

查詢介面的布林邏輯批判性思考：如何培育布林邏輯的資訊素養技能  
Critical Thinking in the Boolean Logic of User Interface: How to Cultivate the Skills for  
Information Literacy

李素蘭  
靜宜大學通識中心人文科

摘要

一般人查詢資料庫時，因資訊需求的關係，須靠著布林邏輯才能勾畫出他複雜主題的需求。四十年來布林邏輯是資料庫查詢系統的主流。可是，許多文獻指出布林邏輯觀念仍是大部份人查詢資料庫的障礙主因。當前，自然語言未成氣候。在視窗環境下許多資料庫的查詢介面由單行指令轉變成填充式及下拉式的選項畫面。各家系統廠商標榜友善的查詢介面 (User Friendly Interface)，卻有著不一樣操作方式，事實上，已造成查詢者的學習負擔。在每家查詢介面的背後，其布林邏輯的運算觀念實際上是相同的。真正了解的查詢者並不多。本文嘗試在每家不同的查詢介面中找出一個通則，以協助查詢者及教育者(館員)對布林邏輯建立批判性的思考模式。

本文就三方面來探討：從查詢者資訊需求的如何建構布林邏輯關係；不同系統中不同的布林邏輯的運算優先順序造成查詢結果，比較其差異性。同時，就查詢者的角度，針對當前的系統提供一些建言：如系統在主題上的自動篩選以避開同詞異義詞的困擾、文獻附加歸類分級等。本文試做系統與使用者之間的詮釋橋樑，協助查詢者培育資訊素養的技能。

關鍵詞：布林邏輯、系統介面、資訊素養技能

ABSTRACT :

Due to the complicity of information needs, boolean logic, the most dominant and popular method in information retrieval, is the best way to achieve it so far. However, variety of researches indicate that the boolean logic concept is the primary obstacle for most people in database searchings. The natural language has not been matured , the single-command driven menu has been changed into fill-up or drawn-down menu under Window environment. The miscellaneous user interfaces from various vendors, been declared as "User friendly", have become as learning burden for users. This article tries to find out the generality of Boolean logic behind all systems in order to help users and librarians build up the concept of critical thinking .

Three aspects covered are: First, search strategy by the combination of boolean logic relationship of user needs. Secondly, the comparison between different outputs based on precedence of Boolean logic of vendor systems. Also, the author gives some suggestions, from the point of users' view, such as: the automated filtering in specific subject avoiding the misconception and chaos of synonyms, the classification in documents by use of implied AND, etc..

Keyword: Boolean logic, User interface, Skills for Information literacy

目前國內大多數的大學圖書館均有引入許多商業性資料庫，來提供校內師生服務。網路及電腦的普及性造成圖書館的讀者線上查詢模式由傳統的服務型態轉變成讀者直接接觸。也就是說，館員不再是讀者與資料庫之間的仲介，館員不再在資訊與讀者之間搭作橋樑。館員變成是一個介紹者，引渡資料庫給讀者認識。更因為校內的師生之教育層度也比一般社會大眾的水準較為稱齊，所以，他們的工作角色已從資訊的代理蒐集者轉成教育者。為達成其任務，許多大學圖書館也都用各種方式來教育讀者。有的以學分課程正式授課；有的以系統為單元來做非正式讀者教育(如研習會)；也有的以主題或資料型態(像如何蒐集商業資料；或如何蒐集專利)來接觸不同需求的讀者；有的設計一些線上課程來讓讀者以自己的步調來學習操作系統等。這些種種不外乎只有一個很簡單的理念：教育讀者讓他能自己上線查詢他想要的資料，也只有他自己最清楚他對主題需求、資料層次的需求、甚至對資料類型的需求。所以，教育讀者學習如何操作系統變成參考館員最重要的職責。

現行的許多資料庫的使用者，包括館員與讀者，對於系統的理解常囿於線上所提供的系統說明。實際上，有許多系統說明缺乏邏輯性的安排。或者說，有些甚至說明很少。有的系統設計者認為使用者不太需要去了解太多。反正資料庫的查詢畫面已經夠清楚，且資料庫所包含的資料量也相當多，怎麼樣都能夠讓使用者查獲資料。館員自身也很少去深入探討系統背後所隱藏的一些功能或意義。

1847年英國數學家 George Boole 發展一套數學邏輯來處理“真”或“假”邏輯功能。在 1960 年代布林邏輯查詢系統幾乎已成一種標準。雖說有向量處理(vector-processing)及或然率處理(probabilistic retrieval model)等方法，近四十年來布林邏輯方式的查詢模式仍舊是資料庫查詢的主流。布林邏輯有下列缺點：

- (一)查詢者無法控制查詢結果數量的多寡(Cooper, 1997)；
- (二)查詢的結果無法顯示每筆資料與題意的相關度(Frants, 1999)。雖說有些系統以查詢詞的出現次數來量化相關度，然仍有待商榷。
- (三)查詢用詞在題意中所佔的重要性比重，查詢者無法在指令中表示；
- (四)布林邏輯是查詢詞與文獻間是二位進位的觀念，就是只有 0 與 1 的選擇。譬如指令 A AND B AND C AND D，若一筆資料中有 A,B,C 但沒有 D，就沒有被查獲。查詢者就不知道在該資料庫尚有該筆資料也頗接近他的需求。又譬如指令 A OR B OR C OR D，此 A,B,C,D 四詞在查獲的每一筆資料中的出現頻率，及比率。查詢者也不會很注意，除非他詳閱該筆資料並細數之(Savoy, 1997)。

