數學導論

李華介

國立台灣師範大學數學系

前言

大學所學的數學和高中階段所學習數學在層次上有所不同. 簡單來說大學數學著重於理論的基礎. 本講義希望介紹同學如何讀懂定理, 理解證明甚而自己寫下證明. 我們將介紹邏輯 (Logic), 集合 (Set) 以及函數 (Function) 的概念. 要注意, 我們不是要介紹邏輯學以及集合論. 這裡談論的僅著重於將來大家學習數學所需要的邏輯及集合的概念.

本講義雖然主要以中文撰寫, 不過當涉及定義或專有名詞時, 為免翻譯的困擾將以英文取代. 因此將以中英夾雜較不傳統的方式顯現, 若有不便請見諒.

本講義編寫費時, 編寫完後並未經過嚴謹的校對. 疏漏在所難免, 雖不至於有理論性上嚴重的錯誤, 但讀者仍應注意不宜概括全收. 若發現錯誤, 歡迎提出寶貴的意見.

Basic Logic

其實學習數學就像學習新的語言. 一些名詞的定義就像“單字”一樣, 而邏輯 *logic* 就好比是將這些單字組合成一個句子所需的“文法”. 一般同學在學習邏輯時, 會不自覺地將一些邏輯規則以背誦的方式記憶, 這會造成以後學習上許多的障礙. 其實這些規則應該是潛意識內的直覺, 這樣學習數學才能通行無阻.

在數學中能明確知道對或錯的論述我們稱之為 *statement*. 例如 $2 > 0$ 是一個 statement, $3 < 2$ 也是一個 statement 但 $x > 0$ 就不是一個 statement (除非我們知道 x 是什麼).

1.1. Connectives

數個 statements 可以組合成一個 statement, 連接這些 statements 的就是所謂 *connectives*. 我們要探討經由 connectives 連結成的 statement 其對或錯的情形.

1.1.1. And. 首先介紹的便是“and”這一個 connective. 這一個 connective 應該是大家最容易理解的一個. 若 P 和 Q 皆為 statement, 我們用 $P \wedge Q$ 表示「P and Q」這一個 statement. $P \wedge Q$ 什麼時候是對的什麼時候是錯的呢? 按照字面的意義“and”就是“且”的意思, 就如同習慣用語當 P 而且 Q 都是對時我們才能說 $P \wedge Q$ 是對的, 而只要 P 和 Q 其中有一個是錯的, 我們便會說 $P \wedge Q$ 是錯的. 例如「$2 > 0$ and $2 < 7$」是對的, 而「$2 > 0$ 且 $2 > 7$」便是錯的.

我們可利用所謂的真值表 *truth table* 來表示用 connectives 連結兩個 statements 後其對錯的情況. 我們用 T 表示對 (true), F 表示錯 (false). 所以我們有以下的 true table.

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

基本上 Truth table 就是將 P, Q 每個可能對錯的情況列出, 然後由 P, Q 所對應的情況, 寫下它們連接後的對錯情況. 例如上表第三橫排為 P 為 T, Q 為 F 故寫下 $P \wedge Q$ 為 F.

很容易發現不管 P,Q 的對錯情況如何 $P\wedge Q$ 和 $Q\wedge P$ 的對錯情形皆相同. 也就是說 $P\wedge Q$ 和 $Q\wedge P$ 在邏輯上是相等的. 我們稱它們為 *logically equivalent.*

Truth table 可以幫助我們判斷許多 statements 用 connectives 連接起來後其對錯的情況, 例如 $(P\wedge Q)\wedge R$ 的 truth table 為

P	Q	R	$P\wedge Q$	$(P\wedge Q)\wedge R$
T	T	T	T	T
T	F	T	F	F
F	T	T	F	F
F	F	T	F	F
T	T	F	T	F
T	F	F	F	F
F	T	F	F	F
F	F	F	F	F

Question 1.1. 你會列出 $P\wedge(Q\wedge R)$ 的 *truth table* 嗎?

注意 $(P\wedge Q)\wedge R$ 和 $P\wedge(Q\wedge R)$ 在意義上是不一樣的. $(P\wedge Q)\wedge R$ 是先探討 $P\wedge Q$ 的對錯再和 R 連結; 而 $P\wedge(Q\wedge R)$ 是先探討 $Q\wedge R$ 的對錯再和 P 連結. 不過從它們的 truth table 我們知道 $(P\wedge Q)\wedge R$ 和 $P\wedge(Q\wedge R)$ 為 logically equivalent.

1.1.2. Or. 當 P 和 Q 皆為 statement, 我們用 $P\vee Q$ 表示「P or Q」這一個 statement. 按照字面的意義 “or” 就是 “或” 的意思. 不過在我們日常用語中 “或” 有兩種用法: 例如在速食店點套餐, 飲料可以選擇「可樂或果汁」. 這裡的 “或” 表示二者擇一, 你不可以兩個都選; 而遊樂園購票時規定「六歲以下或身高 105 公分以下」才可購買兒童票. 這裡的 “或” 表示六歲以下和身高 105 公分以下二者有一個成立就可以, 並不排除六歲以下且身高 105 公分以下同時成立的情況. 在數學邏輯上, “or” 指的是後面那種說法, 也就是說當 P 和 Q 其中有一個是對的 $P\vee Q$ 便是對的 (並不排除 P 和 Q 皆為對的情況). 換言之, 只有當 P 和 Q 都是錯的, $P\vee Q$ 才是錯的.

例如, 「$4<5$ or $4<3$」這個 statement 是對的, 因為 $4<5$ 是對的. 而「$4>5$ or $4>6$」這個 statement 便是錯的, 因為二者皆不成立. 要注意「$4<5$ or $4>3$」這個 statement 依然是對的, 雖然你會認為用 and 比較好, 不過在邏輯上它依然是對的, 千萬別搞錯.

我們有以下關於 $P\vee Q$ 的 truth table.

