢. 不過若考慮 $\neg(P\Rightarrow Q)$, 此時和 $(\neg Q)\wedge P$ 為 logically equivalent (參見式子 (1.12)). 因此若能證明 $(\neg Q)\wedge P$ 一定是錯的, 便證得 $P\Rightarrow Q$ 為對. 這就是所謂的 contradiction method.

Contradiction method 的處理方法就是, 先假設 $(\neg Q)$ 和 P 皆為對, 再從中推導出與我們知道一定對的 statement 相矛盾. 如此一來便表示 $(\neg Q)$ 和 P 是錯的, 而得證 $P\Rightarrow Q$. 這個方法的最大優點就是它一次便同時假設 P 和 $\neg Q$ 為對, 給我們較多的資訊去推導, 而不像 direct method 僅假設 P 為對, 或 contrapositive method 僅假設 $\neg Q$ 為對. 它的缺點就是, 不像 direct method 或 contrapositive method 明確地知道要推導甚麼 (即由 P 推導出 Q 或由 $\neg Q$ 推導出 $\neg P$), 而是要推導出一個“未知”的矛盾.

當要證明 $P\Rightarrow Q$ 時, 若發現單獨 P 的條件或是單獨 $\neg Q$ 的條件似乎不容易幫助我們證明, 這時便可以考慮 P 和 $\neg Q$ 的條件可同時使用的 contradiction method. 我們看以下的例子.

Example 2.1.8. 設 r 為實數, 證明 if $r^2=2$, then r is irrational (無理數).

我們幾乎沒有直接的方法證明一個數是無理數, 所以不可能用 direct method. 而若用 contrapositive method 雖可先假設 r 為有理數, 但要推得 $r^2 \neq 2$ 這個不等式. 前面已提過不等式的推導並不容易, 所以我們用 contradiction method 來證明.

Proof. 用 contradiction method, 即假設 r 為有理數且滿足 $r^2 = r$, 希望能得到矛盾. 依假設 r 為有理數, 表示 r 可以寫成 $r = (m/n)$, 其中 m,n 為整數. 現若 m,n 皆為偶數, 我們可以約掉 2, 如此一直下去, 我們可假設 m,n 為一奇一偶或皆為奇數. 然而 $r^2 = 2$, 即 $m^2 = 2n^2$ 為偶數, 故由 Example 2.1.6 知 m 必為偶數. 也就是說 m 可寫成 $m = 2m'$, 其中 m' 為整數. 此時得 $4m'^2 = 2n^2$, 即 $n^2 = 2m'^2$ 為偶數. 故再由 Example 2.1.6 知 n 亦為偶數. 此與當初假設 m,n 為一奇一偶或皆為奇數相矛盾. 故得證 if $r^2 = 2$, then r is irrational. □

從 Example 2.1.8 中我們應可體會, 當初若沒有想到將 $r = (m/n)$ 中分子分母的 2 約乾淨, 就無法推出矛盾了. 所以用 contradiction method 證明的困難處就是要 "製造矛盾".

回顧在 Example 2.1.2 的證明中, 我們用了一個事實, 即當 $a \neq 1$ 且為正實數時, 若 z 為實數 $a^z = 1$, 則 $z = 0$. 我們可以用 contradiction method 來證明這個 statement.

Example 2.1.9. 設 $a \neq 1$ 且為正實數. 證明若 z 為實數滿足 $a^z = 1$, 則 $z = 0$.

若要用 contradiction method, 我們必須假設 $z \neq 0$ 且 $a^z = 1$ 而得到矛盾. 要如何製造矛盾呢? 我們知道這個 statement $a \neq 1$ 是重要的前提 (否則它會錯), 所以矛盾的關鍵是 $a \neq 1$.

Proof. 我們利用 contradiction method, 先假設 $z \neq 0$ 且 $a^z = 1$. 此時由於 $z \neq 0$, 我們知 $1/z$ 是存在的, 故利用 $(a^z)^{1/z} = a$ 以及 $a^z = 1$, 得

$$a = (a^z)^{1/z} = 1^{1/z} = 1.$$

此與已知 $a \neq 1$ 相矛盾, 得證若 $a^z = 1$, 則 $z = 0$. □

Remark 2.1.10. 在證明 $P \Rightarrow Q$ 時，其實用 contrapositive method 或 contradiction method 它們第一步是一樣的，也就是先假設 $\neg Q$. 接著我們就可以判斷是否條件夠我們推導出 $\neg P$. 若條件夠，那就用 contrapositive method. 若條件不夠，則可以加上 P 的條件，看看是否可推得矛盾，這就是 contradiction method. 這兩種方法，各有很多種中文名稱，甚至兩種分別都有人稱之為 "反證法"。為了方便起見，以後我不太想去區分這兩種方法，我都一律稱之為反證法。也就說只要是要用 $\neg Q$ 這樣的條件來證明，我都一律稱之為「反證法」。

2.1.4. If and Only If 的證明. $P \Leftrightarrow Q$ 的證明基本上就是要證明 $P \Rightarrow Q$ 和 $Q \Rightarrow P$. 大家或許看過有些證明由於每一個步驟逆推回去也是對的, 所以在推導完 $P \Rightarrow Q$ 後便說反向亦然, 而得證 $P \Leftrightarrow Q$. 在這裡特別提醒大家, 除非你確認每一個步驟逆推回去也是對的, 千萬不要隨便認定反向亦然, 就說得證 $P \Leftrightarrow Q$. 尤其在以後許多進階一點的定理, 很可能兩邊的推導方式是用到完全不同的概念或原理. 所以證明 $P \Leftrightarrow Q$ 還是要分別證明 $P \Rightarrow Q$ 和 $Q \Rightarrow P$ 為宜.

例如設 a 為整數要證明 "a^2 為偶數 $\Leftrightarrow$ a 為偶數", 大家可能會先證 $\Leftarrow$ 這個方向. 此時可令 $a=2n$, 其中 n 為整數, 便得 $a^2=(2n)^2=4n^2$ 為偶數. 但這個推導方式, 逆推就有問題了. 因為知道 a^2 為偶數, 為何 a^2 一定可以寫成 $4n^2$ 這個樣子呢? 所以 $\Rightarrow$ 這個方向還是要推導的 (我們在 Example 2.1.6 已證明過了).

