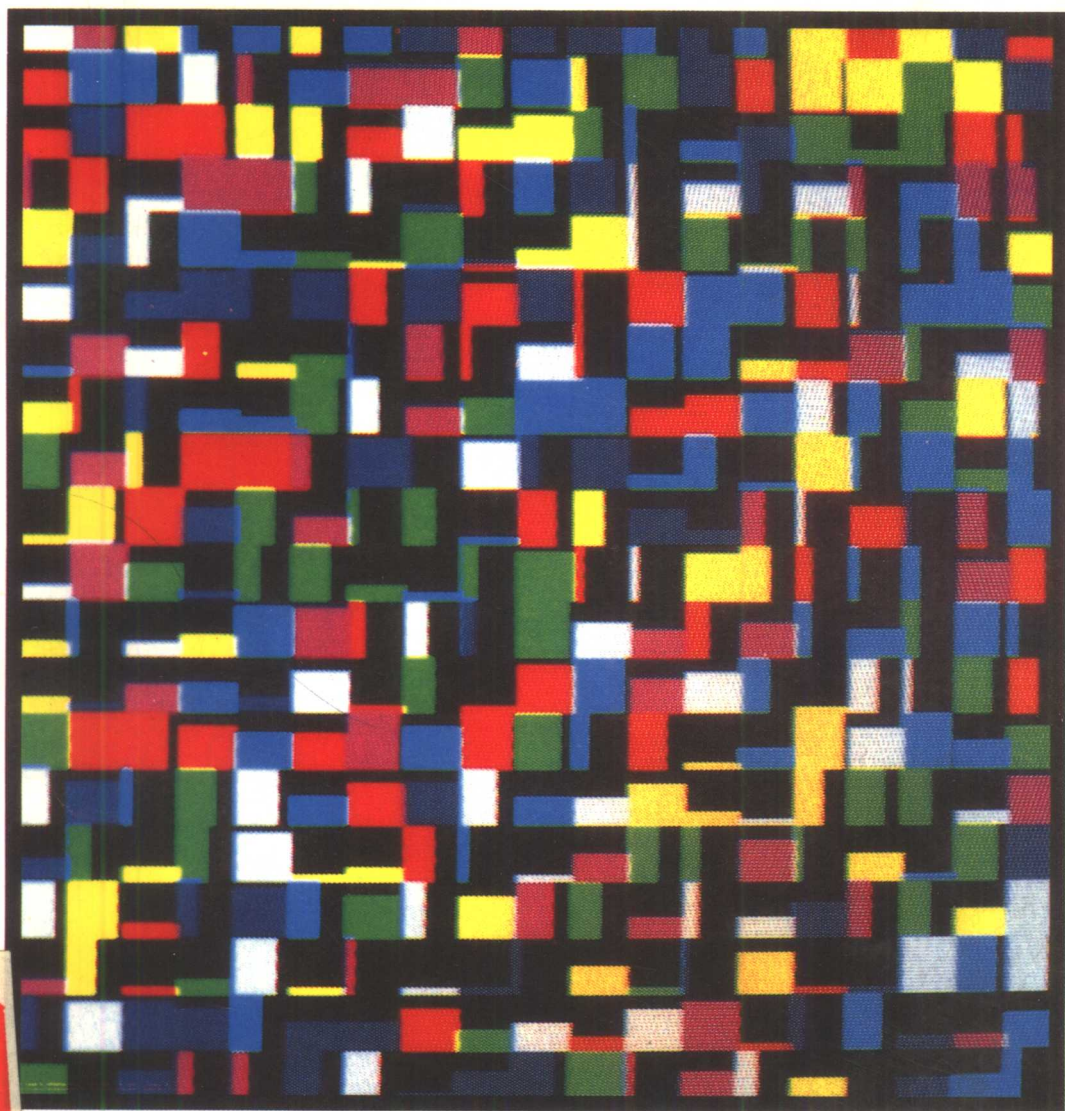


---

# 知識庫系統簡介



許博傑 編著

---

# 知識庫系統簡介

許博傑 編著

儒林圖書公司 印行

---

---

知識庫系統簡介

編 者：許 博 傑

發行人：楊 鏡 秋

出版者：儒林圖書有限公司

地 址：台北市重慶南路一段 111 號

電 話：3118971-3 3144000

郵政劃撥：0106792-1 號

吉豐印刷廠有限公司承印

板橋市三民路二段正隆巷 46 弄 7 號

---

---

行政院新聞局局版台業字第 1492 號

中華民國七十七年元月初版

定價新台幣 170 元正

# 序

本書介紹了電腦科學家們在自動化知識處理方面的表示法和技巧，知識庫系統和資料庫系統不同的地方在於知識庫系統有自動推理的能力。知識表示法及推理的方法都有很多種，但是其中只有命題邏輯 (Propositional Logic) 及敘述邏輯 (Predicate Logic) 有形式化的推理系統，並且是所有知識表示法中最普及的方法，因此本書的重點在於介紹命題邏輯和敘述邏輯的語法 (Syntax)、模型 (Model) 及證明理論 (Proof Theory)，一般談邏輯的書都過於數學化，不適合初學知識庫系統的同學們閱讀，本書以口語化的方法，深入淺出的說明邏輯推理，而沒有討論邏輯本身的嚴格性及完整性的說明，很適合初學知識庫系統的同學們閱讀。除了命題邏輯和敘述邏輯之外，還有許多更複雜的知識表示法及推導系統，但是由於篇幅的關係，這些表示法將在第二冊討論。

本書共有五章，第一章為概論，介紹什麼是知識庫、知識庫系統、知識庫系統和專家系統之間的關係、以及目前知識庫發展的情況，讀完這章之後會對於知識庫系統有一個整體的概念。第二章介紹知識、資訊、知識在自然語言的表示法，在邏輯上的表示法，在一般資料庫的表示法及在程式語言的表示法，本章並且說明如何製作一個形式邏輯語言剖析器 (Parser)，可以做為“編輯程式法” (Compiler/Translator) 課程的教材。第三章是本書的重點，說明知識庫系統和形式邏輯之間的關係，介紹形式邏輯的語法、模型及證明理論，有：分解法 (Resolution) 及非分解法，本章所討論的分解法有線性輸入

(Linear Input)、LUSH、SL 及非子句形式連接圖形分解法 (Connection Graph)，非分解法有自然推導法 (Natural Deduction)、繼起邏輯 (Sequent Logic)、布幕法 (Tableau Method)、矩陣及連接法等。第四章討論邏輯和資料庫系統的關係，第五章做一個總結論，將本書所提到的概念做一個綜合性的整理，並介紹現有的知識庫系統。

作者發現在國內沒有介紹知識庫系統方面的書籍，在國外的書籍和論文中大部份也都是有許多的內容討論邏輯本身的嚴格性和完整性的說明，對於沒有很熟悉數學邏輯證明的人來說不容易看得懂，因此特地編譯此書，作者在編譯的過程中會有疏漏的地方，請各位讀者不吝指正。

許 博 傑

76.10.25