在國外文獻中不可置否的一件事實：許多查詢者不懂得如何將他的資訊需求用布林邏輯方式來化成查詢指令。像 Willet、Ensor、Mader 等人的文獻(Ensor, 1992)(Mader, 1989)(Willet, 1996) Nahl 的文中更明確指出讀者除了用詞方面的能力不足、資料庫的結構不熟外，更嚴重的是在自己主題需求的建構上有著明顯的困難。換言之，也就是布林邏輯的關係組合能力欠缺 (Nahl, 1996)。

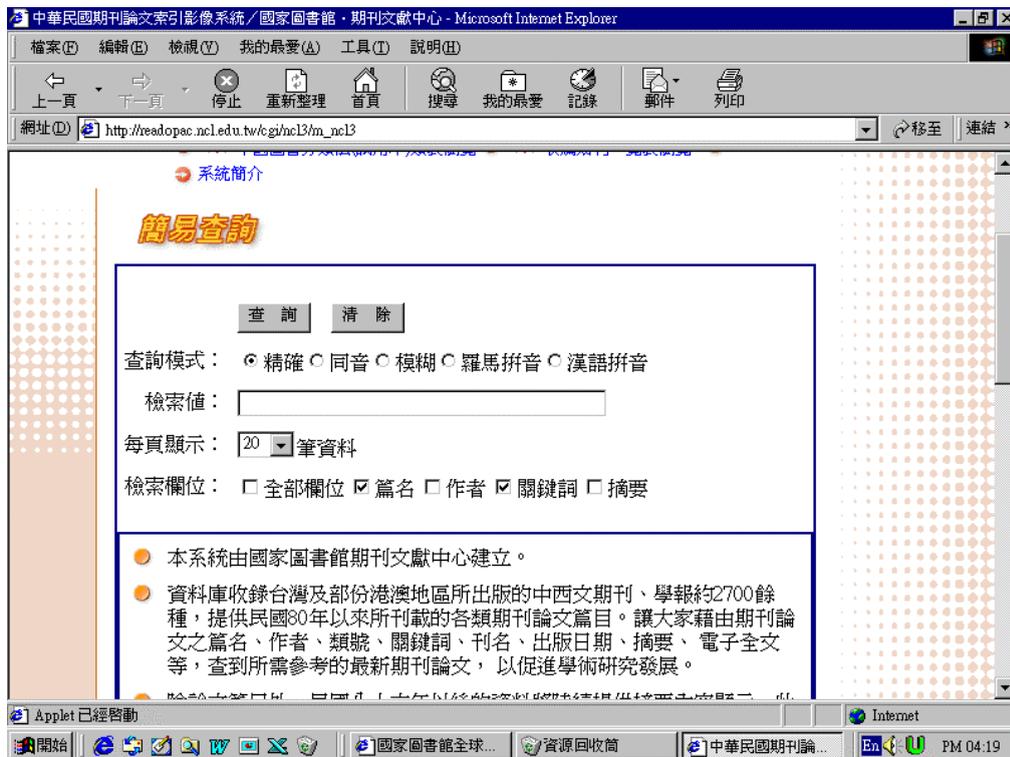
布林邏輯之產生與資料電子化有唇齒不分的關係。以傳統紙本期刊論文索引參考書為例，查詢點有限：如主題、作者、篇名。大多數只提供主題排序。大多數參考書只做到兩個主題交集的排序。囿於篇幅，極少能做三個以上主題交集做交錯的排序。不同查詢點之間的邏輯關係是很難建立。而電子化的資料，不只是查詢點增多，除主題、作者、篇名，外加刊名、資料型態、語文別、出版年等。同一(或同一被查詢)欄位可做多層次的布林邏輯運算外，更可做到不同查詢點(或被查詢欄位)之間布林邏輯的交錯組合。

再就讀者的需求方面，布林邏輯也是最能滿足讀者的方式。在讀者的需求中極少是單一主題的需求。尤其是學術層次愈高的讀者，相對地，他對文獻主題的需求就愈複雜。這種複雜的需求也只有用電腦的布林邏輯運算才可能滿足。這是電腦化的最強烈的因素。就查詢指令來看布林邏輯的組合：(一)、交集：可能是需求主題的交集，像"圖書館"與"網路"。(二)、聯集：相似或同義詞的聯集，如"aids"、"愛滋病"、"愛死病"、"後天免疫缺乏症候群"；不同拼法或寫法，如『台』字也可是『臺』字；如"labor" 或"labour"；任選一者的主題，如"Tennis"或"football"。(三)、否定：否定像是負面交集，例如要"電腦交易"，但不要與"銀行"相關(電子交易 NOT 銀行)。或者，去除相似名稱，如 squirrel NOT squirrel monkeys；"大學" NOT "大學生"。