由於 $(P\Rightarrow Q)\sim((\neg Q)\Rightarrow(\neg P))$ 且 $(Q\Rightarrow P)\sim((\neg P)\Rightarrow(\neg Q))$ 我們知 $P\Leftrightarrow Q$ 和 $(\neg P)\Leftrightarrow(\neg Q)$ 為 logically equivalent. 有的同學會覺得若證明 $P\Leftrightarrow Q$ 較麻煩, 可考慮證明 $(\neg P)\Leftrightarrow(\neg Q)$. 其實這是沒必要的, 不管證明哪一個都需要經過一樣的過程. 例如假設 a,b 為整數, 證明 "ab is even $\Leftrightarrow$ a is even or b is even" 和證明 "ab is odd $\Leftrightarrow$ a and b are odd" 其實是一樣的. 即使後面那一個看起來比較簡潔, 但是不管證明哪一個, 證明的過程都是一樣的. 反倒是有的同學可能會證明了 $(\neg Q)\Rightarrow(\neg P)$ 後又去證 $P\Rightarrow Q$. 重複證明了同一件事而不自知, 誤以為證得了兩個方向, 千萬要注意. 所以基本上在證明若且唯若的 statement 時, 最好表明目前在證明哪一個方向. 這樣自己較不會弄錯, 看證明的人也較清楚整個證明過程. 兩全其美, 何樂而不為呢?

數學上也經常會有類似這樣的 statement:

$$\text{The following are equivalent.}\quad (1)\,P;\quad (2)\,Q;\quad (3)\,R.$$

(有時不只會有 P,Q,R 三項, 可能會有更多項). 這個意思就是說

$$P\Leftrightarrow Q,\quad Q\Leftrightarrow R\quad\text{and}\quad R\Leftrightarrow P.$$

雖是如此, 我們不必 "六" 個 $\Rightarrow$ 都證. 事實上僅要證明

$$P\Rightarrow Q,\quad Q\Rightarrow R\quad\text{and}\quad R\Rightarrow P$$

即可. 這是因為 $Q\Rightarrow P$ 的部分, 可由 $Q\Rightarrow R$ 以及 $R\Rightarrow P$ 得到, 而 $R\Rightarrow Q$ 的部分, 可由 $R\Rightarrow P$ 以及 $P\Rightarrow Q$ 得到. 最後 $P\Rightarrow R$ 的部分, 可由 $P\Rightarrow Q$ 以及 $Q\Rightarrow R$ 得到. 當然了, 有時不一定這個順序好證, 也可考慮倒過來證明

$$R\Rightarrow Q,\quad Q\Rightarrow P\quad\text{and}\quad P\Rightarrow R.$$

甚至有時候會發生不管哪一邊都不好證, 例如 $P\Leftrightarrow R$ 兩邊都不好證, 這時證明

$$P\Leftrightarrow Q\quad\text{and}\quad Q\Leftrightarrow R$$

也可. 因為 $P\Rightarrow R$ 的部分, 可由 $P\Rightarrow Q$ 以及 $Q\Rightarrow R$ 得到, 而 $R\Rightarrow P$ 的部分, 可由 $R\Rightarrow Q$ 以及 $Q\Rightarrow P$ 得到. 總之, 在證明這一類的問題, 要注意推導的方向, 確保任兩個 statement 都可通行無阻. 時時標明目前是從哪一個 statement 推導到哪一個 statement, 是必要的.

2.2. Existence and Uniqueness 的證明

Existence 指的是存在性, 而 uniqueness 指的是唯一性. 這兩個性質的探討也經常在數學定理中出現. 要注意, existence 和 uniqueness 是兩個互相獨立的性質. 也就是說存在未必會唯一. 而這裡的唯一指的是若存在則會唯一, 所以證得唯一也未必會存在. 所以這裡, 我們分開討論 existence 和 uniqueness 的證明.

2.2.1. Existence. 有關 existence 的證明方法大致上有兩類. 一類是所謂的 *constructive method* 指的是確實告知存在的是什麼. 另一類是 *nonconstructive method* 指的是利用已知的理論或邏輯的推導得知一定存在, 但未必知道有哪些.

例如證明存在實數 x 滿足 $6x^2-x-1=0$. 我們可以將 $6x^2-x-1$ 分解得 $(2x-1)(3x+1)$ 故明確找出 $x=1/2$ (或 $x=-1/3$) 這個實數會滿足 $6x^2-x-1=0$. 這就是一個 construct method. 我們也可考慮多項式函數 $f(x)=6x^2-x-1$, 發現 $f(0)=-1<0$ 且 $f(1)=4>0$. 故由多項式函數為連續函數以及連續函數的中間值定理, 證得 $f(x)=0$ 在 $0<x<1$ 之間必有一根, 而證得了存在性. 這個方法雖證出存在性, 但因無法明確指出哪一個 x 會滿足 $6x^2-x-1=0$, 所以是 nonconstructive method. 我們再看以下的例子.

Example 2.2.1. 證明 there exists irrational numbers a,b such that a^b is rational.

Proof. Constructive Method: 考慮 $a=\sqrt{2}$ 且 $b=\log_2 9$. 我們已知 a 為無理數. 同樣的利用反證法可證明 b 亦為無理數. 事實上若存在 m,n 為整數滿足 $\log_2 3=n/m$. 表示 $2^n=3^m$, 我們知道 3 的任何整數次方都不會是偶數, 得到矛盾. 故知 $b=\log_2 9$ 為無理數. 此時

$$a^b=2^{\frac{1}{2}\log_2 9}=2^{\log_2 3}=3,$$

得證存在無理數 a,b 使得 a^b 為有理數.

Nonconstructive Method: 考慮 $a'=\sqrt{2}$ 且 $b'=\sqrt{2}$. 我們知道 a',b' 為無理數. 令 $c=(a')^{b'}$. 若 c 為有理數, 則 $a=\sqrt{2},b=\sqrt{2}$ 為所求. 而若 c 為無理數, 此時令 $a=c,b=\sqrt{2}$, 則

$$a^b=(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}}=\sqrt{2}^2=2.$$

得證存在無理數 a,b 使得 a^b 為有理數. □

這裡這個 nonconstructive method 由於沒有證明 $\sqrt{2}^{\sqrt{2}}$ 是否為有理數. 因此無法確定 $a=b=\sqrt{2}$ 和 $a=\sqrt{2}^{\sqrt{2}},b=\sqrt{2}$ 中哪一個會是符合存在性的要求, 但確定它們兩個其中有一個會符合, 所以是 nonconstructive method. 事實上只要知道 $\sqrt{2}^{\sqrt{2}}$ 是無理數, 令 $a=\sqrt{2}^{\sqrt{2}},b=\sqrt{2}$ 這便是 constructive method. 不過 $\sqrt{2}^{\sqrt{2}}$ 是無理數的證明非常困難 (遠遠超過本講義的範圍). 所以我們避開它的證明, 而仍能證明存在性, 這就是 nonconstructive method 的精神.