# 目 錄

|                                               |    |
|-----------------------------------------------|----|
| 序 .....                                       | V  |
| 第一章 概 論 .....                                 | 1  |
| 1.1 本書簡介 .....                                | 1  |
| 1.2 知識庫 .....                                 | 2  |
| 1.3 知識庫系統 .....                               | 4  |
| 1.4 目前在KBSs 的發展情況 .....                       | 6  |
| 1.5 資料庫技巧、形式邏輯、專家系統及自然語言處理的概念及技<br>巧的整合 ..... | 6  |
| 第二章 知識及知識表示法 .....                            | 9  |
| 2.1 知識 .....                                  | 9  |
| 2.1.1 知識的定義 .....                             | 9  |
| 2.1.2 資料 ( Data ) .....                       | 11 |
| 2.1.3 資訊 ( Information ) .....                | 11 |
| 2.1.4 哲學上的一些考慮 .....                          | 11 |
| 2.2 知識表示法的形式化 .....                           | 12 |
| 2.2.1 知識在自然語言上的表達法 .....                      | 12 |

|       |                      |    |
|-------|----------------------|----|
| 2.2.2 | 知識在資料處理及一般資料庫系統上的表示法 | 13 |
| 2.2.3 | 知識在形式邏輯語言上的表示法       | 13 |
| 2.2.4 | 為命題邏輯建構一個文法剖析器       | 19 |
| 2.2.5 | 知識在生產規則系統中的表示法       | 29 |
| 2.2.6 | 知識在填空系統的表示法          | 31 |
| 2.2.7 | 知識在函數式的表示法           | 32 |
| 2.2.8 | 知識在程式語言的表示法          | 34 |
| 2.3   | 結 論                  | 35 |