P	Q	$P\vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

Question 1.2. $P\vee Q$ 和 $Q\vee P$ 是否為 *logically equivalent?* $(P\vee Q)\vee R$ 和 $P\vee(Q\vee R)$ 是否為 *logically equivalent?*

既然 and, or 皆為 connectives, 我們可以將其混合使用. 例如當 P,Q,R 為 statements 我們可以考慮如 $(P\wedge Q)\vee R$, $(P\vee Q)\wedge R,\ldots$ 等形式的 statements. 如何判定它們的對錯

呢? 例如 $(P \wedge Q) \vee R$ 是對的就必須 $(P \wedge Q)$ 或 R 其中一個是對的. 所以只要是 R 是對的, $(P \wedge Q) \vee R$ 就一定對, 而若 R 是錯的那就必須 P, Q 皆對, $(P \wedge Q) \vee R$ 才會是對的. 注意, 千萬不要誤以為 $(P \wedge Q) \vee R$ 和 $P \wedge (Q \vee R)$ 是 logically equivalent. 很顯然的 $P \wedge (Q \vee R)$ 是對的就必須 P 和 $Q \vee R$ 皆為對的. 例如當 R 是對的時, 不管 Q 為對或錯 $Q \vee R$ 皆為對, 但還必須 P 為對才可得到 $P \wedge (Q \vee R)$ 是對的. 這和只要是 R 是對的, $(P \wedge Q) \vee R$ 就一定對不同, 所以 $(P \wedge Q) \vee R$ 和 $P \wedge (Q \vee R)$ 不是 logically equivalent. 當然我們也可利用以下的 truth table 判定它們不是 logically equivalent.

P	Q	R	$P \wedge Q$	$(P \wedge Q) \vee R$
T	T	T	T	T
T	F	T	F	T
F	T	T	F	T
F	F	T	F	T
T	T	F	T	T
T	F	F	F	F
F	T	F	F	F
F	F	F	F	F

P	Q	R	$Q \vee R$	$P \wedge (Q \vee R)$
T	T	T	T	T
T	F	T	T	T
F	T	T	T	F
F	F	T	T	F
T	T	F	T	T
T	F	F	F	F
F	T	F	T	F
F	F	F	F	F

另一方面, 利用以下 $(P \vee R) \wedge (Q \vee R)$ 的 truth table, 不難發現 $(P \wedge Q) \vee R$ 和 $(P \vee R) \wedge (Q \vee R)$ 為 logically equivalent.

P	Q	R	$P \vee R$	$Q \vee R$	$(P \vee R) \wedge (Q \vee R)$
T	T	T	T	T	T
T	F	T	T	T	T
F	T	T	T	T	T
F	F	T	T	T	T
T	T	F	T	T	T
T	F	F	T	F	F
F	T	F	F	T	F
F	F	F	F	F	F

Question 1.3. 試利用 *truth table* 檢查 $(P \vee Q) \wedge R$ 和 $(P \wedge R) \vee (Q \wedge R)$ 是否為 *logically equivalent.*

我們可以利用 truth table 檢驗一些表示法是否為 logically equivalent. 在一些有關 logic 的書也會有一些 logical equivalences 的列表讓大家檢驗. 不過這些都是為了讓大家熟悉這些 connectives 以及 truth table 的運用. 除了以後和論證有關的 logical equivalences 我們需要注意且會特別提醒大家要熟悉, 一般來說大家不必花時間於記憶這些 logical equivalences.

最後提醒一下和 “or” 有關的數學符號 $\geq$ 和 $\leq$. 在數學上 $x \geq y$ 表示 $x > y$ or $x = y$, 所以 $4 \geq 3$ 這一個 statement 按照 or 的邏輯規則是對的. 同理 $4 \leq 5$, 也是對的.

1.1.3. If - Then. 這是一個數學定理裡常見的 connective 但又是許多同學不甚了解而經常誤解的 connective, 請務必弄清楚. 當 P 和 Q 皆為 statement, 我們用 $P \Rightarrow Q$ 表示「if P then Q」這一個 statement, 即「若 P 則 Q」的意思. 要注意 $P \Rightarrow Q$ 在數學上的意涵與純粹邏輯上有所不同. 主要的區別是, 數學上 $P \Rightarrow Q$ 較常表達的是 P, Q 之間的因果關係 (也就是說 P, Q 通常是相關的). 這裡 P, Q 通常不是 statement, 而是如「x 為實數」這樣的 “性質”.

而邏輯上將 $\Rightarrow$ 看成是一個 connective 可以連結任意的 P,Q (即使它們毫無關係). 例如數學上我們有 "if $x>3$ then $x^2>9$" 這樣的 statement (注意 $x>3$ 和 $x^2>9$ 皆不是 statement, 但用 if-then 連結後, 它是一個 statement). $x>3$ 和 $x^2>9$ 是有關係的. 而在邏輯上在我們有 "if $3>2$ then 2 is even" 這樣的 statement (即使 $3>2$ 和 2 為偶數是沒有關係的). 在探討 $P\Rightarrow Q$ 在邏輯上對錯的情況之前, 我們先強調它在數學理論以及推理與論證上的意涵.

在數學上, 當我們說「if P then Q」意即 "當 P 成立時, Q 一定成立". (注意: 為了區別性質與 statement, 我們說一個性質成不成立, 而不用對錯這樣的說法.) 這裡要強調的是, 當我們說 if P then Q 表示我們僅知道如果 P 成立, 則可確定 Q 一定成立. 如果 P 不成立, 是無法知道 Q 是否成立. 所以在數學上要論述「if P then Q」我們只關心當 P 成立時, Q 是否也成立這樣的 "因果關係", 不必在意 P 不成立的情況. 這一點和邏輯上的「if P then Q」看成 P,Q 這兩個 statements 的 connective 相當的不同, 因為既然要讓「if P then Q」成為一個 statement, 就必須明定 P,Q 在任何的對錯情況時 $P\Rightarrow Q$ 的對錯情況. 另外我們也要強調 $P\Rightarrow Q$ 和 $Q\Rightarrow P$ 在數學上是完全不一樣的. 有許多同學會誤以為可由 $P\Rightarrow Q$ 推得 $Q\Rightarrow P$. 這是不對的, 事實上 $P\Rightarrow Q$ 僅表示由 P 成立可推得 Q 成立, 但不表示當 P 不成立時不會使得 Q 成立. 例如我們知道 if $x>3$ then $x^2>9$, 但這並不表示當 $x\leq 3$ 時不會使得 $x^2>9$. 也就是說我們無法由 Q 成立得到 P 成立. 總而言之, $P\Rightarrow Q$ 並不能確保 $Q\Rightarrow P$. 等一下我們定義「if P then Q」在邏輯上的對錯情況時, 我們也會發現 $P\Rightarrow Q$ 和 $Q\Rightarrow P$ 在邏輯上也不是 equivalent.

Question 1.4. 如果我們知道 P 成立則 Q 成立. 那麼當我們發現 Q 不成立時, 是否可以斷言 P 也不成立?