Question 2.5. 試找到其他的例子或更一般的方法, 利用 *construct method* 證明 *there exists irrational numbers a,b such that a^b is rational.*

大家可以發現利用 constructive method 得到存在性, 證明的重點並不是在於如何找到這些存在的東西, 而是要確實驗證並說明為何它們符合存在的條件. 例如在證明一個方程式的解存在時, 同學們常常利用等式推導出解的可能值, 而沒有代回驗證是否為解就誤以為證得存在性. 這是同學們常弄錯的, 所以我們特別說明一下. 在求解的推導過程, 往往是假設解存在, 再利用一些等式推導出解的 "可能值". 也就是說, 除非你的推導過程是 "雙向" 皆成立的, 否則你推出的結果是,「若解存在的話, 則它們可能的值」, 未必這些真的是解 (這再次

說明若 P 則 Q 方向性的重要). 這些推出的值只是讓我們縮小要驗證的範圍而已, 此時唯有將這些值代回驗證, 才能確保解的存在. 我們看下面的例子.

Example 2.2.2. 是否存在實數 x 滿足 $\sqrt{3-2x}=x-2$?

若直接假設有解, 則兩邊平方得 $3-2x=x^2-4x+4$, 即 $x^2-2x+1=0$. 得 $x=1$. 這表示若有解, 其解必為 $x=1$. 也就是除了 $x=1$ 可能會是此方程式之解外, 其他實數都不可能是解. 但將 $x=1$ 代回原式, 得 $1=-1$ 不合. 故知不存在實數 x 滿足 $\sqrt{3-2x}=x-2$.

其實使用 nonconstructive method 證明存在性, 一般或多或少會用到反證法. 例如前面提過利用連續函數的中間值定理證明存在實數 x 滿足 $6x^2-x-1=0$. 雖然看似沒用到反證法, 不過中間值定理本身的證明一般來說都會用到反證法. 還有一個常用來證明存在性的方法, 就是 *pigeonhole principle* (鴿籠原理). 它原本是 "Dirichlet's drawer principle", 不過現在許多人習慣用 pigeonhole principle 稱之.

Theorem 2.2.3 (Pigeonhole Principle)**.** 令 n 為正整數. 假設有 n 個鴿籠以及多於 n 隻的鴿子. 若要所有的鴿子住進鴿籠裡, 則一定會有一個鴿籠會有兩隻以上的鴿子.

Proof. 很明顯的, 這個存在性無法用 constructive method. 利用反證法, 原本 statement 是說「存在一個鴿籠會有兩隻以上的鴿子」, 它的否定便是「所有的鴿籠都只有一隻或沒有鴿子」但如此一來表示所有 n 個鴿籠裡的鴿子最多只有 n 隻, 與原本假設有多於 n 隻的鴿子矛盾. 得證必有一個鴿籠會有兩隻以上的鴿子. □

將來我們會碰到利用 pigeonhole principle 證明存在性的問題. 大致上, 只要弄清楚甚麼拿來當鴿子, 甚麼拿來當鴿籠就好. 例如證明任取 6 個整數, 其中一定有兩個整數其除以 5 的餘數相同. 這裡我們可以將 6 個整數當成 6 隻鴿子, 將鴿籠從 0 到 4 編號. 若除以 5 的餘數為 0 就放到 0 號鴿籠, 餘數為 1 就放 1 號鴿籠, 依此類推. 因鴿子個數 6 多於鴿籠的個數 5, 利用鴿籠原理, 我們知必有一個鴿籠裡有兩隻以上的鴿子. 也就是說會有兩個整數除以 5 的餘數相同.

我們可以將這個結果稍微推廣一下. 例如任取 16 個整數, 可以證明其中可找到 4 個整數除以 5 有同樣餘數. 這是因為若這些整數除以 5 餘數為 0,1,2,3,4 的都不超過 3 個, 那麼這些整數總共最多不會超過 15 個, 就和原本假設有 16 個整數相矛盾了. 我們有以下 Theorem 2.2.3 的推廣, 證明就不再贅述了.

Theorem 2.2.4. 令 k,n 為正整數. 假設有 n 個鴿籠以及多於 kn 隻的鴿子. 若要所有的鴿子住進鴿籠裡, 則一定會有一個鴿籠會有 k 隻以上的鴿子.

Question 2.6. 試證明 *Theorem 2.2.4*

最後提醒一下, 鴿籠原理不能處理鴿子數少於或等於鴿籠數的情況 (除非另有特殊條件). 它並沒有說在鴿子數少於或等於鴿籠數的情況下不會有鴿籠有兩隻以上的鴿子. 另外它也沒有說在鴿子數多於鴿籠數的情況之下不會有空籠子. 這些都很容易找到反例, 請不要自己過度解讀這個原理.

2.2.2. Uniqueness. 基本上唯一性的證明是在假設存在的前提之下去證明唯一. 所以唯一性的證明一般和存在性的證明是無關的. 當然了, 如果已知不存在了, 就不必去證明唯一性了. 例如在 Example 2.2.2 中當我們假設 x 為解而推得 $x=1$ 時, 便證明了此方程式若有解則解唯一. 不過後來知道方程式無解, 所以這唯一性就不重要了.

大致上唯一性的證明也分成直接證明與反證法兩種. 直接證明就如前述, 直接說明若東西存在應該是什麼. 而反證法一般用的方法是假設有兩個不同的東西滿足條件, 進而推得矛盾. 我們簡單的用 $\mathbb{R}^2$ 上向量的性質來說明.

Example 2.2.5. 證明 $\mathbb{R}^2$ 中若存在一個向量 $\overrightarrow{O}$ 滿足對任意 $\mathbb{R}^2$ 上的向量 $\overrightarrow{V}$ 皆符合 $\overrightarrow{V}+\overrightarrow{O}=\overrightarrow{V}$, 則 $\overrightarrow{O}$ 是唯一的.