## 第三章 形式邏輯概論 37

|       |                              |    |
|-------|------------------------------|----|
| 3.1   | 邏輯概觀                         | 38 |
| 3.1.1 | 非常不正式的討論推論                   | 38 |
| 3.1.2 | 邏輯語言                         | 41 |
| 3.1.3 | 形式推導系統                       | 45 |
| 3.1.4 | 嚴格性和完整性                      | 48 |
| 3.1.5 | 理論和定理證明                      | 49 |
| 3.1.6 | 單純的語法定理證明                    | 52 |
| 3.1.7 | 單純的語意定理證明                    | 53 |
| 3.1.8 | 反證程序 (Refutation Procedures) | 54 |
| 3.1.9 | 決定性 (Decidability)           | 55 |
| 3.2   | 正統命題邏輯                       | 56 |
| 3.2.1 | 命題邏輯的一個公設系統                  | 57 |
| 3.2.2 | 自然推導證明法                      | 63 |
| 3.2.3 | 繼起邏輯／繼起證明                    | 66 |
| 3.2.4 | 由上而下的證明法                     | 69 |
| 3.2.5 | 布幕式的證明方法                     | 70 |
| 3.2.6 | 利用語意布幕來搜尋反例以做恒真性檢查           | 75 |
| 3.2.7 | 一致化證明法和恒真性檢驗法                | 78 |
| 3.2.8 | 分解法                          | 79 |

|        |                                            |     |
|--------|--------------------------------------------|-----|
| 3.2.9  | 非分解法的一致定理證明法 .....                         | 87  |
| 3.2.10 | 定理證明在命題邏輯上的例子 .....                        | 87  |
| 3.2.11 | 在命題邏輯上自動推論的程式實作 .....                      | 89  |
| 3.3    | 一階敘述邏輯 (First Order Predicate Logic) ..... | 107 |
| 3.3.1  | 一階敘述邏輯語言之語法 .....                          | 109 |
| 3.3.2  | 一階語言的語意 .....                              | 112 |
| 3.3.3  | 敘述邏輯上的自然推導 .....                           | 122 |
| 3.3.4  | PRENEX 形式 .....                            | 124 |
| 3.3.5  | 函數形式及 SKOLEM 形式 .....                      | 125 |
| 3.3.6  | 封閉式子的 HERBRAND 擴展 .....                    | 127 |
| 3.3.7  | 封閉的式子及 HERBRAND 擴展的一致性 .....               | 129 |
| 3.3.8  | CNF 和子句形式 .....                            | 131 |
| 3.3.9  | 敘述邏輯的分解法 .....                             | 132 |
| 3.3.10 | 利用分解法做一致性檢驗 .....                          | 138 |
| 3.3.11 | 在敘述邏輯上用分解法做定理證明 .....                      | 139 |
| 3.4    | 在 FOPL 上的自動定理證明 .....                      | 141 |
| 3.4.1  | 線性輸入分解法 (LINEAR INPUT RESOLUTION) .....    | 141 |
| 3.4.2  | LUSH 分解法 .....                             | 142 |
| 3.4.3  | SL 分解法 .....                               | 143 |
| 3.4.4  | 搜尋技巧 .....                                 | 146 |
| 3.4.5  | 非子句形式的分解法 .....                            | 150 |
| 3.4.6  | 利用連接圖形 (Connection Graphs) 做定理證明 ) .....   | 154 |
| 3.4.7  | 非子句形式的連接圖形分解定理證明法 .....                    | 159 |
| 3.4.8  | 非分解法的定理證明：自然推導 .....                       | 159 |
| 3.4.9  | 非分解法的定理證明：矩陣和連接法 .....                     | 159 |
| 3.4.10 | 在 FOPL 上自動證明程式的實做 .....                    | 167 |
| 3.5    | 結 論 .....                                  | 167 |

