

人工智慧之另一方向——專家系統

高中組應用科學科第二名

台南一中

作　　者：魏川淇、黃政陽
指導教師：林家模

一、研究動機

每當我們到醫院看病時，醫生總是不厭其煩地詢問病人有什麼病情。如有沒有發燒呢？有沒有頭痛呢？有沒有流鼻涕呢？問了些問題後，醫生就可以下判斷：你得了××病。由此可知，似乎只要有一些規則，我們就可以判斷有何結果。例如：你既頭痛頭昏又打噴涕，你可能得了感冒。當然，以上的例子只是隨便說說而已，但若是實際上請醫術高明的醫生，把一些診斷疾病上的徵兆或條件，寫成明確的規則，交由電腦去處理，如此診斷的工作就可交給電腦，而一般不懂醫術的人也可藉助電腦自行判斷疾病。

二、研究目的

我們都知道，電腦其實只看得懂一些很粗陋的符號，偏偏這些符號又非我們一般人可接受。程式設計師的工作，就在於設計有親和力的程式，使一般不懂電腦的人能很輕易的操作。我們最初目的是設計一套能處理規則的診療系統，在精益求精之下，竟發展成一套多功能的軟體系統。只要是合乎邏輯，有規則的條件，都可藉此程式求得結論。而我們現在的工作，就是設計一套能處理規則的程式，而規則的輸入要儘量口語化和隨意，且讓不懂電腦的人也能很方便的使用。

三、研究設備器材

(一)硬體部分：

1. 16Bit電腦一部 (80286 20MHz PC/ AT)
2. 24Pin列表機一台 (Epson LQ 2500C)
3. 磁片數張 (Fuji 2HD)
4. 人腦兩具 (Made in Taiwan)

(二)軟體部分：

1. Turbo Pascal 5.5一套
2. GW-Basic 2.0一套
3. IBM DOS 3.3一套
4. 倚天中文系統 飛碟一號ET-2416F V1.52
5. 龍碟中文系統 V3.1

四、研究過程或方法

(一)事實上，此程式有商業化之潛力，我們一方面研究其理論上之處理程序，一方面應用其程序設計程式，使之能符合商場上之要求。因此在研究理論上，我們以除錯容易之Basic寫一個實驗版本，雖然並不實用，但在研究理論上極為方便。而實際應用上我們以結構化且速度快之Turbo Pascal 5.5來製作程式。Turbo C雖然功能強大，內部運算快（約較Pascal快1.7倍），但保養不易，函數大且編輯慢，因而不考慮採用。

(二)要教電腦憑空推理事物，那是不可能的事，唯有透過程式的溝通，才能賦與電腦生命。如今我們的工作是製作一個可供查詢的專家系統，我們的思考路線如下：

1. 要讓電腦有判斷的依據，也就是給電腦什麼是對，什麼是錯的資料庫。
2. 要給電腦一個脈絡可尋，否則茫茫資料庫教電腦從何下手？

3. 要隨時給予資訊，讓電腦知道你需要什麼，進而滿足你的需要。

但電腦和使用者之間要如何溝通呢？這是我們遭遇到的第一個問題。

(三)我們都生過病，看過醫生。醫生並非病人，但醫生總有辦法判斷病人得了什麼病。醫生用什麼方法呢？不外乎以詢問的方式。我們先假設世界上只有8種病，分別為感冒、腸炎、霍亂、狂犬病、砂眼、AIDS、偏頭痛及白血症。如果醫生問你有什麼症狀，你說你有發燒，此時醫生得到你有“發燒”的資訊，然後把不會引起發燒的病砂眼及偏頭痛刪除，剩下感冒、腸炎、霍亂、狂犬病、AIDS、白血症這六種。接著你說你還有流鼻涕，於是醫生就刪掉不會流鼻涕的腸炎及狂犬病，剩下感冒、霍亂、AIDS、及白血症四種。然後你說你還有咳嗽和發燒，於是醫生再刪掉不會引起咳嗽的白血症及霍亂刪除，把不會引起鼻塞的AIDS除去。於是醫生可以下定論：你得的是感冒。這種每問一個條件，再刪掉一部分俾獲致結果的方式是方法之一。