(1) 直接證明: 假設 $\overrightarrow{O}=(x,y)$, 對任意 $\overrightarrow{V}=(a,b)\in\mathbb{R}^2$. 由於 $\overrightarrow{O}$ 須符合 $\overrightarrow{V}+\overrightarrow{O}=\overrightarrow{V}$, 得 $(a,b)+(x,y)=(a+x,b+y)=(a,b)$. 利用向量相等的定義得 $a+x=a, b+y=b$, 即 $x=0, y=0$. 得證 $\overrightarrow{O}$ 若存在, 則必須等於 $(0,0)$.

(2) 反證法: 假設 $\overrightarrow{O},\overrightarrow{Q}\in\mathbb{R}^2$ 且 $\overrightarrow{O}\neq\overrightarrow{Q}$ 皆滿足對任意 $\overrightarrow{V}\in\mathbb{R}^2$,

$$\overrightarrow{V}+\overrightarrow{O}=\overrightarrow{V} \tag{2.1}$$

以及

$$\overrightarrow{V}+\overrightarrow{Q}=\overrightarrow{V} \tag{2.2}$$

考慮 $\overrightarrow{Q}=\overrightarrow{V}$ 的情形代入式子 (2.1) 得 $\overrightarrow{Q}+\overrightarrow{O}=\overrightarrow{Q}$. 同理將 $\overrightarrow{O}=\overrightarrow{V}$ 代入式子 (2.2) 得 $\overrightarrow{O}+\overrightarrow{Q}=\overrightarrow{O}$. 由於 $\overrightarrow{Q}+\overrightarrow{O}=\overrightarrow{O}+\overrightarrow{Q}$, 得 $\overrightarrow{Q}=\overrightarrow{O}$. 此與當初假設 $\overrightarrow{O}\neq\overrightarrow{Q}$ 相矛盾, 得證唯一性.

Question 2.7. 給定 $\overrightarrow{V}\in\mathbb{R}^2$ 試利用直接證法以及反證法證明: $\mathbb{R}^2$ 中若存在一個向量 $\overrightarrow{W}$ 滿足 $\overrightarrow{V}+\overrightarrow{W}=\overrightarrow{O}$, 則 $\overrightarrow{W}$ 是唯一的.

注意在 Example 2.2.5 中的直接證法中, 我們求出 $\overrightarrow{O}$ 若存在, 則 $\overrightarrow{O}=(0,0)$. 如再帶回驗證, 確認 $\overrightarrow{O}=(0,0)$ 確實符合, 我們便也證得存在性了. 這是直接證法的好處. 而反證法就沒辦法推得存在性了. 所以一般不是用直接證明時, 存在性及唯一性的證明是要分開來處理的. 然而將來我們會碰到較抽象的數學問題時, 大多直接證明是行不通的. 此時只好仰賴反證法了.

Example 2.2.6. 證明若存在一個實數 r 滿足, $r^3=3$, 則此實數 r 是唯一的.

Proof. 回顧一下在 Example 2.1.5 中我們證明了, 若 x,y 為實數且 $x\neq y$, 則 $x^3\neq y^3$. 假設 $r\in\mathbb{R}$ 滿足, $r^3=3$ 且 $s\neq r$ 是另一個實數滿足 $s^3=3$, 則利用 Example 2.1.5 的結果得 $3=s^3\neq r^3=3$. 由此矛盾知不可能有另一個實數 s 會滿足 $s^3=3$. □

在 Example 2.2.6 中我們是先證明 $x\neq y$ 時, x,y 不可能都符合某性質, 再利用反證法推得唯一性. 這是我們一般證明唯一性常用的方法. 再強調一次, Example 2.2.6 的證明中我們無法得知是否存在實數 r 滿足 $r^3=3$, 我們只知道若存在的話必唯一. 至於此存在性的證明, 是需要另外利用實數的完備性 (或利用多項式函數 $f(x)=x^3$ 的連續性) 來證明的. 這樣有了存在性和唯一性我們才進一步用符號 $3^{1/3}$ 來表示這個唯一滿足 $r^3=3$ 的實數 r.

2.3. Mathematical Induction

另一個常見的證明方法就是所謂的 "數學歸納法". 其實數學歸納法牽涉到建立整數系的 axiom (公設), 不過這裡我們不去談論這些公設邏輯的問題. 而著重於理解並正確使用數學歸納法. 我們將介紹三種數學歸納法, 雖然它們看起來不大一樣, 不過背後的原理是相同的, 事實上它們是等價的.

數學的理論證明其實根源是一些大家都能接受但無法證明的 axiom (公設). 介紹數學歸納法之前, 我們先了解所謂 *well-ordering principle*. 它是可以用其他的公設來證明的, 不過由於我們目前不想牽涉到這方面的課題. 所以我們直接把 well-ordering principle 當成是一個公設. 也就是說它和我們的直觀吻合, 所以我們相信它而不去證明.

所謂 well-ordering 字面上的解釋是 "好的排序" 的意思, 這個排序原理簡單來說是:「將一些正整數收集起來所成的非空集合中, 一定有一個最小的元素」. 相信大家應該不會覺得這個 principle (原則) 不對吧! 直觀上, 自然數是有最小元素 1 的, 所以有下界. 因此大家應不會懷疑這集合裡會有最小的元素. 問題是這集合可能有無窮多的元素, 我們可能無法真正找出這最小元素來, 不過我們相信它一定存在.

接下來我們來看, 最基本的第一種數學歸納法.

Theorem 2.3.1 (Mathematical Induction)**.** 假設以下兩個 *statement* 是對的

I1: $P(1)$ 成立

I2: 若 $k \in \mathbb{N}$ 且 $P(k)$ 成立, 則 $P(k+1)$ 成立

那麼對任意正整數 n, $P(n)$ 皆成立.