|                            |     |
|----------------------------|-----|
| <b>第四章 邏輯和資料庫系統</b>        | 171 |
| 4.1 概 論                    | 171 |
| 4.2 我們所用到的一些形式邏輯的概念        | 175 |
| 4.3 理論和完整的關係結構方法           | 176 |
| 4.3.1 整合 ( Integrity ) 的維護 | 177 |
| 4.3.2 查詢的求值                | 179 |
| 4.3.3 定義過的關係               | 181 |
| 4.3.4 封閉空間、定義域封閉及單一名稱假設    | 182 |
| 4.3.5 理論和完整的關係結構方法的適用性     | 182 |
| 4.4 純理論的方法                 | 183 |
| 4.4.1 一致性檢查                | 185 |
| 4.4.2 求查詢值                 | 187 |
| 4.4.3 將任意的查詢換算為完整的開放查詢     | 193 |
| 4.4.4 封閉空間假設／無法證明則視為負號的    | 199 |
| 4.4.5 純理論方法的適用性            | 203 |
| <b>第五章 結 論</b>             | 207 |
| 5.1 現有的知識庫系統               | 207 |
| 5.2 本書所提到的概念及技巧的總整理        | 211 |
| 5.3 建議讀者再看的書               | 212 |
| <b>中英名詞對照</b>              | 215 |
| <b>參考書目</b>                | 225 |

# 第一章

## 概 論

### 1.1 本書簡介

人類的知識(knowledge)實際上是如何運作的,到底有多少生理機能牽涉在裏頭,到目前為止還不是很清楚,不過,最近三千年來,人類在儲存和處理知識的能力比以前進步了,造成進步的主要因素之一就是知識溝通技術的進步以及機械化處理知識方法的發展。

在人類演進過程的初期,手勢和說話是溝通知識的方法,後來,更用了各種比較耐久的溝通方式,如:繪畫、象形文字及寫字等,由於各種溝通技巧的成長,人類也發展出一套方法來處理知識,使得自己的思考方式能夠藉著這種方法表達出來,例如,數學表示法及數學系統的發展是用來做數字計算,形式邏輯(formal logic)是用來做推理,圖書館及檔案系統是用來做大量知識的管理,近年來,我們更由於發明及利用機械、電子、光學系統做通訊、儲存及處理知識而有更大的進步,例如,資料庫管理系統(Data Base Management

## 2 知識庫系統簡介

System)能讓許多使用者同時存取大量同一規格的事實(Fact)，專家系統(Expert System)能夠利用簡單的事實及複雜的規則來做推論(Reasoning)，還有許多其他的系統有能力了解部份的自然語言(Natural Language)。

最近，大家對設計及實作知識庫系統(Knowledge Base System, KBS)的興趣越來越濃厚，本書就是知識庫系統的簡介，本章討論知識庫及知識庫系統包括那些部份，再來介紹資料庫技術，形式邏輯和專家系統的表示法及技巧是如何運作的，以及了解自然語言研究可能可以整合在電腦知識庫系統中，以提供一個以理論及實際為基礎的系統。

我們將在第二章概略的介紹一些知識表示法，第三章是形式邏輯概論，有許多人都覺得這一部份是非常難以了解的，本章嘗試以易讀的原則來討論形式邏輯，所以我們並沒有很正式的證明，形式邏輯在目前的知識庫系統中扮演一個非常重要的角色，因此第三章是本書的重點，第四章中我們列舉出目前已發展出來的一些知識庫系統，並且做個結論。

由於表達知識的方式很多，如果要一一詳加介紹，則會佔用太多篇幅，所以，我們打算在第二冊中再做詳細的介紹，另外，在日常生活中，我們所遇見的事情不見得都是十分確定的，這些問題也將在第二冊中介紹。

# 1.2 知識庫

在1.1節中，我們用了“知識庫”這個字眼，但是並沒有定義什麼是知識庫，到目前為止，還沒有一個被大家所接受的定義，而這個字眼對不同的人，應用在不同的領域也有不同的意思，本書中，知識庫的意思是：

“一些表示某些我們所討論的範圍(Universe of Discourse)的簡單事實(simple fact)及通用規則(general rule)”

這裏的“簡單事實”及“通用規則”、“我們所討論的範圍”等字眼，在後面幾章會有詳細的說明。

知識庫系統中的簡單事實可以比通用規則多，通用規則也可以比簡單事實多，例如：一個表示火車時刻表的知識庫就是前者，而一個好的表示下棋的知

識庫就是後者。

大部份知識庫和一般資料庫不一樣的地方在於知識庫包含了明顯表達 (Explicit) 的通用規則及簡單事實，而資料庫則包含了很多明顯表達的簡單事實及對應的一小部份隱含 (Implicit) 的通用規則。