現在我們來看在邏輯上如何定義 $P\Rightarrow Q$ 的對錯情況. 從前面數學上的意義來看, 當 P,Q 為 statements 時, 如果 P 是對的且 Q 是對的, 那麼並未違背 $P\Rightarrow Q$ 的說法, 所以在這種情況我們定 $P\Rightarrow Q$ 為對. 但若 P 是對的而 Q 是錯的, 那麼就違背 $P\Rightarrow Q$ 的說法, 所以在這種情況我們定 $P\Rightarrow Q$ 為錯. 但是若 P 是錯的, 如何定 $P\Rightarrow Q$ 的對錯呢? 由於 $P\Rightarrow Q$ 並未論及當 P 是錯時, Q 會如何, 所以當 P 是錯時, 不管 Q 的對錯都未違背前述 $P\Rightarrow Q$ 的說法, 所以此時我們都定義 $P\Rightarrow Q$ 為對. 例如 $2>3$ 是錯的且 $2^2>9$ 是錯的, 但這並不違背前面所提 if $x>3$ then $x^2>9$ 這一個對的 statement. 另一方面, $-4>3$ 是錯的, 但 $(-4)^2>9$ 是對的, 也不違背前述 if $x>3$ then $x^2>9$ 這一個對的 statement. 總而言之, 關於 $P\Rightarrow Q$ 我們有以下的 truth table.

P	Q	$P\Rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

Question 1.5. 試利用 *truth table* 判斷 $Q\Rightarrow P$ 和 $P\Rightarrow Q$ 是否為 *logically equivalent?* $(P\Rightarrow Q)\Rightarrow R$ 是否和 $P\Rightarrow(Q\Rightarrow R)$ 為 *logically equivalent?*

或許有些同學對 $P\Rightarrow Q$ 的對錯情況為何這麼定義仍有疑慮, 在我們介紹 "if and only if" 這個 connective 時會再進一步說明.

最後我們補充 $P \Rightarrow Q$ 在英文上的幾種說法. 除了「if P then Q」外, 還有

- 「Q if P」
- 「P implies Q」
- 「P is sufficient for Q」(意即 P 成立足以使得 Q 成立)
- 「Q is necessary for P」(意即需要 Q 成立才有可能使得 P 成立)
- 「P only if Q」(意即只有當 Q 成立時 P 才可能成立)
- 「Q whenever P」(意即每當 P 成立時 Q 都會成立)

1.1.4. If and Only If. 當我們將 $P \Rightarrow Q$ 和 $Q \Rightarrow P$ 用 and 連接時, 即 $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$, 我們稱之為 "$P$ if and only if Q", 用 $P \Leftrightarrow Q$ 來表示.

我們依然先探討在數學上 $P \Leftrightarrow Q$ 的意義. 依定義在數學上我們說 $P \Leftrightarrow Q$ 表示 $P \Rightarrow Q$ 且 $Q \Rightarrow P$. 也就是說若 P 成立則 Q 一定成立, 另一方面若 Q 成立則 P 一定成立. 因此 P, Q 有一個成立時另一個一定也成立. 換言之, $P \Leftrightarrow Q$ 表示若 Q 成立則 P 一定成立而且只有當 Q 成立時才會使得 P 成立 (否則會造成 P 成立但 Q 不成立的情況). 這也是在中文我們將 $P \Leftrightarrow Q$ 稱之為 "P 若且唯若 Q" (或 P 當且僅當 Q) 的原因.

現在我們來看在邏輯上 $P \Leftrightarrow Q$ 的對錯情況. 從前面數學上的意義來看, 我們可以知道 $P \Leftrightarrow Q$ 表示 P 對則 Q 且 Q 對則 P 對. 不會有一對一錯的情況. 因此若 P, Q 有一個錯則另一個一定也是錯的. 也就是說在邏輯上 $P \Leftrightarrow Q$ 是對的表示 P 和 Q 必須是同時是對的或同時是錯的. 所以我們有以下關於關於 $P \Leftrightarrow Q$ 的 truth table.

P	Q	$P \Leftrightarrow Q$
T	T	T
T	F	F
F	T	F
F	F	T

Question 1.6. 試利用 $P \Rightarrow Q$ 以及 $Q \Rightarrow P$ 的 *truth table* 寫下 $P \Leftrightarrow Q$ 的 *truth table.*

Question 1.7. $P \Leftrightarrow Q$ 和 $Q \Leftrightarrow P$ 是否為 *logically equivalent?* $(P \Leftrightarrow Q) \Leftrightarrow R$ 和 $P \Leftrightarrow (Q \Leftrightarrow R)$ 是否為 *logically equivalent?*

邏輯上 $P \Leftrightarrow Q$ 對錯的情況, 和數學上的情況很一致, 大家應該覺得較為自然. 現在我們利用 $P \Leftrightarrow Q$ 來解釋為何邏輯上只要 P 是錯的, 不管 Q 的對錯, $P \Rightarrow Q$ 都定義為對的. 當然了, 因為 $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$ 就是 $P \Leftrightarrow Q$, 所以當 P, Q 皆為錯時, 為了讓 $P \Leftrightarrow Q$ 為對, 我們當然要定義 $P \Rightarrow Q$ 和 $Q \Rightarrow P$ 為對. 所以當 P, Q 皆為錯時, 我們定義 $P \Rightarrow Q$ 為對. 至於 P 錯 Q 對的情形, 由於此時 $Q \Rightarrow P$ 為錯, 不管 $P \Rightarrow Q$ 怎麼定都可以使得 $P \Leftrightarrow Q$ 為錯. 然而此時若 $P \Rightarrow Q$ 定為錯, 將會導致 $P \Rightarrow Q$, $Q \Rightarrow P$ 和 $P \Leftrightarrow Q$ 皆有相同的 truth table (亦即 equivalent), 此和前述數學上不能由 $P \Rightarrow Q$ 推得 $Q \Rightarrow P$ 相違背, 所以當 P 錯 Q 對的情形, 我們依然定義 $P \Rightarrow Q$ 為對.

最後我們補充 $P \Leftrightarrow Q$ 在英文上的幾種說法. 除了「P if and only if Q」外, 還有

- 「P iff Q」
- 「P is equivalent to Q」
- 「P is necessary and sufficient for Q」

1.2. Logical Equivalence and Tautology

前面我們介紹過 logical equivalence 的概念. 我們可以利用 logical equivalence 的一些規則推導出更多的 logical equivalences. 這樣的好處是不必每次都用 Truth table 來探討有關 logical equivalence 的問題.