Proof. 由於我們不可能直接證明所有的正整數 n 都會使得 $P(n)$ 成立, 所以我們用反證法. 也就是說假設 (1), (2) 是對的以及「對任意正整數 n, $P(n)$ 皆成立」是錯的來得到矛盾.「對任意正整數 n, $P(n)$ 皆成立」是錯的, 亦即「存在正整數 n, 使得 $P(n)$ 不成立」. 因此我們可以將使得 $P(n)$ 不成立的這些正整數 n 收集起來. 因為它不是空集合, 故由 well-ordering principle 知, 必存在最小的正整數 m 使得 $P(m)$ 不成立. 由 (1) 我們知 $P(1)$ 成立, 故得 $m \neq 1$. 也就是說 m 為大於 1 的正整數. 現由於 $m-1$ 為正整數且 $m-1 < m$, 故由 m 為使得 $P(m)$ 不成立的最小正整數之假設知 $P(m-1)$ 成立. 然而由 (2) 知, 當 $P(m-1)$ 成立時, $P((m-1)+1) = P(m)$ 必成立. 此與 $P(m)$ 不成立之假設相矛盾. 故知不可能存在正整數 n, 使得 $P(n)$ 不成立, 也就是說對任意正整數 n, $P(n)$ 皆成立. □

數學歸納法是很好理解的, 它是說由 (I1) 知 P(1) 是對的, 將 (I2) 代 $k = 1$ 的情況, 故由 $P(1)$ 對可推得 $P(2)$ 是對的. 接著代 $k = 2$ 的情況, 由 $P(2)$ 是對的推得 $P(3)$ 是對的. 這樣一直下去. 所以 $P(1)$ 的起頭非常重要. 另外要強調的是 (I2) 指的是假設 $P(k)$ 對推得 $P(k+1)$ 是對的. 所以它並不是要證明 $P(k)$ 是對的, 你也不必擔心 $P(k)$ 到底對不對, 只要想法子利用 $P(k)$ 是對的假設證出 $P(k+1)$ 是對的. 若你沒辦法單純由 $P(k)$ 是對的推得 $P(k+1)$ 是對的, 那基本上就無法用數學歸納法證明了. 例如考慮多項式 $f(x) = x^2 + x + 41$. 當我們代 $x = 1$ 時 $f(1) = 43$ 為質數. 代 $x = 2$, 得 $f(2) = 47$ 仍為質數. $f(3) = 53$, $f(4) = 61$,

$f(5)=71$ 也都是質數. 或許你會有一股衝動認為 x 代任何的整數 n 都會使 $f(n)$ 為質數. 事實上一直代到 $x=39$ 它都會是質數. 但是光由代的動作, 而不去考慮如何由 $f(k)$ 是質數得到 $f(k+1)$ 是質數, 是沒有辦法用數學歸納法的. 事實上我們可以看出, 當 $x=40$ 時, $f(40)=40^2+40+41=40(40+1)+41$ 是可以被 41 整除的, 所以不是質數.

我們看下一個可以用數學歸納法證明的例子.

Example 2.3.2. 設 a,b 為相異的整數, 利用數學歸納法證明: 對任意的正整數 n 皆有 a^n-b^n 為 $a-b$ 的倍數.

Proof. 我們可以將 a^n-b^n 分解成 $(a-b)(a^{n-1}+a^{n-2}b+\cdots+b^{n-1})$, 得證 a^n-b^n 為 $a-b$ 的倍數. 不過這裡想介紹如何用數學歸納法證明. 首先我們第一步代入 $n=1$ 得 $a-b$. 當然是 $a-b$ 的倍數, 故成立. 第二步是直接假設 a^k-b^k 為 $a-b$ 的倍數, 要推得 $a^{k+1}-b^{k+1}$ 為 $a-b$ 的倍數. 所以我們要想辦法看看 $a^{k+1}-b^{k+1}$ 和 a^k-b^k 的關係. 為了要讓 $a^{k+1}-b^{k+1}$ 和 a^k-b^k 扯上關係, 我們自然將之寫成

$$a^{k+1}-b^{k+1}=aa^k-bb^k=aa^k-ab^k+ab^k-bb^k=a(a^k-b^k)+(a-b)b^k.$$

此時由假設 a^k-b^k 為 $a-b$ 的倍數, 我們可將 a^k-b^k 寫成 $(a-b)m$, 其中 m 為整數. 所以 $a^{k+1}-b^{k+1}=a(a-b)m+(a-b)b^k=(a-b)(am+b^k)$. 得證 $a^{k+1}-b^{k+1}$ 為 $a-b$ 的倍數. 故由數學歸納法知, 對任意的正整數 n, a^n-b^n 為 $a-b$ 的倍數 □

一般來說, 數學歸納法的第一步只是代值驗證, 應該沒問題. 而第二步就是一個若 P 則 Q 的證明, 可以利用前面 2.1 節介紹的方法證明. 在證明過程中若無法馬上看出 $P(k)$ 和 $P(k+1)$ 的關係, 可以嘗試先找出 $P(1),P(2)$ 的關係, $P(2),P(3)$ 的關係, 等. 再推敲出 $P(k)$ 和 $P(k+1)$ 的關係. 另外要謹記, 要單純從 $P(k)$ 成立推出 $P(k+1)$, 在這的過程中不能加入其他的假設. 我們看一個錯誤的例子.

Example 2.3.3. 以下的數學歸納法是要證明任意取 n 個數都會相等. 這個結論當然是錯的, 我們必須找出推論錯誤之處.

第一步: 任取一個數, 因為只有一個當然成立.

第二步: 假設任取 k 個數都會相等, 要證明任取 $k+1$ 個數也都會相等. 現任取 $k+1$ 個數, 將這些數留下一個設其值為 a, 而其他 k 個數放入袋中. 依假設袋中 k 個數都會相等設為 b. 現將袋中取出一個數, 再將當初留下的那個數 a 放入袋中. 依假設此時袋中這 k 個數都應相等, 故得 $a=b$. 因此依數學歸納法, 得證任意取 n 個數都會相等.

這個證明出錯的當然是在第二步. 它假設在袋中取出一個數後仍有 $k-1$ 個數在袋中. 然而當 $k=1$ 時, 此時袋中沒有東西, 最後放入袋中的數 a 無其他的數與之比較, 故無法得知 $a=b$. 所以雖然上面的論述當 $k\geq 2$ 時, 確實由假設 $P(k)$ 證得 $P(k+1)$ 對, 但在 $k=1$ 時就無法由 $P(k)$ 對推得 $P(k+1)$ 對. 這不符合數學歸納法要求對所有的正整數 k, 都要滿足若 $P(k)$ 對, 則 $P(k+1)$ 對. 所以論證是錯誤的。由這個例子我們建議，若要用數學歸納法，通常在驗證 $P(1)$ 對之後，不要馬上去證明 $P(k)\Rightarrow P(k+1)$, 而是想看看如何能單純由 $P(1)$ 是

對的推得 $P(2)$ 是對的。這樣不只能讓我們比較有想法知道如何去證明 $P(k) \Rightarrow P(k+1)$，也可避免如上述的錯誤。

有時有的性質並不會從 $n=1$ 開始就對, 例如當 $n \geq 5$ 時, 證明 $2^n > n^2$. 這裡數學歸納法若只能從 1 開始, 就無法處理了. 事實上, 依照 Theorem 2.3.1 的推論, 我們有以下更一般化的數學歸納法.