例如，在管理醫院住院病人的應用中，每位病人的生日記錄方式在商一般料庫及知識庫系統的表示法不一樣，在一般資料庫中，要增加一筆病人記錄時，會有一個輸入資料及檢查是否合法的程式，相反的，在知識庫系統中，病人必須有一個明顯表達的生日日期，也可能以一階敘述邏輯 (First Ordered Predicate Logic) 的語言表達，像這樣的式子，電腦可以把它看成像數學運算式一樣：

$$\forall x (x. \in. \text{patients}) \rightarrow \exists y (x. \text{hasdateofbirth. } y)$$

上式可以讀成“對所有  $x$ ，如果  $x$  屬於病人集合，則存在一個  $y$ ，使得  $x$  的生日日期為  $y$ ”，在上式中，一些符號的意義如下：

$\forall$  表示“對所有的”(for all)

$\in$  表示“屬於”(member of set)

$\rightarrow$  表示“則”(implies)

$\exists$  表示“存在一個”(there exists)

一般資料庫系統和知識庫系統使用不一樣的方法，前者將生日日期的要求放在資料錄結構及程式中，而後者則只將這個需求明顯的表達在一個地方，而且，以後系統要用到這個需求時可以把它當做一種知識的形式儲存，而這個形式應該要很容易修改。

在這個例子中，這個明顯表達的規則以一種形式語言 (Formal Language) 稱為“一階敘述邏輯”(First Order Logic) 語言的方式表達，這裏“形式”的意思是(a)一種定義明確 (Well Defined) 的規則，用來建構合法的運算式，(b) 一種定義明確的規則，使得能夠由一些運算式推導出一些合法的運算式。

形式語言和表達知識的方式很重要，因為它們能讓不是寫那些知識的人了解真正的意思，並且能夠自動推理，得到新的知識，但是，到目前為止，形式

語言還存在一些問題：形式語言的表達能力有限，而且它們所用來做自動推論的方法並不是很有效率。因此，人們就發展出另外一些比較不是那麼形式化的知識表示法，我們在這本書會概略的介紹這些表示法。

你可能會問，為什麼我們不用自然語言，像英文、中文等來表示知識，建立一個以自然語言表達的知識庫系統？其中最主要的因素之一是：自然語言不是形式語言，因此就沒有我們上面所講的形式語言的優點，到目前為止，還沒有人能夠在任何一種自然語言中指出決定那些是合法句子的所有規則，如果有人能夠將決定句子是合法的所有規則都列出來，或許就能寫成電腦程式來了解自然語言，但是，即使如此，以規則庫來了解自然語言的方法是值得懷疑的，因為在自然語言中，一個句子的意思，往往是根據上下文而定，一個相同的句子在不同的地方可能會有不同的解釋，所以不太可能寫一些通用演算法來處理自然語言的句子。

另外一個不能用自然語言來當做知識庫系統中表達知識的方法之原因，是因為自然語言缺乏一致的表達方式，這個缺失會使得我們用電腦來處理這樣的句子時非常沒有效率，在資料庫技巧中已經發展出很多方法，而自動推論只有在一致的形式 (Uniformly Formatted) 及資料結構上才能適用。

## 1.3 知識庫系統

我們已經定義過，知識庫是由一些簡單事實，像 (John works for IBM) 以及一些通用規則，像 (all humans are either male or female) 等所組合而成。所謂的知識庫系統是一些資源 (Resources) 所組成的集合，這些資源包含硬體、軟體，可能還有人，這些資源的職責有：儲存知識庫、維護安全、維護一致性 (Integrity)、提供使用者能夠查詢的輸入 / 輸出程序，也包含了推理、攫取資料的能力，使得知識庫能依照個人的需要而獲取資料，如上所述，知識庫和一般資料庫不同的地方有四點：

- (a) 知識庫是典型的將規則很明顯的表達出來，像簡單表達簡單事實一樣的表達出來 (如上所述)。

- (b) 知識庫的儲存結構和資料庫結構比較起來是典型的低結構化語意內容 (Low Structure Semantic Content) (如上所述)。
- (c) 知識庫系統除了像一般資料庫系統一樣有語法檢查之外，還包含自動維護語意一致性的部份 (見後面說明)。
- (d) 知識庫系統包含了能夠在知識庫做推理 (Inference) 的部份，因此，提供了自動推理以攫取資料的能力 (見後面說明)。