首先我們再釐清一個說法. 當 P,Q 是確定的 statements 時, $P\wedge Q$ 和 $Q\wedge P$ 也會是確定的 statements (也就是說它們對錯的情況已經固定), 所以此時說 $P\wedge Q$ 和 $Q\wedge P$ 是 logically equivalent 並不是很恰當. 事實上我們是將 P,Q 看成變數一樣, 它們可以用任意的 statement 取代, 所以此時 $P\wedge Q$ 的對錯會因為 P,Q 的不同而有所不同, 故此時說 $P\wedge Q$ 是 statement 也不恰當. 為了方便起見, 這裡 (指的是本講義) 當 P,Q 是可變動的情況之下, 我們便稱它們利用 connectives 連結起來的結果為 "statement form", 例如我們會說 $P\wedge Q$ 和 $Q\wedge P$ 這兩個 statement forms 為 logically equivalent. 另外我們用 "$\sim$" 來表示兩個 statement forms 為 logically equivalent, 例如我們有 $(P\wedge Q)\sim(Q\wedge P)$.

第一個常見的 logical equivalence 的使用規則是: 我們可以將 logically equivalent 的兩個 statement forms 其中同一個變數用其他的 statement form 取代, 仍可得到 logical equivalence. 例如已知 $(P\wedge Q)\sim(Q\wedge P)$, 我們可將 P 用 $P\Rightarrow Q$ 取代得

$$((P\Rightarrow Q)\wedge Q)\sim(Q\wedge(P\Rightarrow Q)).$$

這個規則的原因很簡單, 因為既然 logically equivalent 的 statement forms 有相同的 truth table, 我們將其中某個變數任意變換當然最後所得新的 statement forms 仍會有相同的 truth table. 同樣的道理, 我們可以將其中某個變數用兩個 (或好幾個) logically equivalent 的 statement forms 取代, 最後所得新的 statement forms 仍為 logically equivalent. 例如已知 $(P\wedge Q)\sim(Q\wedge P)$ 以及 $(R\vee S)\sim(S\vee R)$, 所以可以將 $(P\wedge Q)\sim(Q\wedge P)$ 左邊的 P 用 $R\vee S$ 取代, 而右邊的 P 用 $S\vee R$ 取代得

$$((R\vee S)\wedge Q)\sim(Q\wedge(S\vee R)).$$

還有一個常用的規則是, 如果兩個 statement forms A,B 是 logically equivalent 而 B 和另一個 statement form C 也是 logically equivalent, 那麼 A 和 C 也是 logically equivalent. 例如我們有 $((P\wedge Q)\vee R)\sim((Q\wedge P)\vee R)$, 也有 $((Q\wedge P)\vee R)\sim(R\vee(Q\wedge P))$, 故可得

$$((P\wedge Q)\vee R)\sim(R\vee(Q\wedge P)).$$

這個規則會成立的原因仍然由 truth table 的全等可以得到.

利用這些規則我們可以不必藉由 truth table 很容易推得一些 statement forms 為 logically equivalent. 簡單來說我們可以將 logically equivalent 如"等號"一樣運用. 我們前

面學過的 logical equivalences, 例如 $\wedge$ 的交換性和 $\vee$ 的交換性, 即

$$(P\wedge Q)\sim(Q\wedge P),\quad (P\vee Q)\sim(Q\vee P) \tag{1.1}$$

以及 $\wedge$ 的結合性和 $\vee$ 的結合性, 即

$$((P\wedge Q)\wedge R)\sim(P\wedge(Q\wedge R)),\quad ((P\vee Q)\vee R)\sim(P\vee(Q\vee R)) \tag{1.2}$$

還有 $\wedge,\vee$ 之間的分配性質, 即

$$((P\wedge Q)\vee R)\sim((P\vee R)\wedge(Q\vee R)),\quad ((P\vee Q)\wedge R)\sim((P\wedge R)\vee(Q\wedge R)) \tag{1.3}$$

都是常用來幫助我們推導許多 logical equivalences 的工具.

Example 1.2.1. 考慮 $(P\wedge Q)\vee(P\vee Q)$ 這一個 statement form. 利用式子 (1.3) 中的 $((P\wedge Q)\vee R)\sim((P\vee R)\wedge(Q\vee R))$, 將 R 用 $P\vee Q$ 取代, 我們有

$$(P\wedge Q)\vee(P\vee Q)\sim((P\vee(P\vee Q))\wedge(Q\vee(P\vee Q))). \tag{1.4}$$

再由 $(P\vee(P\vee Q))\sim((P\vee P)\vee Q)$ 以及 $(Q\vee(P\vee Q))\sim(Q\vee(Q\vee P))\sim((Q\vee Q)\vee P)$ 得

$$((P\vee(P\vee Q))\wedge(Q\vee(P\vee Q))\sim(((P\vee P)\vee Q)\wedge((Q\vee Q)\vee P)). \tag{1.5}$$

很容易檢查 $(P\vee P)\sim P$ 以及 $(P\wedge P)\sim P$, 故知

$$(((P\vee P)\vee Q)\wedge((Q\vee Q)\vee P))\sim((P\vee Q)\wedge(Q\vee P))\sim(P\vee Q). \tag{1.6}$$

最後連結式子 (1.4), (1.5), (1.6), 得

$$((P\wedge Q)\vee(P\vee Q))\sim(P\vee Q).$$

當一個 statement form 其 truth table 在任何情況之下皆為對, 我們稱此 statement form 為 *tautology*. 意即它是重複多餘的. 例如 $P\Leftrightarrow P$ 的 truth table 為

P	$P\Leftrightarrow P$
T	T
F	T

,

故 $P\Leftrightarrow P$ 為 tautology.

Question 1.8. *$P\Rightarrow P$ 是否為 tautology? $P\Rightarrow(P\Rightarrow P)$ 是否為 tautology?*

Tautology 雖然有重複多餘的意思, 但它在邏輯上仍是有意思的. 它可以幫我們用另一種方法來詮釋 logically equivalent. 當兩個 statement forms A,B 為 logically equivalent 時, 因為 A,B 的對錯情況一致, 我們有 $A\Leftrightarrow B$ 恆為對. 意即 $A\Leftrightarrow B$ 為 tautology. 反之, 當 $A\Leftrightarrow B$ 為 tautology 時, 由於 A,B 的對錯情形一致, 它們有相同的 truth table. 意即 $A\sim B$. 我們有以下的性質.