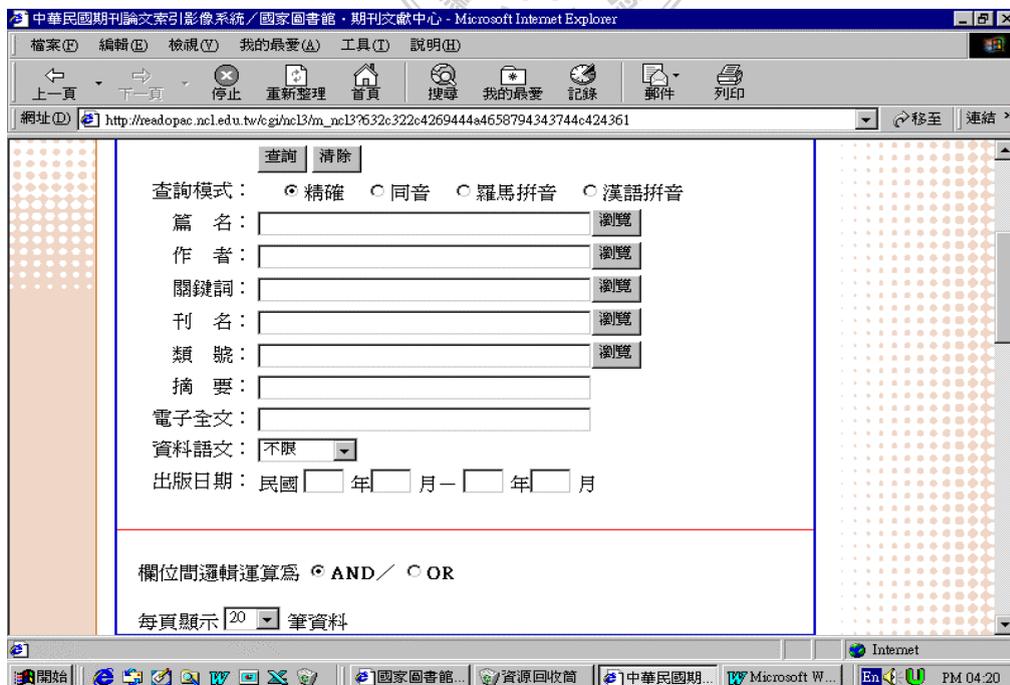
再者，又可以將讀者方面的需求藉由資料庫欄位用布林邏輯做出更複雜的交錯關係。譬如 人機互動 AND au=林司田 AND py=1999。或者，可以在暫存集中做布林邏輯關係，如 S1 AND S2。或集合與前面的主題或欄位做布林邏輯，如 S1 AND 電腦動畫。指令中的隱喻式布林邏輯關係因涉及字詞處理方式，範圍很廣，本文中並不擬探討過多。

傳統的指令式(Command Query)因為視窗環境的改變，遂使得布林邏輯查詢模式在查詢介面上產生多樣化的型態。其目的也是為讓查詢介面更友善(User Friendly)。此外，查詢者可依個人所好，可以有更多查詢介面的選擇。像中華民國期刊論文索引影像系統在布林邏輯的查詢方式就提供三種方式：簡易查詢(圖一)、詳細查詢(圖二)及指令查詢(圖三)。只要是查詢條件的邏輯組合相同的話，這三種方式所得的查詢結果應該是一樣。此外，使用者在某些查詢介面雖沒有明確下布林邏輯指令 NOT, AND, OR，但實際上，系統程式暗地裏用布林邏輯方式在處理讀者所傳送的訊息。譬如在簡易查詢下"愛滋病"，同時在下面選項中勾選「篇名」及「關鍵詞」欄位時，這就相當於在指令查詢畫面下 TI=愛滋病 + KW=愛滋病"一樣。也就是說，在簡易查詢下，系統做跨兩個欄位的查詢，同時也做布林邏輯隱喻式聯集的運算。此外，在簡易查詢下"愛滋病"，同時圈選"精確"查詢模式時，系統會自動做 "愛滋病 OR 愛死病 OR aids OR 後天免疫缺乏症候群"，類似隱喻式聯集的運算。

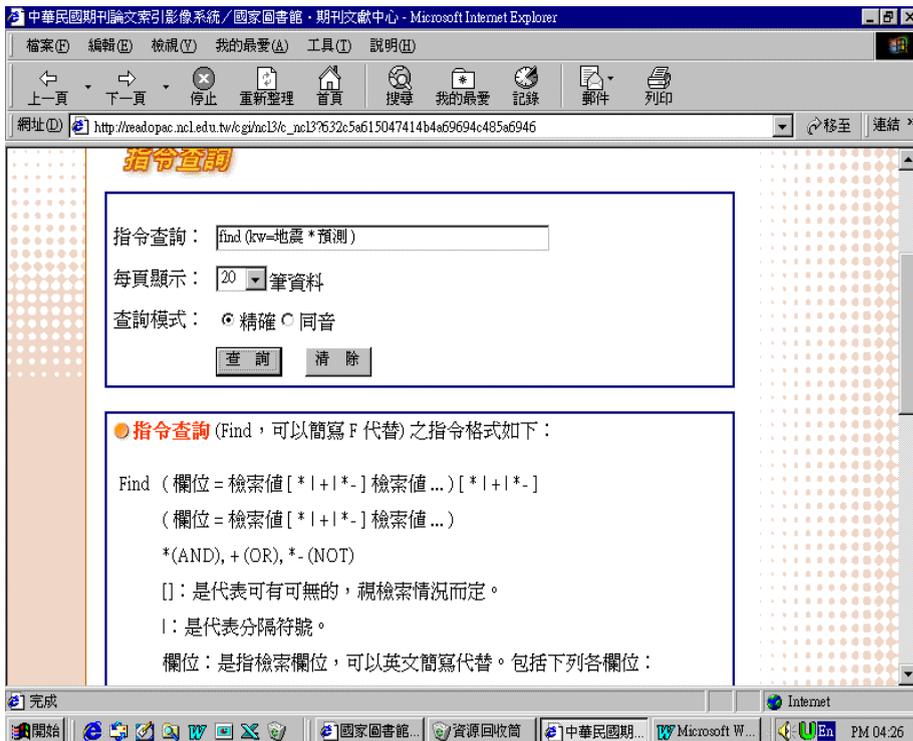
許多查詢者常不會去注意線上說明。有電腦課或寫電腦程式經驗的查詢者會用 AND, OR NOT 分別去表達交集、聯集及否定。但就一般人即使在查詢 Internet 的搜尋引擎，面對大量資料量的搜尋時，AND 是比較常被使用的布林邏輯運算符號。而 NOT 及 OR 則很少被用。在做交集 AND 時，許多人不知道最多該用幾個詞或字。尤其當一個系統沒有提供暫存記憶的集合時，最後常以零筆作收，或者出現其他的錯誤訊息(Barnett, 1999)。當然，系統的好壞也可以從布林邏輯略窺端倪。譬如從反應時間及查詢結果。查詢結果比較難偵測因為有些系統常以零筆作收或以內定的上限筆數顯示查詢結果。面對資料庫的大量資料，查詢者無法印證其正確性。筆者建議可單獨下 NOT 指令來看系統處理大量資料的能力如何。