KBSs (知識庫系統, Knowledge Base Systems 的縮寫) 和“專家系統”也不一樣，專家系統常常用來設計在某個特定的工作，只有專家才能解決的領域中，如診斷病情及找出數學定理證明中的錯誤等等。

KBSs 可以用來建構專家系統，但是它們的功能不只如此，也可以用來建構複雜的一般資料庫系統，也可以用來建構“特殊功能”的系統，如圖樣辨認等等。

KBSs 和 Nijssen (1984) 所定義的“第五代資料庫系統”的界線並不是很清楚，我們可以把第五代資料庫系統視為 KBSs 的一個特定型態，只是規則相對的減少，並且非常固定。第五代資料庫系統的概念架構表示式中反映出通用規則及使用這些規則的固定性。

一個不很複雜的知識庫系統的組織圖如圖 1.1 所示：

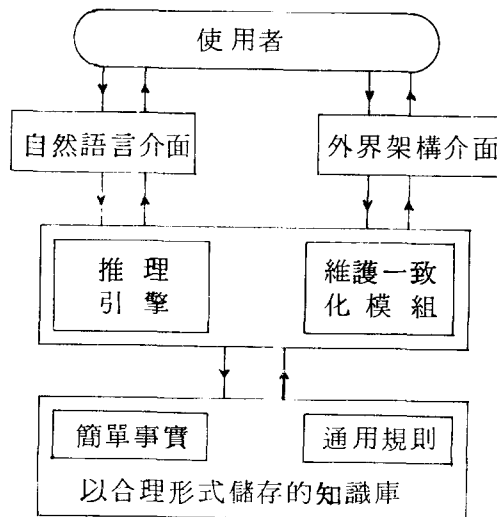


圖 1.1 一個不很複雜的知識庫系統組織圖

## 1.4 目前在KBSs的發展情況

目前有許多研究單位從事KBSs上不同領域的研究，例如：

- (a) 在硬體方面，已經發展出大型儲存含有低結構化語意的簡單事實。
- (b) 硬體上已發展出一些語言如LISP、PROLOG等的規則加速推理的能力。
- (c) 已經發展出利用形式邏輯為基礎的語言之規則表達，來做語意一致性的自動維護。
- (d) 已經發展出加速推理以攫取資料的方法，這個方法是將順序的 (Sorted) 一階敘述邏輯的定理證明技巧和在關係資料庫系統 (Relational Data Base System) 中所用的關係代數運算，如投影 (Projection) 等概念融合在一起。學者們也開始研究PROLOG的擴展，例如，新的順序邏輯已用在擴展的PROLOG上。
- (e) 已經有人研究利用邏輯來表達及推理一些含有不確定資料，相信程度及時間等的知識。
- (f) 以同樣標準合理的形式儲存知識而能允許不同的使用者以不同的角度來使用這些知識的方法也是大家研究的方向，也有人從事將自然語言轉為標準合理形式的研究。

還有其他不同的研究，本書會有一些介紹。

## 1.5 資料庫技巧、形式邏輯、專家系統及自然語言處理的概念及技巧的整合

知識庫系統設計的進步要歸功於將資料庫技巧，形式邏輯，專家系統及自然語言處理概念及技巧的整合，資料庫技巧可以讓不同的使用者以不同的觀點存取一致形式的簡單事實，形式邏輯允許表達更通用的指述 (Statement)，並且可以推論，自然語言可以提供讓我們表達一些複雜的事物，一連串發生的

事件及含糊 (Vague) 的知識，專家系統則提供處理不確定 (Uncertain) 知識的技巧。

知識的儲存及管理方法是依照知識表達方式而定，也就是依照我們對所討論的範圍的看法而定，在資料庫系統有四種最常用的看法：階級的 (Hierarchical)，網路的 (Network)，關係的 (Relational) 及二元關係的 (Binary Relational) 看法。在形式邏輯中，有一部份我們所討論的範圍視為像一階關係結構的數學結構。自然語言處理把所討論的範圍視為更複雜的結構，有一部份更可以利用更複雜的表示法，如架構 (Frame) 及劇本 (Script) 等來表示。