Proposition 1.2.2. *假設 A,B 為兩個 statement forms. 則 A 和 B 為 logically equivalent 等同於 $A\Leftrightarrow B$ 為 tautology.*

其實在前面的說明中, 我們先假設 $A \sim B$ 成立推得 $A \Leftrightarrow B$ 為 tautology (即若 $A \sim B$ 則 $A \Leftrightarrow B$ 為 tautology), 後又由 $A \Leftrightarrow B$ 為 tautology 推得 $A \sim B$. 故 Proposition 1.2.2 可以說成 $A \sim B$ 若且唯若 $A \Leftrightarrow B$ 為 tautology.

Question 1.9. 假設 A,B 為兩個 *statement forms.* 若 $A \sim B$ 可否推得 $A \Rightarrow B$ 為 *tautology?* 若 $A \Rightarrow B$ 為 *tautology* 可否推得 $A \sim B$*?*

Question 1.10. 假設 A,B,C 為 *statement forms.* 若 $A \Leftrightarrow B$ 和 $B \Leftrightarrow C$ 皆為 *tautology,* 是否可推得 $A \Leftrightarrow C$ 為 *tautology?*

和 tautology 相反的是所謂的 *contradiction* (矛盾). 它指的是一個 statement form 在任何情況之下皆為錯的. 關於 contradiction, 我們會在下一節介紹 “not” 之後再探討.

Question 1.11. 假設 A,B 為 *statement forms.*

(1) 若 A 為 *tautology,* 試說明 $(A \wedge B) \sim B$ 且 $A \vee B$ 為 *tautology.*

(2) 若 A 為 *contradiction,* 試說明 $(A \vee B) \sim B$ 且 $A \wedge B$ 為 *contradiction.*

1.3. Not and Contradiction

我們介紹 “not” 以及和 not 有關的 equivalences. 本節內容分量比前面幾節重, 而且許多情形很可能和你的直覺不同. 希望大家能好好熟習, 糾正錯誤的直覺, 而將正確觀念成為你的本能反應而不是盲目地記誦.

Not 有否定和相反的意思, 給定一個 statement P, 我們用 $\neg P$, 來表示 not P, 一般稱為 “非 P”. 它的定義就是當 P 為對時, $\neg P$ 就為錯. 反之, 當 P 為錯時, $\neg P$ 就為對. 所以我們有以下 $\neg P$ 的 truth table.

P	$\neg P$
T	F
F	T

.

利用這個定義, 我們馬上有

$$P \sim \neg(\neg P). \tag{1.7}$$

Not P 雖然定義簡單, 但是對於由許多 connectives 連結的 statement 取 not 之後, 其對錯狀況就較複雜了. 例如 $\neg(P \wedge Q)$, 或許很多人會誤以為是 $(\neg P) \wedge (\neg Q)$, 不過檢查一下 truth table 可得

P	Q	$P \wedge Q$	$\neg(P \wedge Q)$
T	T	T	F
T	F	F	T
F	T	F	T
F	F	F	T

P	Q	$\neg P$	$\neg Q$	$(\neg P) \wedge (\neg Q)$
T	T	F	F	F
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

很明顯看出, 在 P 對 Q 錯或 P 錯 Q 對時, $\neg(P \wedge Q)$ 和 $(\neg P) \wedge (\neg Q)$ 是不同的. 事實上, 利用 truth table, 我們可得

$$\neg(P \wedge Q) \sim (\neg P) \vee (\neg Q). \tag{1.8}$$

我們藉由大家熟知的數學例子來理解這個事實. 考慮 $0 \leq x \leq 1$, 這表示 $x \leq 1$ and $x \geq 0$. 它的相反, 大家都知是 $x > 1$ or $x < 0$. 我們可以任取一個數 x 令 P 為 $x \leq 1$ 這一個 statement, 而 Q 為 $x \geq 0$, 則 $\neg P$, $\neg Q$ 分別為 $x > 1$, $x < 0$. 也就是說 $0 \leq x \leq 1$ 可以用 $P \wedge Q$ 表示而 $x > 1$ or $x < 0$ 就是 $(\neg P) \vee (\neg Q)$. 由此可以看出 $\neg(P \wedge Q)$ 和 $(\neg P) \vee (\neg Q)$ 為 logically equivalent, 而不是 $(\neg P) \wedge (\neg Q)$ (否則會得到 $x > 1$ and $x < 0$ 這個矛盾).

我們可以用上一節有關於 statement form 的 logically equivalent 的規則來處理 not. 例如將式子 (1.8) 中的 P, Q 分別用 $\neg P$ 和 $\neg Q$ 取代, 可得

$$\neg((\neg P) \wedge (\neg Q)) \sim (\neg(\neg P)) \vee (\neg(\neg Q)).$$

再利用 $\neg(\neg P) \sim P$, 得

$$\neg((\neg P) \wedge (\neg Q)) \sim (P \vee Q).$$

最後兩邊取 not, 得

$$\neg(P \vee Q) \sim (\neg P) \wedge (\neg Q). \tag{1.9}$$

例如考慮 $x \geq 0$ 的情形, 我們知它的相反為 $x < 0$. 若令 P, Q 分別為 $x > 0$, $x = 0$, 則 $x \geq 0$ 即為 $P \vee Q$. 此時 $\neg P$ 為 $x \leq 0$, $\neg Q$ 為 $x \neq 0$. 而 $(\neg P) \wedge (\neg Q)$ 為 $x \leq 0$ and $x \neq 0$, 即為 $x < 0$ 也就是 $x \geq 0$ 的相反.

式子 (1.7), (1.8), (1.9) 對於推導和 not 有關的 statement forms 之間的 logical equivalence 相當重要. 其中式子 (1.8), (1.9) 稱為 *DeMorgan's laws*.

接下來我們自然會問, 怎樣的 statement form 會和 $\neg(P \Rightarrow Q)$ logically equivalent 呢? 或許大家會認為是 $P \Rightarrow \neg Q$, 不過利用 truth table 檢查一下, 大家會發現在 P 是對的時 $P \Rightarrow Q$ 和 $P \Rightarrow \neg Q$ 確實對錯相反, 但是當 P 為錯時它們兩個皆為對. 所以 $\neg(P \Rightarrow Q)$ 和 $P \Rightarrow \neg Q$ 並不是 logically equivalent, 千萬要記住.

Question 1.12. *試寫下會使得 $x \geq 0 \Rightarrow x \geq 1$ 為對的所有實數 x, 也寫下會使得 $x \geq 0 \Rightarrow x < 1$ 為對的所有實數 x. 它們是否相反呢?*

大家常忽略的就是 $P \Rightarrow Q$ 中 P 錯的情況, 而造成邏輯的錯誤, 千萬要注意. 不過另一方面, 若 A, B 為 statement form 且 A 為 tautology, 那麼 $\neg(A \Rightarrow B)$ 就和 $A \Rightarrow \neg B$ 為 logically equivalent. 主要的原因是, A 既然全為對, 那麼 $A \Rightarrow B$ 的對錯完全會和 B 的對錯完全一致了.