Corollary 2.3.4 (Extended Mathematical Induction)**.** 設 m 為整數. 假設以下兩個 *statement* 是對的

EI1: $P(m)$ 成立

EI2: 若 $k \geq m$ 為整數且 $P(k)$ 成立, 則 $P(k+1)$ 成立

那麼對任意大於等於 m 的整數 n 皆會使得 $P(n)$ 成立.

Proof. 事實上，可以學 Theorem 2.3.1 用 well-ordering principle 來證明，不過這裡我們想用 Theorem 2.3.1 來證明。首先令 $Q(n) = P(m+n-1)$. 因已知 $P(m)$ 成立, 故知 $Q(1) = P(m)$ 成立. 亦即 Q 滿足 Theorem 2.3.1 的條件 (I1). 接著我們檢查 Q 是否符合 Theorem 2.3.1 的條件 (I2), 也就是假設 $k \geq 1$ 為整數且 $Q(k)$ 成立, 是否可推得 $Q(k+1)$ 成立。現假設 $k \in \mathbb{N}$ 且 $Q(k)$ 成立, 此時 $m+k-1 \geq m$ 為整數, 且 $P(m+k-1) = Q(k)$ 成立, 故由 (EI2) 的假設得 $P(m+k) = Q(k+1)$ 成立. 我們證得, 若 $k \in \mathbb{N}$ 且 $Q(k)$ 成立, 則 $Q(k+1)$ 成立. 故由 Theorem 2.3.1 知對所有的 $n' \in \mathbb{N}$, $Q(n') = P(m+n'-1)$ 成立. 因此當 n 為大於等於 m 的整數時, 令 $n = m+n'-1$, 此時 $n' \in \mathbb{N}$, 故得 $P(n) = P(m+n'-1) = Q(n')$ 成立. □

這個 "extended mathematical induction" 我們用 Corollary 稱之, 是因為它可由 Theorem 2.3.1 直接推得. 事實上在 $m=1$ 的情況 Corollary 2.3.4 就是 Theorem 2.3.1, 所以我們知道 Theorem 2.3.1 和 Corollary 2.3.4 是 equivalent 的.

Example 2.3.5. 證明對任意正整數 n, 當 $n \geq 5$ 時, $2^n > n^2$.

Proof. 我們用 extended mathematical induction $m=5$ 且 $P(n)$ 為 $2^n > n^2$ 的情況證明. 當 $n=5$ 時, $2^5 = 32 > 25 = 5^2$, 故 $P(5)$ 成立. 假設 $k \geq 5$ 為整數且 $2^k > k^2$. 因 $2^{k+1} = 2 \times 2^k$, 故由 $2^k > k^2$ 之假設得 $2^{k+1} > 2k^2$. 而 $(k+1)^2 = k^2 + 2k + 1$ 若能證得 $2k^2 > k^2 + 2k + 1$, 則得證 $2^{k+1} > (k+1)^2$, 即 $P(k+1)$ 成立. 然而 $2k^2 > k^2 + 2k + 1$ 等同於 $k^2 - 2k > 1$, 又 $k^2 - 2k = k(k-2)$, 故由 $k \geq 5$ 得知 $k^2 - 2k > 1$. 我們證得了若 $k \geq 5$ 為整數且 $P(k)$ 成立, 則 $P(k+1)$ 成立, 故由 extended mathematical induction (Corollary 2.3.4) 得證對任意大於等於 5 的整數 n 皆會使得 $P(n)$, 即 $2^n > n^2$ 成立. □

Question 2.8. 在 *Example 2.3.5* 的證明中要用到 $k(k-2) > 1$. 不過此式在 $k=3$ 就會成立, 為何 $2^n > n^2$ 需要到 $n \geq 5$ 時才都會成立呢*?*

數學的理論推導，常常是由許多論證利用邏輯堆砌起來。因此在每一步驟，都應清楚這個步驟是在論證什麼。常見的錯誤就是不明究理，盲目的推導，最後連自己的論證對錯

都不知，甚至證出什麼都不曉得。若你能回答 Question 2.8 這樣的問題，那非常的好，表示你能注意到每一步驟在論證什麼。在 Example 2.3.5 我們是證明了當 $k \geq 3$ 時可由 $P(k)$ 成立推得 $P(k+1)$ 成立。它只告訴我們 $P(3)$ 成立的話 $P(4)$ 就會成立，並不表示 $P(3)$ 成立。事實上 $P(3)$, $P(4)$ 都不成立。然而我們知道 $P(5)$ 成立，而又 $5 > 3$, 所以可推得 $P(6)$ 成立，以至於對整個大於等於 5 的整數皆成立。同樣的道理，假如 $P(5)$ 成立，而在推導 $P(k) \Rightarrow P(k+1)$ 的過程中發現只有在 $k \geq 10$ 的情況成立。此時無法推得 $P(6)$ 成立，所以也就無法馬上下結論說 $P(n)$ 會對 $n \geq 5$ 都會成立了。但因為這裡僅剩幾個情況 (即 $6 \leq k \leq 10$) 需檢驗，若驗證它們都成立，當然就能下結論說 $P(n)$ 會對 $n \geq 5$ 都會成立。

Question 2.9. 假設當 $k \geq 10$ 皆會滿足 $P(k) \Rightarrow P(k+1)$. 試寫下在以下情況當 n 大於等於多少時，$P(n)$ 會成立（有可能無法確定）。

(1) $P(9)$ 成立。

(2) $P(11)$ 成立。

(3) $P(8), P(9), P(10)$ 成立。

(4) 當 n 是 3 的倍數時 $P(n)$ 皆成立。

在數學歸納法的證明中有時 $P(k)$ 對的條件不足以直接證明 $P(k+1)$ 對. 例如一些遞迴數列的問題, 有時需要之前更多項才能決定下一項的性質. 事實上當我們由 $P(1)$ 對證得 $P(2)$ 對, 再由 $P(2)$ 對要證 $P(3)$ 時, 其實 $P(1)$ 已經知道是對的, 所以我們不只有 $P(2)$ 對的前提, 我們還有 $P(1)$ 對. 同理在證得 $P(3)$ 對要證明 $P(4)$ 時, 其實 $P(1), P(2)$ 為對的條件也可以用上, 所以我們有以下條件更強的數學歸納法.