爲了要提供一個整合資料庫技巧、形式邏輯、專家系統及自然語言處理的基礎，我們要介紹一種對範圍內所有事物的看法，我們定義爲：

“一些概念及規則所組成的集合，這個集合可以用來將我們對這個範圍的看法連繫起來，並且公式化”。

像這樣的看法能夠幫助我們辨認、歸類及表達那些我們所感興趣的部份範圍的成份及處理它。定義明確的看法能夠將我們所討論的部份範圍以一致的方法表達，並且結果也能利用相同的看法而得到正確的解釋。

不同的看法可以用來爲不同的目的建構不同的表示法，例如，圖 1.2，如果我們所討論的範圍某部份用牛頓力學來分析，一個對應的看法爲將那部份視為包含一些事物，而這些事物的位置可以精確的定義出來，如果相同的部份以量子力學來分析，則對應的看法爲將包含那些事物的範圍視為事物在空間的位置是由機率分配所給定的。

在資料庫工作上，不同的觀點可以幫助設計者選擇對應的資料結構來儲存知識，例如，在某個觀點中須要區分出“實體”(Entities)、屬性 (Attributes) 及“關係”(Relationships)，實體可以用資料錄表示，關係可以用資料錄之間的存取路徑來表示，這種觀點非常接近資料庫中的資料模式，但是，資料模式主要注重在資料儲存及預定存取路徑，而觀點是注重我們對實際生活中各種事物的看法。

在形式邏輯中，將我們所討論的部份範圍視為由一些數學結構所組成，例如，在一階敘述邏輯中，我們所討論的部份範圍視為一階關係結構，每一個結

構包含一些實體及一階關係的集合以及定義在這個集合上的函數，在“一階關係”的觀點和資料庫中的關係資料庫的觀點差不多。

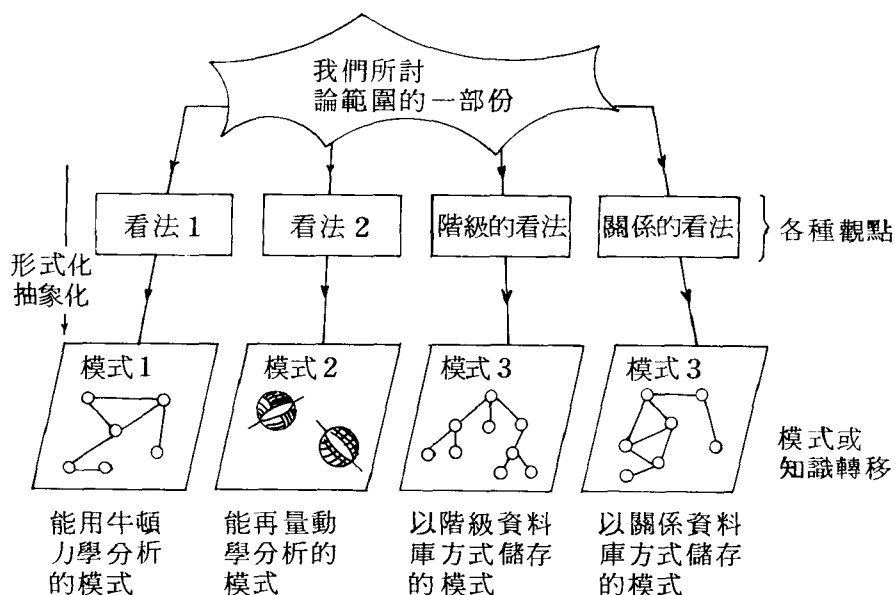


圖 1.2 對我們所討論的範圍有不同的看法

每一種觀點並不意味著有一種特定的資料結構來完成這個觀點。我們不能再犯下當初設計資料庫系統及資料庫管理系統相同的錯誤，許多資料庫的觀點是什麼大都以資料儲存方式來決定，我們希望以觀點為主，同一種觀點可以有不同的方法來儲存知識，不過，我們這樣說並不表示實作方面不重要。

我們要以那一種觀點來看我們所討論的範圍比較好呢？我們相信不能只以一種形式邏輯來看，要建立一個通用的知識庫系統必須要有各種不同的表達能力，一種觀點的好壞也可以從它是否容易轉成另外一種觀點來看，不同的使用者對我們所討論的範圍有不同的認知，我們最好能夠以和他們想法相近的方式來表示給他們看。

本章是學習知識庫系統的初步，所以，我們以介紹知識、知識庫表示法及對我們所討論的範圍之不同觀點來做為起點，在不同的觀點下，我們有不同形式的知識表示法，在第二章中，我們會概略介紹一些不同的知識表示法。