Question 1.13. *試寫下會使得 $x^2 \geq 0 \Rightarrow x > 0$ 為對的所有實數 x, 也寫下會使得 $x^2 \geq 0 \Rightarrow x \leq 0$ 為對的所有實數 x. 它們是否相反呢?*

要處理 $\neg(P \Rightarrow Q)$ 會和什麼為 logically equivalent, 我們可以換一個角度來看 $P \Rightarrow Q$. 首先回顧一下 $P \Rightarrow Q$ 較通俗的說法是 P 對則 Q 一定對. 所以我們知道 Q 會對, 除非 P 是錯的. 也就是說要不然是 Q 對, 要不然就是 P 錯. 這讓我們想到 $Q \vee \neg P$ 這一個 statement form. 事實上用 truth table 檢驗

P	Q	$\neg P$	$Q \vee \neg P$
T	T	F	T
T	F	F	F
F	T	T	T
F	F	T	T

我們得到
$$(P \Rightarrow Q) \sim (Q \vee \neg P). \tag{1.10}$$
利用 $(Q \vee \neg P) \sim ((\neg P) \vee Q)$ 以及 $\neg(\neg Q) \sim Q$, 我們得 $(P \Rightarrow Q) \sim ((\neg P) \vee \neg(\neg Q))$, 再利用式子 (1.10) 得 $((\neg P) \vee \neg(\neg Q)) \sim ((\neg Q) \Rightarrow (\neg P))$, 故知
$$(P \Rightarrow Q) \sim ((\neg Q) \Rightarrow (\neg P)). \tag{1.11}$$
這和我們提過 $P \Rightarrow Q$ 為對, 表示若 Q 為錯則 P 一定錯, 相吻合.

利用式子 (1.10), 我們可得 $\neg(P \Rightarrow Q) \sim \neg(Q \vee \neg P)$. 而由 DeMorgan's laws 知
$$\neg(Q \vee \neg P) \sim ((\neg Q) \wedge \neg(\neg P))$$
故得
$$\neg(P \Rightarrow Q) \sim (P \wedge (\neg Q)). \tag{1.12}$$
式子 (1.10), (1.11), (1.12) 是我們將來處理“若 P 則 Q” 這種類型的論述時常用的 logical equivalences.

由式子 (1.10) 我們知道, 所有的 statement form 都可以利用 logical equivalence 寫成 $\neg, \wedge, \vee$ 的組合. 例如由 $P \Leftrightarrow Q$ 的定義, 我們可得
$$(P \Leftrightarrow Q) \sim (Q \vee (\neg P)) \wedge (P \vee (\neg Q)). \tag{1.13}$$
再利用 $\wedge, \vee$ 的分配性 (即式子 (1.3)) 推得
$$(P \Leftrightarrow Q) \sim (P \wedge Q) \vee ((\neg P) \wedge (\neg Q)). \tag{1.14}$$
因此我們可以用 DeMorgan's laws, 式子 (1.7), 以及 $\wedge, \vee$ 之間的關係式 (式子 (1.1),(1.2), (1.3)), 推導出一個 statement form 取 not 之後的 logical equivalence. 例如式子 (1.13) 取 not 可得
$$\neg(P \Leftrightarrow Q) \sim ((\neg Q) \wedge P) \vee ((\neg P) \wedge Q).$$
有趣的是, 若比較式子 (1.14) 中的 Q 用 $\neg Q$ 取代後的結果, 我們得到
$$\neg(P \Leftrightarrow Q) \sim (P \Leftrightarrow \neg Q).$$

當 A 為 statement form 時, $\neg A$ 的對錯完全和 A 的對錯相反, 所以 $A \Leftrightarrow \neg A$ 的 truth table 在任何情況之下皆為錯, 可知 $A \Leftrightarrow \neg A$ 為 contradiction. 反之, 若 B 為 statement form 且 $A \Leftrightarrow B$ 為 contradiction, 表示在任何情況下 A 和 B 的對錯情況相反, 可知 $B \sim \neg A$. 因此我們有以下和 Proposition 1.2.2 相對應的性質.

Proposition 1.3.1. 假設 A, B 為兩個 *statement forms.* 則 $\neg A$ 和 B 為 *logically equivalent* 等同於 $A \Leftrightarrow B$ 為 *contradiction.*

1.4. Quantifiers

我們已經了解在已知各 statement 的對錯情況之下它們用 connective 以及 not 連接之後其對錯的狀況, 我們也知道一個 statement form 的否定為何. 不過一個單一的 statement, 很可能就很複雜, 不容易判斷對錯. 例如在數學上一個 statement 常常會有一些 quantifier (量詞) 出現, 而增加了判斷對錯的困難度. 在本節中我們將介紹常見的 quantifiers, 並探討它們取否定的情形.

數學上常見的 quantifiers 有以下幾種:

- “for all”, “for every” (即對所有的), 常用 $\forall$ 表示.
- “there exists”, “there is” (即存在, 可以找到), 常用 $\exists$ 表示.
- “there is a unique” (即存在唯一的), 常用 $\exists!$ 表示.

$\exists!$ 牽涉到唯一性的問題, 以後我們在談論證明方法時會提到它, 這裡我們先探討 $\forall$ 和 $\exists$. 首先要說明的是, 在談論這些 quantifiers 時必須說明清楚是在怎樣的集合內. 比方說對所有的整數和對所有的有理數就是完全不同的兩回事, 而存在一個自然數和存在一個偶數也不同. 不過由於我們僅介紹這些 quantifiers 的概念, 而不觸及證明. 所以這裡為了簡單起見我們說明的例子考慮的都是整個實數. 例如我們說 $\forall x$ 或 $\exists x$, 它們分別表示的就是 for all x in $\mathbb{R}$ 或 there exists an x in $\mathbb{R}$, 以後就不再聲明指的是實數了.

我們先看簡單的例子: $\forall x,\ x^2 \geq 0$. 指的就是所有的實數 x 皆會滿足 $x^2 \geq 0$. 我們知道這個 statement 是對的, 因為每一個實數 x 都對, 沒有例外. 這類的 statement 我們可以用以下的形式表示 “$\forall x,\ P(x)$”. 這裡 $P(x)$ 指的是和 x 有關的條件 (例如上例中 $P(x)$ 就是 $x^2 \geq 0$). 它指的就是所有的 x 皆會滿足 $P(x)$ 這個條件. 這個 statement 要對就必須所有的 x 都對, 一個都不能錯. 例如 $\forall x,\ x^2 > 0$ 便是錯的 ($x = 0$ 就不成立).