(圖一)



(圖二)



(圖三)

布林邏輯運算符號在不同的系統中有不同之代號。這些在線上使用說明中都會明白陳述，然運算的優先順序就未必明言。(圖四)表示出台灣地區圖書館常用系統中的布林邏輯運算的優先順序及在線上操作說明中有無表述。[註：圖中所示”由左而右”及”由右而左”，二者是相同的，只是不同系統的不同說詞。]

許多不解布林邏輯的人會誤將日常生活中的用詞轉化成布林邏輯的運算符號。讀者需求的題目中若有「與」、「和」、「及」，並不相當於布林邏輯中的交集“AND”。題目中的「或」、「抑」，也不等於聯集“OR”。交集或聯集的使用完全視查詢者本身的需求及背景而定。

有的系統運算優先順序是 NOT/AND/OR；有的是 NOT/OR/AND；有的是由左而右(或由右而左)；有的卻是由上而下。(圖四)中可發現一點是國外的系統大多數會在線上說明中明確表述。而國內的系統則不。但線上說明中不表述並不表示系統沒有運算的優先順序，它還是照著它自訂的優先順序在處理查詢者所下的指令。

系統名稱	運算優先順序	線上說明優先順序
TTSWeb(單行指令)	由右而左	有
TTSWeb(下拉式畫面)	由上而下	無
中央通訊社剪報系統(單行指令)	由右而左	有
中華民國期刊論文索引影像系統 (簡易查詢)	由右而左	無
國立教育資料館教育論文資料庫	由左而右	無
國立師大教育論文線上資料庫(單行指令)	由右而左	無
國立師大教育論文線上資料庫(填充下拉式)	由上而下	無
飛資得 Flybase 系統	NOT/AND/OR	有
ACM Digital Library	NOT/AND/OR	有
Eshaman	NOT/AND/OR	無
ProQuest Direct	NOT/OR/AND	有
WEBSPIRS	由左而右(只有在查詢者忘記加入括弧時， 系統自動加入，且以由左而右方式處理)	有
CSA IDS	先 AND 再 OR(系統已供查詢者在兩字以上 做交集或連集的選擇)	有

(圖四)

甚麼是運算符號的優先順序？電腦程式語言本身就有，譬如像 Dbase III 的優先順序是 NOT, AND, OR。但是，許多系統的布林邏輯運算優先順序是系統工程師所訂定，與其所採用系統之程式語言無關。但是，在資料查詢時，布林邏輯的運算優先順序就會表現出不同的邏輯關係組合。而這不同的組合關係就會呈現出不同的查詢結果。到底運算符號的優先順序是什麼？如何讓查詢者了解它的作用差異性，筆者試以最簡單的方式來做比喻。

算術中加、減、乘、除的運算中，我們既定的運算優先順序是先乘/除、後加/減，由左而右。也只有()才能改變其優先順序。而括弧有破壞優先順序的作用。也就是說，括弧內的部份先運算。當然，在括弧裏面也是依其既訂的運算優先順序。譬如：

$$3+5*2$$

在先乘/除、後加/減的運算優先順序中， $3+5*2$  相當於  $3+(5*2)$  因為乘號有較高的優先順序；換言之，就好像在  $5*2$  有一道隱藏的括弧()，先將  $5*2$  做運算。所以，其結果就  $=3+10=13$ 。

若加/減先做，再做乘/除的話，則  $3+5*2$  就像在  $3+5*2$  中的  $3+5$  插入一個括弧改變其優先順序，則成  $(3+5)*2$ 。先做  $3+5=8$ ，再做  $8*2=16$ 。所以其結果是不相等的。將兩者之運算式列出如下：

先乘/除，後加/減	$3+5*2=3+(5*2)=3+10=13$
先加/減，後乘/除	$3+5*2=(3+5)*2=8*2=16$

同理，這種方式應用在邏輯符號的運算。在大多數系統中，這三種運算符號的運算優先順序(PRIORITY, PRECEDENCE)是 NOT, AND, 最後是 OR，像 Wilson 公司的 Wilsearch 系統。但也非每個系統均是如此，如 Bell & Howell (UMI)公司的 ProQuest Direct 系統則是 NOT 最高，再 OR，最後才 AND。有的系統則不管任何運算符號的運算優先秩序，一切運算順序是由左而右(或稱由右而左，像照鏡子)，像 AIP(American Institute of Physics)、國家圖書館的「中華民國期刊論文索引影像系統」及漢珍公司的 TTSWeb。不管以上任何一種，唯有()才能改變其優先順序，換言之，()之運算優先順序最高。當然，在括弧裏面也是依其既訂的運算優先順序。