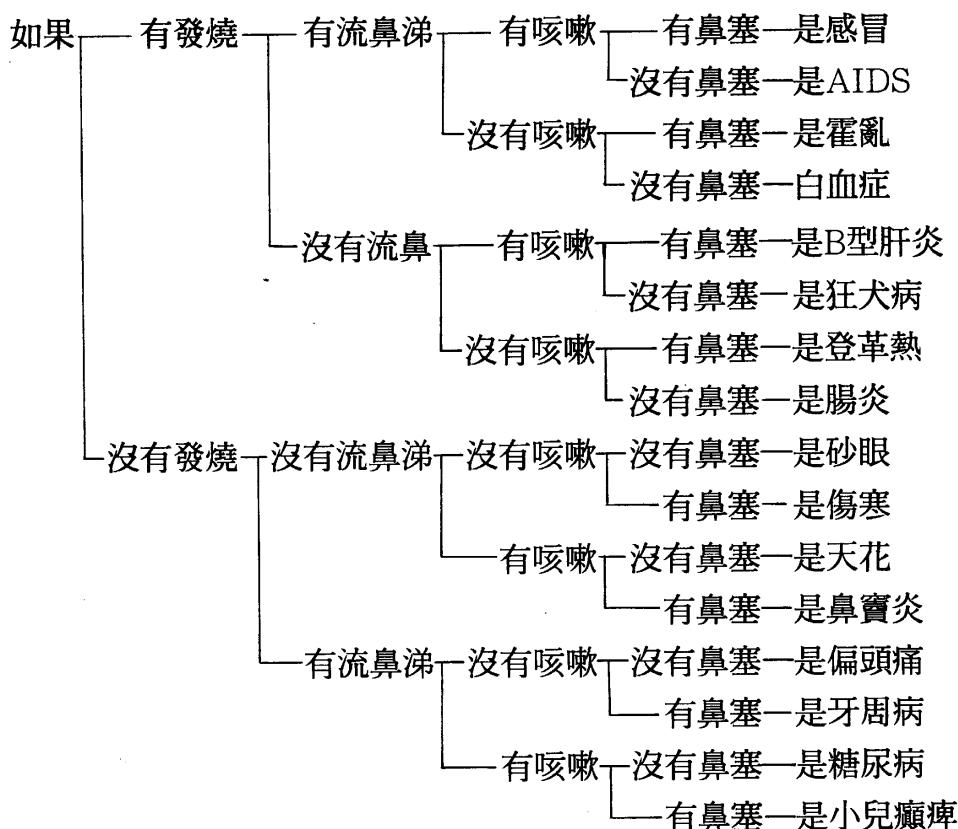
(四)我們可以教電腦模擬這種方式和使用者溝通，但首先電腦得先知道一些事，如AIDS不會引起鼻塞，狂犬病不會流鼻涕等。以下是上述症狀的規則圖：

有發燒	有流鼻涕	有咳嗽	有鼻塞	→是感冒
有發燒	有流鼻涕	有咳嗽	沒有鼻塞	→是AIDS
有發燒	有流鼻涕	沒有咳嗽	有鼻塞	→是霍亂
有發燒	沒有流鼻涕	有咳嗽	沒有鼻塞	→是狂犬病
有發燒	有流鼻涕	沒有咳嗽	沒有鼻塞	→是白血症
有發燒	沒有流鼻涕	沒有咳嗽	沒有鼻塞	→是腸炎
沒有發燒	沒有流鼻涕	沒有咳嗽	沒有鼻塞	→是砂眼
沒有發燒	有流鼻涕	沒有咳嗽	沒有鼻塞	→是偏頭痛

照組合來看，四種病徵可決定二的四次方共十六種病情，但這裡只列出八種，表示資料庫不足，（其實是限於篇幅）另外的

八種要記得補齊。不過到目前為止我們在算幾種情形都是以二為底，這是因我們都一直以“有”及“沒有”來決定。其實我們也可以用“輕微發燒”及“發高燒”等多樣化的選項來決定，並非只有兩種選擇。

(五)如此一來脈絡是清楚多了，但有一個缺點，就是重覆率高。試想，若病徵有二十種，而選項以“有”及“沒有”選擇，就是兩種選擇而已，以上列表就要一百萬行！要依序寫出來當然可以，我們對這種有恆心有毅力的表現，一定會非常感動的，但事倍功半。為了改進這問題，我們不得不刪掉第一種方法——刪除法（其實這種方法還有很多缺點，留待後述），改用新的方法——樹狀排列法。樹狀圖如下：



還好是樹狀圖，否則以條列狀排列會非常壯觀！這又有一個優點，就是脈絡更明顯，更節省空間。

(六)目前為止，電腦的資料儲存結構已大致定案，理論上已可以開始製作程式。但現在我們碰到另一個問題，那就是溝通上的問題。電腦並聽不懂你在說什麼，因此我們訂定一些文法以方便電腦溝通。以下就是我們和電腦的溝通法則，也是此程式之使用說明，使用者可用PE II 或Word Star等文書編輯軟體先行鍵入再進行執行。