類似的, 我們可以用 “$\exists x,\ P(x)$” 來表示, 存在 x 使得 $P(x)$ 成立. 這個 statement 要對, 只要能找到一個 x 使得 $P(x)$ 成立即可. 注意它並沒有說有多少個會對, 有可能很多, 有可能只有一個, 所以只要找到一個對即可 (這就是英文用 there exists 的原因). 上面提過 $\forall x,\ x^2 > 0$ 是錯的, 但若改為 $\exists x,\ x^2 > 0$ 便是對的 (取 $x = 1$, 即可).

$\forall$ 和 $\exists$ 有著有趣的關係, 例如 “$\forall x,\ P(x)$” 是對的話, 那麼 “$\exists x,\ P(x)$” 就一定對 (只要挑隨便一個 x 即可). 不過反過來就不對. 你不能隨便挑幾個 x 符合 $P(x)$, 就聲稱對所有的 x 都會符合 $P(x)$. 另外 $\forall$ 和 $\exists$ 在取否定時關係就更密切了. 當你發現 “$\forall x,\ P(x)$” 有可能錯時, 如何說明它是錯的呢? 前面說過 “$\forall x,\ P(x)$” 只要有一個 x 不符合 $P(x)$ 就是錯的, 所以要否定它, 我們只要找到一個 x 讓 $P(x)$ 不成立即可. 用符號表示就是 $\exists x,\ \neg P(x)$. 例如前面提過 $\forall x,\ x^2 > 0$ 是錯的, 因為我們發現 $\exists x,\ x^2 \leq 0$.

注意很多同學會誤以為 “$\forall x,\ P(x)$” 的否定是 “$\forall x,\ \neg P(x)$”. 雖然若 “$\forall x,\ \neg P(x)$” 是對的可以知道 “$\forall x,\ P(x)$” 是錯的. 但是 “$\forall x,\ P(x)$” 是錯的, 並不表示 “$\forall x,\ \neg P(x)$” 是對的. 所以不能說 “$\forall x,\ P(x)$” 的否定是 “$\forall x,\ \neg P(x)$”. 例如 $\forall x,\ x^2 > 0$ 是錯的, 但 $\forall x,\ x^2 \leq 0$ 也是錯的, 唯

有 $\exists x,\, x^2 \le 0$ 才會對. 大家千萬注意, 不要弄錯. 總而言之我們有以下的 logical equivalence

$$\neg(\forall x, P(x)) \sim (\exists x, \neg P(x)). \tag{1.15}$$

同理要否定 “$\exists x,\, P(x)$”, 表示找不到 x 使得 $P(x)$ 成立. 所以我們便需說明所有的 x 皆不滿足 $P(x)$, 也就是說 $\forall x,\, \neg P(x)$. 同樣的, 很多同學會誤以為 “$\exists x,\, P(x)$” 的否定是 “$\exists x,\, \neg P(x)$”. 這是錯的, 因為找到 x 不滿足 $P(x)$ 還是有可能找到另一個 x 會滿足 $P(x)$. 因此光由 “$\exists x,\, \neg P(x)$” 並不能否定 “$\exists x,\, P(x)$”. 總而言之我們有以下的 logical equivalence

$$\neg(\exists x, P(x)) \sim (\forall x, \neg P(x)). \tag{1.16}$$

Question 1.14. 試利用式子 *(1.15)* 以及 *logical equivalence* 的規則推導出式子 *(1.16)*.

Quantifier 有時會發生在兩個或更多變數的情形, 這裡我們僅探討兩個變數的情形, 更多變數的情況可以依兩個變數的情況類推下去. 所謂兩個變數的情況, 是形如 “$\forall x, \exists y, P(x,y)$” 的 statement, 這裡 $P(x,y)$ 指的是和 x,y 有關的條件. 例如微積分中, 函數 $f(x)$ 在 $x = a$ 的極限為 l (即 $\lim_{x\to a} f(x) = l$) 的定義 “$\forall \varepsilon > 0, \exists \delta > 0, 0 < |x-a| < \delta \Rightarrow |f(x)-l| < \varepsilon$” 就是兩個變數的情況. 大致上我們會有下面四種類型的 statement.

$$(1)\forall x, \exists y, P(x,y) \quad (2)\exists x, \forall y, P(x,y) \quad (3)\forall x, \forall y, P(x,y) \quad (4)\exists x, \exists y, P(x,y).$$

(1) 指的是: 對於所有的 x 皆可找到 y 使得 $P(x,y)$ 成立. 注意這裡 x 的部分先講, 再提存在 y, 所以這個存在的 y 並不是固定的, 它可能會隨著 x 的選取而變動. 例如 $\forall x, \exists y, x+y=0$ 這個 statement 是對的. 它說任意選取 x, 皆可找到 y 滿足 $x+y=0$. 這裡 y 會隨著 x 而變動, 即 $y=-x$. 例如 $x=1$ 時 $y=-1$, 而 $x=2$ 時 $y=-2$. 這裡 x,y 的先後順序很重要, 千萬要注意.

(2) 指的是: 存在 x 使得對所有的 y 都會滿足 $P(x,y)$. 注意這裡存在的 x 先講, 再提所有的 y, 所以這個存在的 x 並是固定的, 它不可以隨著 y 而變動. 例如 $\exists x, \forall y, x+y=y$ 這個 statement 是對的. 它是說可以找到 x 讓任意的 y 皆滿足 $x+y=y$. 這裡 x 找到後便固定下來了, 即 $x=0$. 不過例如在 (1) 的情形我們知道 $\forall x, \exists y, x+y=0$ 這個 statement 是對的, 但若將 $\forall x$ 和 $\exists y$ 的順序交換得 $\exists y, \forall x, x+y=0$ 這個 statement 便是錯的. 因為我們無法找到一個固定的 y 使的所有的 x 都會滿足 $x+y=0$. 再次強調, 這裡先後順序很重要, “$\forall x, \exists y, P(x,y)$” 和 “$\exists y, \forall x, P(x,y)$” 雖然只是 $\forall x$ 和 $\exists y$ 先後順序調動, 但意義完全不同千萬要注意.

Question 1.15. $\exists x, \forall y, x+y=y$ 這個 *statement* 是對的, 但若換成 $\forall y, \exists x, x+y=y$, 是否為對呢? 又換成 $\forall x, \exists y, x+y=y$ 及 $\exists y, \forall x, x+y=y$, 哪一個對呢?