以下所做之示範是假設在運算符號的優先順序是 NOT, AND, OR。以下 A, B, C 代表查詢之關鍵字(詞)；"=" 代表左、右之查詢結果是一致的；" " 代表左、

右查詢結果不相等； "?" 代表截字符號。

$A \text{ AND } (\text{NOT } B) = A \text{ AND NOT } B = (\text{NOT } B) \text{ AND } A = \text{NOT } B \text{ AND } A$   
 $\text{NOT } A \text{ AND } B \quad \text{NOT } (B \text{ AND } A)$

[其中若運算符號之優先順序是 NOT, OR, AND 的話，則結果也會一樣。]

$A \text{ AND } B \text{ OR } C = (A \text{ AND } B) \text{ OR } C \quad A \text{ AND } (B \text{ OR } C)$

[其中若運算符號之優先順序是 NOT, OR, AND 的話，則  $A \text{ AND } B \text{ OR } C = A \text{ AND } (B \text{ OR } C)$ 。]

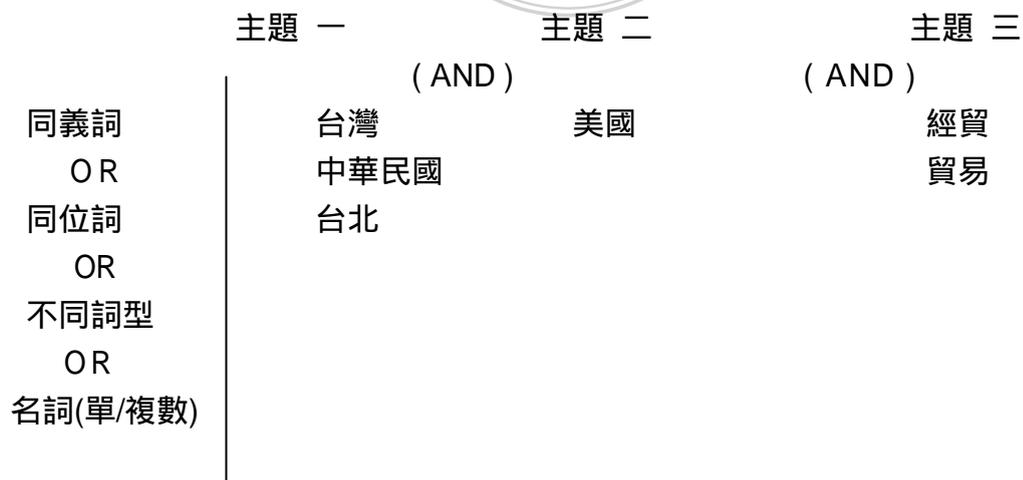
$A \text{ OR } B \text{ AND } C = A \text{ OR } (B \text{ AND } C) \quad (A \text{ OR } B) \text{ AND } C$

[其中若運算符號之優先順序是 NOT, OR, AND 的話，則  $A \text{ OR } B \text{ AND } C = (A \text{ OR } B) \text{ AND } C$ 。]

假設在運算符號的優先順序是 NOT, AND, OR，舉下列實例：

題目一：中美經貿發展之探討

將此題目化成資訊檢索策略圖，如(圖五)。



(圖五)

將(圖五)中的同一主題下先將同義詞、同位詞、不同詞型及英文名詞(單、複數)先做聯集。並用括弧將之合在一起。之後，再將不同主題之間做交集(Leverence, 1994)。這就形成了『標準型單行指令』。

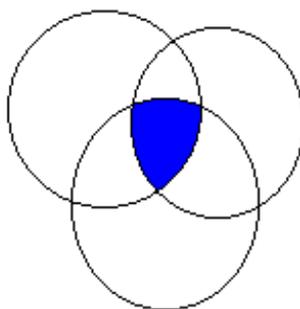
(圖五)所提及之不同詞型，在中文指的是同義不同字。如『體』及『体』。英文中不同詞類的變化，像動詞、名詞、形容詞、副詞等。或可能用直接動詞、介系詞片語、或其他方式表達同一種意思。為因應英文書目資料中篇名、摘要、正文等欄位中各種可能的表達方式。譬如 Education 的各種不同詞型變化有 Educate, Educational, Educates, Educated 等字都是。因此，許多系統提供截字符號，如 Truncation 或 Wildcard 等功能。事實上，截字功能就是一種隱喻式聯集表現，一般查詢者未必完全瞭解這點。

依本題之題意，中華民國、台灣及台北是同義詞，經貿及貿易也為同義詞。為簡易本例說明，『台灣』OR『臺灣』及『台北』OR『臺北』暫不考慮。(圖五)中所顯示的同義詞先做聯集。並各用括弧先括之。再將不同主題間交集之。就成了『標準型單行指令』：

(台灣 OR 中華民國 OR 台北) AND 美國 AND (貿易 OR 經貿)

其中(台灣 OR 中華民國 OR 台北) 及 (貿易 OR 經貿) 先自行做聯集，(圖六)Venn Diagram 就不做詳細表示。上面單行指令在 NOT/AND/OR、NOT/OR/AND 或由左而右(由右而左)的運算優先順序下皆可行。

(台灣 OR 中華民國 OR 台北)      美國



(貿易 OR 經貿)

(圖六)