Corollary 2.3.6 (Strong Mathematical Induction)**.** 設 m 為整數. 假設以下兩個 *statement* 是對的

SI1: $P(m)$ 成立

SI2: 若 $k \geq m$ 為整數且 $P(m), P(m+1), \ldots, P(k-1), P(k)$ 皆成立, 則 $P(k+1)$ 成立

那麼對任意大於等於 m 的整數 n 皆會使得 $P(n)$ 成立.

Proof. 對於大於等於 m 的整數 n, 令 $Q(n) = P(m) \wedge P(m+1) \wedge \cdots \wedge P(n-1) \wedge P(n)$. 因假設 $P(m)$ 成立, 故知 $Q(m) = P(m)$ 成立，亦即 Q 滿足 Corollary 2.3.4 的條件 (EI1). 接著我們檢查 Q 是否符合 Corollary 2.3.4 的條件 (EI2), 也就是假設 $k \geq m$ 為整數且 $Q(k)$ 成立, 是否可推得 $Q(k+1)$ 成立。然而 $Q(k)$ 成立表示 $P(m), \ldots, P(k-1), P(k)$ 皆成立, 故由 (SI2) 的假設得 $P(k+1)$ 成立. 然而已知 $P(m), P(m+1), \ldots, P(k-1), P(k)$ 皆成立, 故知 $Q(k+1) = P(m) \wedge P(m+1) \wedge \cdots \wedge P(k) \wedge P(k+1)$ 成立. 我們證得, 若 $k \geq m$ 為整數且 $Q(k)$ 成立, 則 $Q(k+1)$ 成立. 故由 Corollary 2.3.4 知對任意大於等於 m 的整數 n 皆會使得 $Q(n)$ 成立. 然而 $Q(n)$ 成立表示 $P(m), P(m+1), \ldots, P(n-1), P(n)$ 皆成立, 自然 $P(n)$ 成立, 故得證當 n 為大於等於 m 的整數時, $P(n)$ 皆成立. □

Remark 2.3.7. Strong Mathematical Induction 的 (SI2) 一般我們可以改寫為：若 $k \geq m$ 為整數且對於滿足 $m \leq i \leq k$ 的整數 i, $P(i)$ 皆成立, 則 $P(k+1)$ 成立。

我們稱 Corollary 2.3.6 為 strong mathematical induction 意即它比 Theorem 2.3.1 強。在數學上，我們稱一個定理比另一個定理強，大致上的意思是它可以在更廣泛的情況使用。例如 Corollary 2.3.4 就比 Theorem 2.3.1 強，因為它可以應用在任意整數 m 起頭的情況，而不只是 1. 通常比較強的定理很容易就可推導出比較弱的。例如 Corollary 2.3.4 只要考慮 $m=1$ 的情況就可得 Theorem 2.3.1. 不過感覺上比較弱的定理，未必就真的比較弱。例如我們是用 Theorem 2.3.1 證得 Corollary 2.3.4, 所以邏輯上來說他們是等價的，沒有誰強誰弱之分。不過在使用上當然 Corollary 2.3.4 比較方便。

同樣的 Corollary 2.3.6 直觀上也比 Corollary 2.3.4 強（因為 (SI2) 的條件比較多比較好使用）。我們可以很容易用 Corollary 2.3.6 來證明 Corollary 2.3.4. 事實上當 P 符合 Corollary 2.3.4 的 (EI1), (EI2), 我們希望證明 P 也會符合 Corollary 2.3.6 的 (SI1), (SI2) 因此由 Corollary 2.3.6 的結論得到 $P(n)$ 對所有 $n \geq m$ 都成立。其實 (EI1) 和 (SI1) 是一樣的，所以只要探討 P 若符合 (EI2) 是否會符合 (SI2), 也就是假設 $P(m),\ldots,P(k-1),P(k)$ 成立是否可得 $P(k+1)$ 成立。然而由這個假設當然表示 $P(k)$ 成立，而又已知 P 符合 (EI2), 也就是 $P(k)$ 成立的話 $P(k+1)$ 一定成立，也因此推得 P 確實符合 (SI2)。另一方面 Corollary 2.3.6的證明是利用 Corollary 2.3.4 證得的，也因此知它們是等價的。最後由等價的遞移性，我們知這裡介紹的三個數學歸納法都是等價的。這是從邏輯的觀點來看（它們沒有強弱之分），實際在證明時當然是挑最適合的來證明。下一個例子我們可以看出，應該用 strong mathematical induction 來證明較合適。

Example 2.3.8. 證明所有大於 1 的整數都可以寫成有限多個質數的乘積.

或許很多同學會用以下的方法證明. 設 n 為大於 1 的整數, 若 n 為質數, 則 n 符合可寫成有限多個質數乘積. 若 n 不是質數, 依定義 n 可以寫成兩個比 n 小但大於 1 的整數相乘, 這樣一直下去可證得可以寫成有限多個質數的乘積. 相信大家都能接受這樣的說法來解釋這個性質是對的. 但它並不是好的證明, 比方說如何“一直下去”且為何這個程序經有限多次後會停止 (這樣才能說是有限多個質數的乘積) 也要說明清楚. 有了數學歸納法, 就是能幫我們解決這些不容易說清楚的地方. 在這個例子, 若我們僅假設 k 可以寫成有限多個質數乘積, 是無法證得 $k+1$ 可以寫成有限多個質數乘積, 所以我們必須用 strong mathematical induction 來證明.