1. 規則之啟始——命名 (NAME)。給資料庫一個名字，這樣的話有個好處——若是將來資料庫太多，可以藉此辨別。
(最多78字)

name疾病判斷診療系統

2. 規則之指標——步驟 (STEP)。一開始讀到name後的名字，接著電腦會馬上找Step。Step後接一比較元，另外還附有相對於if之位置，當電腦找到Step後，順著附加之if位置去尋找比較值。當“Step比較元”後還有說明時（以引號括起來）會先列出說明再跳至if。舉例如下：

Step有無發燒

“請問有無發燒？”

“若37度以上算發燒。”

3. 規則比較元之提示——提示 (PASS)。這是個可有可無的一項。當程式進入比較元時，如碰到if xxx is yyy時，xxx這個比較元應該要有些提示說明，否則程式將列出較無意義的問題“What or How is”。舉例如下：

pass有無流鼻涕

“請問有沒有流鼻涕”

4. 規則之主體樹——如果 (IF)。每棵樹都是由if作起始，而樹枝(連線)則是以“And”及“＼”表示。我們可以這樣說：當比較元之比較值符合時，電腦走“And”的路線，否則走“＼”的路線。最後以“then”結束一棵樹。

if 有無發燒 is 有 \若沒有樹枝
and 有無流鼻涕 is 有
then 疾病 is 重感冒
\若沒有樹枝 沒有
and 有無流鼻涕 is 沒有
then 疾病 is 小感冒

5. 規則之說明——說明 (MEMO)。這個指令可放於任何地方，且不影響資料庫之運作。舉例如下：

MEMO 這只是個小範例。

(七)我們又碰到新的問題：由於資料庫是文書檔，並不能像剛才所說的跳來跳去。怎麼辦呢？有一種辦法，就是利用陣列，把全部的資料讀入記憶體。但以PC有限的記憶體，若是小的資料庫還消受的起，但以實用的觀點來看，照理說實用的資料庫都是非常龐大的。於是我們把記憶體移向磁片，以隨機檔代替陣列。我們設計一個Link的程式，把文書檔換成隨機檔，並順便除錯。有些錯誤是允許的，如太多空白、空行、大小寫交錯等，都可藉此由電腦自行更正。最後我們舉例一個資料庫做結束，它可以判斷您手中所持的錢金額為多少並提供建議！

NAME 判斷您手中的錢面值多少及建議

MEMO 這是一個小小的範例。

STEP 外觀

“你現在拿的錢為何？”

“是紙鈔還是硬幣？”

STEP 幣值

“請問您所擁有的錢幣值為何？”

PASS 人像

“在紙幣上的人像為何？”

MEMO 請特別注意，並非每個問題都有題示。(如顏色)

IF 外觀 is 紙鈔 \連到硬幣外型

and 人像 is 國父 \連到其它

and 顏色 is 紫色 \或是紅色

then 幣值 is 伍十元

memo 注意！所有條件都要放在 I F 之後，但並不限與其它條件之樹交叉。

memo 如上面是一棵樹的開首，但其分支竟已延到下面幣值這棵樹後面去了。

memo 事實上，以下這棵樹在執行上並不須輸入答案，

memo 因為上一棵樹已告知本題的答案（有關幣值部分），對於已學過的東西，

memo 電腦會自動跳過，不重覆學習。

if 幣值 is 一佰元＼50

then 你的錢最好的用途 is 去買三片磁片。

＼50 伍十元＼1

then 你的錢最好的用途 is 去買一個漢堡。

＼1 一元＼500

then 你的錢最好的用途 is 積少成多，存起來吧。

＼500 伍佰元＼1000

then 你的錢最好的用途 is 去買一盒磁片。

＼1000 一千元＼5

then 你的錢最好的用途 is 去買一隻滑鼠。

＼5 伍元＼100

then 你的錢最好的用途 is 或許該可買包乖乖。

＼100 一佰元＼10

then 你的錢最好的用途 is 買三片磁片。

＼10 十元＼.5

then＼你的錢最好的用途 is 買包洋芋片算了。

＼.5 伍角

then 你的錢最好的用途 is 存起來算了，畢竟這種錢並不多見。

memo 以下為第一棵樹（幣值）之分枝。

＼或是紅色 紅色

then 幣值 is 一佰元

\連到其它 總統

and 顏色 is 紅色 \藍色

then 幣值 is 伍佰元

\藍色 藍色

then 幣值 is 一千元

\連到硬幣外型 硬幣

and 顏色 is 銀色 \或是黃色

and 人像背後 is 有梅花 連沒有梅花

then 幣值 is 十元

\連沒有梅花 沒有梅花

then 幣值 is 伍元

\或是黃色 黃色

and 有沒有人像 is 有 \沒有

then 幣值 is 一元

\沒有 沒有

then 幣值 is 伍角

五、實驗結果

以下是Pascal實用版之執行結果，它是以上列之資料庫運作。其中標題我們只列出一次，其餘是把重要部分列印出來。限於篇幅，未能把全部程式列出，如有興趣研究此程式，請洽台南一中工藝館。

執行程式(→)：Link.Pas 對以上之資料庫做轉換及除錯。

B>link6 money3

Expert System Link & Debug System V2.0 by William

Wei with Turbo Pascal 5.5

Read data from file : MONEY3.KB

Write data to file : MONEY3.CON

Get knowledge NAME !