若運算符號優先順序以 NOT/ AND/ OR 的狀況，(圖五)可行的單行指令就是『標準型單行指令』(台灣 OR 中華民國 OR 台北) AND 美國 AND (貿易 OR 經貿)。而且，這是唯一正確的指令。

若在運算符號優先順序以 NOT/OR/AND 的狀況，其可行指令是下列任一：

(台灣 OR 中華民國 OR 台北) AND 美國 AND (貿易 OR 經貿)  
台灣 OR 中華民國 OR 台北 AND 美國 AND 貿易 OR 經貿

若在運算符號優先順序以由左而右(由右而左)，其可行指令可能是下列任一：

(台灣 OR 中華民國 OR 台北) AND 美國 AND (貿易 OR 經貿)  
(((台灣 OR 中華民國) OR 台北) AND 美國) AND (貿易 OR 經貿)  
台灣 OR 中華民國 OR 台北 AND 美國 AND (貿易 OR 經貿)

當第一次操作一新系統時，若該系統的線上說明沒有明述該系統運算符號之優先順序時，查詢者如何來確定？筆者建議如下：先將標準型單行指令中所有之括弧去之(暫時不要顧及其邏輯關係正確否)，執行該指令。以此『測試型指令』

台灣 OR 中華民國 OR 台北 AND 美國 AND 貿易 OR 經貿

執行之，若得筆數暫訂為 X 筆數。再以該行指令分別模擬成在 NOT/AND/OR 之下(也就是用假設的括弧括出先執行的部份，也就是 AND 的部份)；及 NOT/OR/AND 的狀況，(先用則以假設的括弧括出先執行的部份，也就是 OR 的部份)；由左而右的狀況，以下列的三行分別試之

NOT/AND/OR:

台灣 OR 中華民國 OR (台北 AND 美國 AND 貿易) OR 經貿

NOT/OR/AND:

(台灣 OR 中華民國 OR 台北) AND 美國 AND (貿易 OR 經貿)

由左而右：

((((台灣 OR 中華民國) OR 台北) AND 美國) AND 貿易) OR 經貿

其中之一若結果為 X 值的話，即可確認該系統的運算優先順序了。

在(圖五)資訊檢索策略圖中的同位詞是用在題意中若兩個主題詞地位相同且其中任何一者出現即可時，就稱之為「同位詞」。舉另外一例說明。

譬如：題目是「檸檬與橘子中維他命 C 之成份比較」，若查詢者是營養專家，在 NOT/AND/OR 之下可以將它化成指令

(檸檬 OR 橘子) AND 維他命 C

該指令會查到只談橘子或檸檬其中一種水果的維他命 C 之文章外 [註：=(((檸檬 OR 橘子) NOT (檸檬 AND 橘子))AND 維他命 C) 即 Exclusive OR ]。當然，同時兼論該兩種水果的維他命 C 之文章也會查到 [註：= (檸檬 AND 橘子) AND 維他命 C。]，二者水果任一均可。其中「檸檬」與「橘子」是同位詞。在 NOT/AND/OR 之下，"(檸檬 OR 橘子) AND 維他命 C" 是唯一的指令。但在 NOT/OR/AND 或由左而右之下，尚有另一組答案：檸檬 OR 橘子 AND 維他命 C。

若查詢者是非營養專家時，需要文章中同時兼論橘子及檸檬二者才在收錄範圍的話，在 NOT/AND/OR 的運算符號的優先順序下，指令就變成

(檸檬 AND 橘子) AND 維他命 C  
或 檸檬 AND 橘子 AND 維他命 C

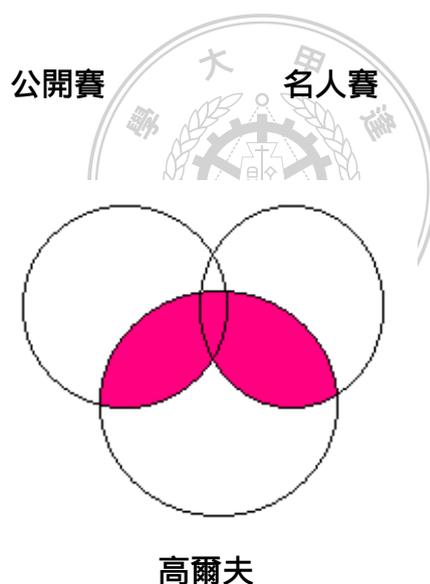
在 NOT/OR/AND 或由左而右下，這二組答案均可被接受。有些人可能反駁以()括弧就可以解決布林邏輯運算優先順序的問題，未必全然正確。有些系統提供暫存集合，人機交互反應時，唯有靠正確之布林邏輯運算觀念才不致於迷失其中。再者，下面例子更可為佐證。

在許多系統所提供的填充式的由上而下的下拉式畫面中有可選擇之布林邏輯符號：NOT, AND, OR。有的是統一式(如圖二)，有做上下行之間布林符號的選擇(AND 或 OR)；有非統一式，如 TTSWeb。兩者都沒有提供查詢者上下行之間括弧的選擇。前者尚未造成誤解，後者會造成詮釋布林邏輯關係的另一種特殊的狀況。以 TTSWeb 為例，在同一種畫面中提供兩種查詢方式：單行指令及由上而下的查詢模式。單行指令其布林邏輯的運算優先順序是由左而右；但是其由上而下的運算優先順序是由上而下。前者系統有明確說明，但後者則不。而且系統並不提供上下行間的括弧，也不允許查詢者在兩行之間下()。這對於指令的下達有差別嗎？筆者舉例試比較說明。