Proof. 當 $n=2$ 時, 因 2 為質數, 故成立. 假設當 $k \geq 2$ 時對所有滿足 $2 \leq i \leq k$ 的整數 i 都可以寫成有限多個質數的乘積. 現考慮 $k+1$ 的情形. 因為 $k+1$ 是質數時自然成立, 所以我們僅需考慮 $k+1$ 不為質數的情形. 此時 $k+1=ab$, 其中 $1<a,b\leq k$. 故由前歸納之假設知 a,b 皆為有限多個質數的乘積, 因此 $k+1=ab$ 自然可以寫成有限多個質數的乘積. 故由 strong mathematical induction 知所有大於 1 的整數都可以寫成有限多個質數的乘積. □

在利用數學歸納法證明的過程中, 最重要且最難的部分就是第二步驟由假設 $P(k)$ 成立 (或 $P(m),P(m+1),\ldots,P(k)$ 成立) 證得 $P(k+1)$ 成立. 這裡常常在推導的過程中發現 k 要在某些範圍內才會對. 例如在 Example 2.3.3 的錯誤示範中, 其實“任取 k 個數相等推得任取 $k+1$ 個數會相等”的證明在 $k \geq 2$ 時是對的. 所以此時若能再補上 $k=1$ 的情況也對, 整個

證明就完成了 (當然在 Example 2.3.3 這個例子這是不可能做到的). 前面提過，當我們在處理第二步驟時, 若發現 k 要有所限制才能對, 此時我們可以多檢查那些 k 無法涵蓋的情況, 若這些情況也都對, 就完成歸納法的證明了. 總之, 在數學歸納法的證明中, 有時並不是僅檢查初始的情況就好, 我們再多看一些例子。

Example 2.3.9. 考慮所謂的 *Fibonacci sequence* $\{F_0, F_1, F_2, \dots\}$, 即 $F_0 = 0, F_1 = 1$ 且對任意 $i \geq 2$, F_i 滿足 $F_i = F_{i-1} + F_{i-2}$. 證明 $F_n < 2^{n-2}$, for $n \geq 4$.

很顯然地, 因 F_{k+1} 的值由 F_k 和 F_{k-1} 所決定, 我們無法僅由 F_k 來推得 F_{k+1}. 所以這裡我們用 strong mathematical induction 來處理. 依定義 $F_2 = F_1 + F_0 = 1, F_3 = F_2 + F_1 = 1 + 1 = 2$, 故當 $n = 4$ 時, $F_4 = F_3 + F_2 = 2 + 1 = 3$, 所以 $F_4 = 3 < 2^{4-2} = 4$ 成立. 現假設 $k \geq 4$ 且對所有 $4 \leq i \leq k$, 皆有 $F_i < 2^{i-2}$. 此時 $F_{k+1} = F_k + F_{k-1}$. 我們希望用到 $F_k < 2^{k-2}$ 和 $F_{k-1} < 2^{(k-1)-2} = 2^{k-3}$ 的假設推得 $F_{k+1} < 2^{(k+1)-2} = 2^{k-1}$. 不過當 $k = 4$ 時, $i = k - 1$ 並不符合 $4 \leq i \leq k$, 所以此時無法使用 $F_{k-1} < 2^{k-3}$ 的假設 (事實上此時 $F_{k-1} = F_3 = 2 = 2^{4-3}$). 所以我們再補上 $k = 4$ 的情況, 即直接驗證 $F_{k+1} = F_5 = F_4 + F_3 = 5 < 2^{5-2} = 8$, 才可完成證明.

Proof. 首先直接驗證得 $F_4 = 3 < 2^{4-2}, F_5 = 5 < 2^{5-2}$.

現假設 $k \geq 5$ 且對任意 $i = 4, 5, \dots, k$ 皆有 $F_i < 2^{i-2}$. 因為 $4 \leq k - 1 \leq k$ 且 $4 \leq k \leq k$, 我們有 $F_k < 2^{k-2}, F_{k-1} < 2^{(k-1)-1} = 2^{k-3}$, 故得

$$F_{k+1} = F_k + F_{k-1} < 2^{k-2} + 2^{k-3} = 2^{k-3}(2+1) < 4 \times 2^{k-3} = 2^{k-1} = 2^{(k+1)-2}.$$

依數學歸納法得證 $F_n < 2^{n-2}$, for $n \geq 4$. □

在下一個例子中, 我們想證明所有大於 20 的整數都可以寫成 4 和 5 的正整數倍之和. 也就是證明若 $n > 20$, 則存在正整數 l, m 使得 $n = 4l + 5m$. 或許同學會想到 $1 = 5 - 4$, 所以若 $k = 4l + 5m$, 則 $k + 1 = 4l + 5m + (5 - 4) = 4(l - 1) + 5(m + 1)$. 不錯, 用這個方法可以證明所有的整數皆可以寫成 4 的倍數和 5 的倍數之和, 但我們要求的是寫成 4 和 5 的正整數倍之和. 因此無法用這方法證明。不過觀察一下發現若 $k = 4l + 5m$, 則 $k + 4n = 4(l + n) + 5m$, 所以我們可以先驗證 $21, 22, 23, 24$ 都可以寫成 4 和 5 的正整數倍之和, 就可利用 proof by cases 將所有大於 20 的正整數分成 $21 + 4n, 22 + 4n, 23 + 4n, 24 + 4n$ 來探討，而得證。根據這個想法，我們用 strong mathematical induction 來證明。

Example 2.3.10. 證明若 n 為整數且 $n > 20$, 則存在 l, m 為正整數滿足 $n = 4l + 5m$.

Proof. 由於 $21 = 4 \times 4 + 5 \times 1$, $22 = 4 \times 3 + 5 \times 2$, $23 = 4 \times 2 + 5 \times 3$ 和 $24 = 4 \times 1 + 5 \times 4$ 故知當 $n = 21, 22, 23, 24$ 時成立. 現假設 $k \geq 24$ 時對於所有滿足 $21 \leq i \leq k$ 的整數 i, 皆存在正整數 l, m 使得 $i = 4l + 5m$. 考慮 $k + 1$ 的情形, 由於 $k + 1 = (k - 3) + 4$, 且 $i = k - 3$ 滿足 $21 \leq i \leq k$, 故存在正整數 l, m 使得 $k - 3 = 4l + 5m$, 得 $k + 1 = 4l + 5m + 4 = 4(l + 1) + 5m$. 由數學歸納法知, 當 n 為大於 20 的整數, 存在 l, m 為正整數滿足 $n = 4l + 5m$. □

數學歸納法是一個很好使用的數學工具. 它不只可以拿來處理整數的問題, 其實很多可以用整數分類的問題都可以用數學歸納法處理. 例如代數中許多有關於多項式的問題, 我們

可以依多項式的次數來做數學歸納法. 線性代數中許多有關矩陣的問題, 也可以用 row 的個數或 column 的個數做數學歸納法。