Question 1.16. 假設 $f(x,y), g(x,y)$ 皆為兩個變數的多項式. 已知 “$\forall x, \exists y, f(x,y)=0$” 和 “$\exists y, \forall x, g(x,y)=0$” 皆為對. 試問 $f(x,y)=0$ 和 $g(x,y)=0$ 在坐標平面上的圖形哪一個一定會包含一條水平直線, 哪一個一定會和鉛直線 $x=101$ 相交?

(3) 和 (4) 的情況較為單純. (3) 指的是任取一個 x, 對於任意的 y 都會使得 $P(x,y)$ 成立. 利用坐標平面的看法, 我們可以說平面上任一點 (x,y) 都會使得 $P(x,y)$ 成立, 所以此時

$\forall x$ 和 $\forall y$ 變換順序並不會改變整個 statement. 而 (4) 指的是可以找到 x 使得有一個 y 滿足 $P(x,y)$. 利用坐標平面的看法, 我們可以說平面上存在一點 (x,y) 使得 $P(x,y)$ 成立. 因此此時 $\exists x$ 和 $\exists y$ 變換順序並不會改變整個 statement. 例如若我們在 $x=3$ 時, 可找到 $y=7$ 使得 $P(3,7)$ 是正確的, 此時我們也可以說 $y=7$ 時, 可找到 $x=3$ 使得 $P(x,y)$ 為對. 總而言之 (3), (4) 因兩個變數的 quantifier 皆相同, 所以 x,y 的先後不重要. (3) 一般會簡化成 $\forall x,y$, $P(x,y)$, 而 (4) 簡化成 $\exists x,y$, $P(x,y)$.

接下來我們來看有兩個變數的 statement 取否定時 quantifier 的變化情形. 在 (1) 的情形, 即 "$\forall x,\exists y,P(x,y)$". 此時, 我們可以把 "$\exists y,P(x,y)$" 看成是 $H(x)$ 這樣的條件. 所以原 statement 可看成 $\forall x,H(x)$. 利用式子 (1.15), 我們知道它的否定為 $\exists x,\neg H(x)$. 然而式子 (1.16) 告訴我們 $\neg H(x)\sim(\forall y,\neg P(x,y))$, 所以我們得

$$\neg(\forall x,\exists y,P(x,y)) \quad\sim\quad (\exists x,\forall y,\neg P(x,y)).$$

同理我們可得

$$\begin{aligned}\neg(\exists x,\forall y,P(x,y)) &\quad\sim\quad (\forall x,\exists y,\neg P(x,y))\\ \neg(\forall x,\forall y,P(x,y)) &\quad\sim\quad (\exists x,\exists y,\neg P(x,y))\\ \neg(\exists x,\exists y,P(x,y)) &\quad\sim\quad (\forall x,\forall y,\neg P(x,y)).\end{aligned}$$

例如前面所提, 函數 $f(x)$ 滿足 $\lim_{x\to a}f(x)=l$ 的否定應為

$$\exists\varepsilon>0,\forall\delta>0,\neg(0<|x-a|<\delta\Rightarrow|f(x)-l|<\varepsilon).$$

利用式子 (1.12) 我們知

$$\neg(0<|x-a|<\delta\Rightarrow|f(x)-l|<\varepsilon) \quad\sim\quad ((0<|x-a|<\delta)\wedge(|f(x)-l|\geq\varepsilon)).$$

所以 $\lim_{x\to a}f(x)=l$ 的否定應為

$$\exists\varepsilon>0,\forall\delta>0,(0<|x-a|<\delta)\wedge(|f(x)-l|\geq\varepsilon).$$

最後, 我們說明一下 $\forall$ 和 $\exists$ 在習慣上用法的差異. 在習慣上的用語, 我們常會省略 $\forall x$. 例如 $x\geq3\Rightarrow x^2\geq9$, 這一個 statement 嚴格來說應寫成 $\forall x,x\geq3\Rightarrow x^2\geq9$. 也就是說, 在邏輯上我們說這個 statement 是對的應該是對所有的實數 x 都是對的. 給定一實數 x, 當 $x\geq3$, 當然可得 $x^2\geq9$. 而當 $x<3$, 因為它已不符合 $x\geq3$ 的前提, 我們知道此時 $x\geq3\Rightarrow x^2\geq9$ 也是對的. 所以我們可以認定 $\forall x,x\geq3\Rightarrow x^2\geq9$ 是對的 (這也是邏輯上定義 P 錯時 $P\Rightarrow Q$ 為對的用意, 希望同學能體會). 要注意的是 $\exists x$ 就絕不能省略, 否則就弄不清楚是 $\forall x$ 或 $\exists x$ 了. 總而言之, 當我們碰到「所有 x 只要符合 $P(x)$ 也會符合 $Q(x)$」這種 statement 時, 原本應寫成「$\forall x,P(x)\Rightarrow Q(x)$」, 我們常省略 $\forall x$ 而用「$P(x)\Rightarrow Q(x)$」來表達. 而如果是「存在 x 符合 $P(x)$ 也符合 $Q(x)$」, 我們就不能省略 $\exists x$. 不過要注意, 即使 $\exists x$ 沒省略, 也不可以寫成「$\exists x,P(x)\Rightarrow Q(x)$」, 而是用「$\exists x,P(x)\wedge Q(x)$」來表達. 這是因為,「$\exists x,P(x)\Rightarrow Q(x)$」表示只要找到 x 使得 $P(x)\Rightarrow Q(x)$ 為對就可, 但所有不滿足 $P(x)$ 的 x 都會使得 $P(x)\Rightarrow Q(x)$ 為對. 這會造成即使沒有 x 滿足 $P(x)$, 也會使得「$\exists x,P(x)\Rightarrow Q(x)$」為對, 這就和原來的「存在 x 符合 $P(x)$ 也符合 $Q(x)$」意義相左了. 例如「存在一個大於 3 的實數 x, 滿足 $x^2=10$」就應

寫成「$\exists x,(x>3)\wedge(x^2=10)$」(此時 $x=\sqrt{10}$), 而不是「$\exists x,(x>3)\Rightarrow(x^2=10)$」(此時 $x=2$ 也會對).

Question 1.17. 假設 $f(x,y)$ 是一個兩個變數的多項式.「存在一實數 $a>0$ 使得 $f(a,y)=0$ 無解」這一個 *statement,* 數學的表示法為何? 並寫出這 *statement* 的否定.