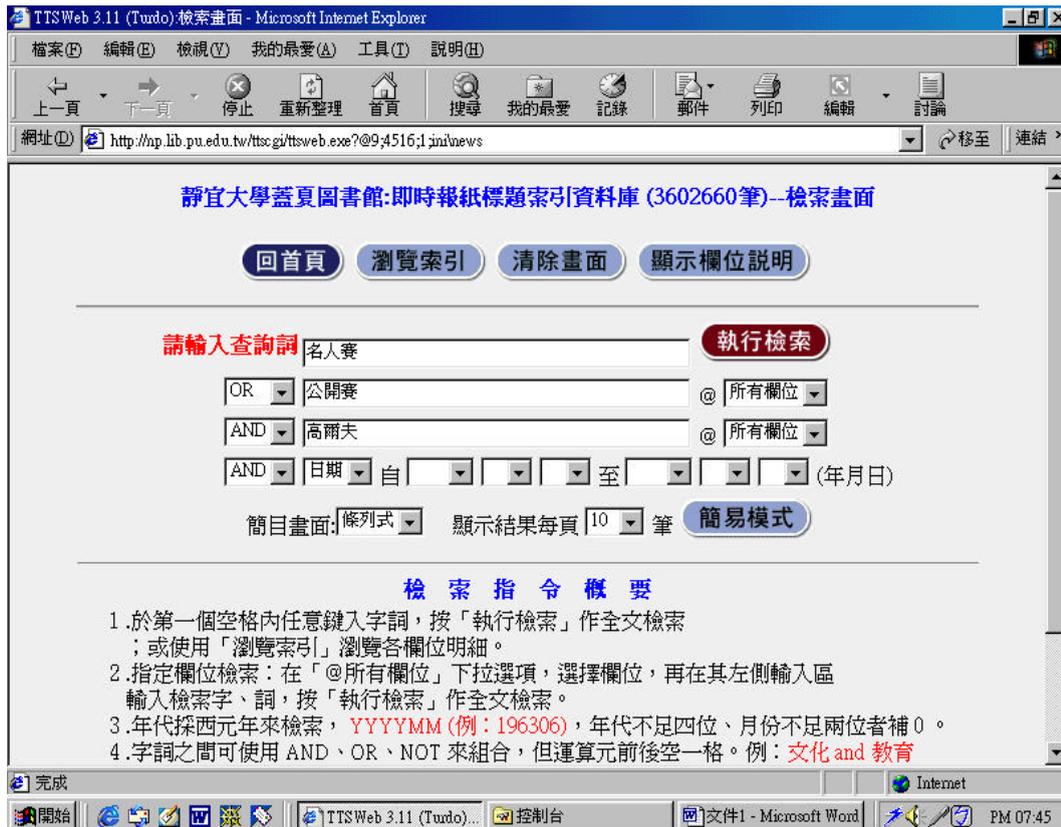
譬如欲查有關「高爾夫公開賽或高爾夫名人賽」的文獻時，其 Venn Diagram 如(圖七)。在 TTSWeb「即時報紙標題索引」的查詢畫面先以單行指令來下 "高爾夫 AND (名人賽 OR 公開賽)"，或 "(名人賽 OR 公開賽) AND 高爾夫"，或 "名人賽 OR 公開賽 AND 高爾夫" 三者所得之結果都正確，因該系統在單行指令時其運算順序是由左而右。但該指令若改成 "高爾夫 AND 公開賽 OR 名人賽"，或 "高爾夫 AND 名人賽 OR 公開賽"，那就與題意相違了。

本題在由上而下的填充式查詢畫面裏，原來單行指令就轉換成了(圖八)。其中"名人賽"與"公開賽"二者可以上下對調，其結果都一樣且吻合題意。但是若將之轉換成(圖九)的話，就會查獲圍棋名人賽或網球名人賽的錯誤資訊。