Get knowledge STEP !
Get knowledge PASS !
Get knowledge IF !
Search and check STEP to IF.
Search first PASS and IF state.
Check JUMP after if, and or \.
Ok! All are through !
There are 73 lines in the source file.
And 48 lines in the target file with ".CON" .

B>

執行程式(二) : expert.pas. 為主程式。

EXPERT SYSTEM by William Wei

判斷您手中的錢面值多少及建議 !

你現在拿的錢為何 ?

是紙鈔還是硬幣 ?

Your choice : (Please use Arrow UP, DOWN, RIGH
T, LEFT)

紙鈔 硬幣

QUIT

在紙幣上的人像為何 ?

Your choice : (Please use Arrow UP, DOWN, RIGH
T, LEFT)

國父 總統

PROCESS QUIT

What or How is 顏色 ?

Your choice : (Please use Arrow UP, DOWN, RIGH
T, LEFT)

紫色 紅色

PROCESS QUIT

Our result is that 幣值 is 一佰元

Stop working and exit to DOS. Ok ? No

請問您所擁有的錢幣值為何？

From memory of computer, We have know about :

幣值 is 一佰元

Press any key to continue. —

Our result is that 你的錢最好的用途 is 去買三片磁片。

Stop working and exit to DOS. Ok ? No

六、討論

(一)先前提到第一種方法並不能採用，原因如下

1. 由於它是先得到結果，再以刪除法刪除，因此一開始就得設立巨大的陣列以儲存結果。由於是以刪除法運作，因此結果必得設立一布林值以確認刪除否，這對碟片運作以代替陣列有困難（由於必需寫入，不能防寫，且執行一次得清除其布林值）。
2. 由於我們是先得到結論，再行淘汰，因此若資料庫撰寫有誤，往往很難追蹤，且會得到邏輯錯誤或多重結論，換句話說，寫資料庫的人若是一不小心寫錯資料，電腦本身就無法工作，甚至連除錯都很困難！

(二)第二種方法為本程式執行之方法，其運行路線如下：

1. 每碰到一個比較元，因為資料是以樹枝狀排列，因此電腦可以一下子把全部運算值找出，讓使用者選擇。再依使用者的選擇，進入下一個比較元，如此由廣漸深，向樹的末梢推進。
2. 使用此方法可以保証其結論必定存在且唯一，且在link時也可以順便試著運行其結果，達到除錯之目的。以下是程式的大致流程：
 - (1)Link部分：找出Name→找出Step→找出Pass1若無則提出警告→找出IF. And. 及then必關閉資料檔

→找出Step相對於IF之位→找出第一個IF及Pass之位→檢查Jump之相對位置→關閉隨機檔。

(2)Expert部分：找出Nmae→找出Step及其附加位址，必跳至IF→取其比較元→跳至Pass取提示→回到原位址取全部可能答案→選擇答案並進入下一比較元→直到then為止把then後之結論納入學習記憶庫→回到Step的步驟。

七、結論

研究人工智慧是現在每個科技進步的國家之首要目標，由於其強大的功能，及無可限量之前程，每個國家莫不把其研究結果納為最高機密。其實人工智慧並不如每個人想的那麼高深莫測，而是一種有技巧性、有結構性的技術，其理論並非很難。如今我們國家已邁入已開發國家，自然發展人工智慧也是必然之傾向，現在各國研究已有不小之成就，趁著各國尚未成熟，讓我們一同奮起直追！愈多人的投入，會使人工智慧的發展愈快、愈成熟。

專家系統是目前人工智慧中理論最淺顯，也是最有成就的一門技術，但其中仍有無數可開發的地方，如使資料庫更口語化，使溝通更深入，使其作運算功能等，都是今後可繼續發展的地方。

八、參考資料

- (一)Turbo Pascal 5.5 使用手冊
- (二)Turbo Pascal 5.5 程式設計
- (三)人工智慧Basic 程式設計技巧
- (四)專家系統理論與設計

評語

本作品對於以規則為基礎的人工智慧系統 (rule-based expert system) 有相當了解，處理「若…則」(If…then) 之句形所表達之知識，亦頗中規矩，高中程度能做到這程度，已屬難得。唯此類研究涉及之基礎甚廣，以高中程度做來，自然有許多限制。在學術知識上之追求，是更上層樓之不二法門。