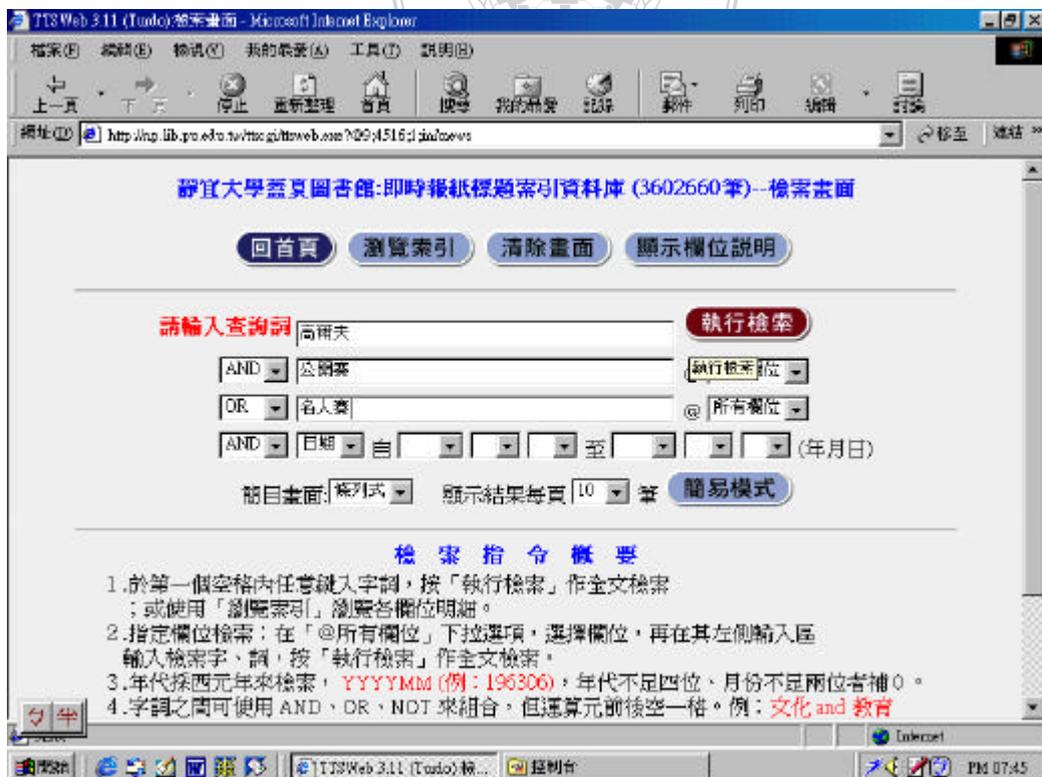
以上的說明表達由左而右的布林邏輯關係並不相等於由上而下的布林邏輯關係。



(圖七)



(圖八)



(圖九)

當前，許多資料庫的走向是多元化的主題、多種化的期刊收入。其目的就是希望一種資料庫就可以滿足各種讀者多樣化的資訊需求。同詞異義(Synonym)在不同學科領域有不同的定義(像"介面"一字在電腦學科、化學就有不同之定義)，所以，一行指令可能查出不同學科或主題的期刊文獻。導致查詢者須花許多時間在查詢結果中過濾或剔除不相干的資料。以 Bell & Howell 的 Academic Research Library 為例，其收入的刊物達 1,800 種，主題涵蓋有人文、科技、社會科學、自然科學等。所收入期刊的層次是通俗性、學術性兼收。所以，查詢者可能蒐集到不同領域及不同層次的大量文獻。卻須花許多心力在刪除不合理想的資料。筆者以下建議，僅供做參考。資料庫的建檔者(製造商)對所收錄的期刊做細類主題的分類，以每種期刊單元做做分類。譬如電腦的期刊可先分為軟體、硬體；在軟體下在細分為作業系統、介面設計、通訊、安全防護等。在期刊內容的層次方面，像 ProQuest Direct 下有同儕審核的刊物的選擇，仍嫌粗糙。所以，建議在學術性刊物中以每篇文章為單元，細分總覽性(對一主題的歷史及未來發展的介紹)、實驗性(做先生性的試驗)、評估性(對同一主題做兩種不同的研究方法之比較)、及深入性(對一主題有突破性的見解)。雖說 ProQuest Direct 下 Academic Research Library 提供另類的選擇，目前有 Document Type 欄位下分 Review(書評或影評), Feature(專稿), Speech(演講), Fiction(小說), Editorial(編者的話), Instructional(介紹性)等分類，筆者仍嫌不足。以上所建議的，可以在查詢畫面由查詢者在下指令時，同時也勾選文獻的主題及層次的需求。這種隱喻式交集(Implied AND)出來的文獻應該會比較提昇滿意程度。

總論：在當前布林邏輯的查詢系統仍為主流。可是，當系統設計者在介面的設計及線上說明不是那麼明確又易解地傳達訊息給查詢者的話，縱然有多強的搜尋引擎也枉然。然而以館員的立場，一個資料庫的引渡常是對資料庫內容重於對系統好壞的考量。所以，在讀者教育上更須花把勁來教育查詢者。或者，將將系統中的盲點或缺點轉告系統廠商，以便改進。筆者歸納如下：從資訊需求到資訊檢索策略圖(如圖五)，這一定要讓查詢者能掌握。其次，詞藻之間、主題之間、個人特殊需求、集合與欄位(或全文檢索)之間、等等交錯組合的布林邏輯關係，加上邏輯的優先順序讓查詢者清楚理解。換言之，協助查詢者建立批判性的布林邏輯思考模式。再隨著不同的查詢介面做調整，這才是上策的讀者教育。

參考書目：

- Barnett, A.(1999). A survey of Internet Searches and their results. Reference & User Services Quarterly. 39(2), Winter, 177-181.
- Cooper, W.S.(1997). Exploiting the Maximum Entropy Principle to Increase Retrieval Effectiveness. Journal of the American Society for Information Science. 34(1). 32-45.
- Ensor, P. (1992) Knowledge Level of Users and Nonusers of Keyword/Boolean Searching on an Online Public Access Catalog. RQ 31, Fall. 60-74.
- Frants, V. I., et al. (1999) Boolean Search: Current State and Perspectives. Journal of the American Society for Information Science 50(1), January, 86-95.
- Leverence, M. E. (1994) Time-tested ERIC Bibliographic Instruction(BI): Give Them a Pole and Teach Them to Fish(Fiche). ERIC ED 381 123. May 19, 1994.
- Mader, S., (1989). Teaching End Users to Search Online Information Retrieval System: Lesson from the Field. In Online Searching : The Basics, Settings, and Management. Ed. Joann H. Lee. 2<sup>nd</sup> ed. Englewood, CO: Libraries Unlimited. 63-66.
- Nahl, D. & Harada, V. H. (1996) Composing Boolean Search Statements: Self-Confidence, Concept Analysis, Search Logic, and Error. School Library Media Quarterly. Summer, 199-206.
- Savoy, J., (1997). Ranking Schemes in Hybrid Boolean Systems: A New Approach. Journal of the American Society for Information Science. 48(3), 235-253.
- Willet, P. (ed.) (1996). Document retrieval system. London: Taylor Grm.,.