實質空間之宣告式語言描述方法

Declarative Languages for Describing Physical Space

doi:10.6154/JBP.1990.5.012

建築與城鄉研究學報, (5), 1990

Journal of Building and Planning, (5), 1990

作者/Author： 林峯田(Feng-Tyan Lin)

頁數/Page： 141-151

出版日期/Publication Date：1990/02

（國立臺灣大學建築與城鄉研究學報）
第五卷　第一期　民國七十九年二月　評介　第 141 頁～151 頁
JOURNAL OF BUILDING AND PLANNING NATIONAL TAIWAN UNIVERSITY
VOLUME 5, NUMBER 1, FEB. 1990, REVIEW AND CRITICISM, pp. 141～151

實質空間之宣告式語言描述方法

林峯田*

Declarative Languages for Describing Physical Space

by

Feng-Tyan Lin*

摘　要

近年來，宣告式語言的發展在計算機科學界受到相當的重視，它被認爲是電腦邁向人工智慧境界所必須具備的條件之一。但是，它在實質空間環境的規劃與設計應用領域，仍未受到應有的重視。從使用者的觀點來看，一種以描述問題結構爲取向的語言，毋寧是必要的，它可讓使用者免除於程式設計的細節工作。本文從描敍實質空間環境的應用需求角度，介紹了一些具代表性的宣告式語言。

ABSTRACT

In these years computer scientists have devoted a lot of efforts to the development of declarative languages which constitute an essential part in the field of artificial intelligence. However, only a few planners and physical designers recognize the potential usages of declarative languages. From the viewpoint of users, declarative languages provide an environment such that users can concentrate on the problems which they are interested without worrying about the details of programming design. This article introduces some declarative languages, such as many-sorted algebraic language, formal language, logic programming language (Prolog), and constraint logic programming language, by which the spatial objects can be described.

民國 78 年 12 月 4 日收稿

*國立台灣大學建築與城鄉研究所副教授

Manuscript received on December 4, 1989.

*Associate Professor, Graduate Institute of Building and Planning, National Taiwan University, Taipei, Taiwan, Republic of China.

關鍵字：宣告式語言（Declarative Language），
多型代數語言（Many-sorted Algebraic Language），
形式語言（Formal Language），
邏輯語言（Logic Programming Language），
限制式邏輯語言（Constraint Logic Programming Language）。

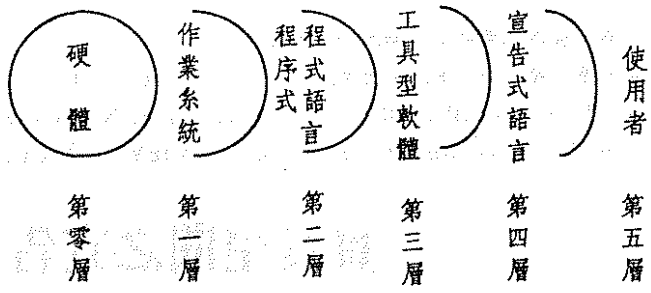

圖1　計算機應用系統層次

一、前　言

自從五〇年代以來，電腦科技的快速進步，幾乎對各行各業均造成了深遠的影響。尤其，它的影響並不僅止於技術層面而已，同時也深入到研究方法、基本理論及哲學理念等各個層次。在建築及都市計劃這個行業，雖然它的影響力尚十分微弱，大多在計量性的分析及設計圖的繪製等方面，但是已有人開始深思，這個行業的養成教育、規劃設計方法、決策過程及基本理論將會受到什麼影響，是否應該做些調整（註1）。面對這些問題，有的學術單位及民間設計顧問公司十分勇於嚐試，而有些單位則根本拒斥，而大部份的團體採取的是觀望或較保守的態度，僅作一些技術性的調適與追逐而已。

在能夠回答「電腦科技將會對建築及都市計劃的養成教育、規劃設計方法、決策過程及基本理論產生什麼影響」這個問題之前，我們有必要從計算機科學的理論層面這個角度來看此一問題。所謂「計算機理論」，涉及的範圍十分廣泛，包括：計算機結構、程式語言、軟體工程、資料結構、資料庫管理、演算法則、人工智能以及專家系統……等等。每一電腦理論的領域當然都會對在應用層面的使用者造成影響。所以，爲了使我們的討論明確起見，本文將針對電腦科學中的「語言理論」這個角度，來探討電腦如何與使用者（規劃師）溝通，以及規劃師又如何透過電腦來認知實質空間環境的問題。

「語言」是人類彼此溝通的主要媒介之一。雖然，它是一種有效的工具，卻不夠精確，常常出現文法錯誤或語意不清的情形。尤其，當我們要與電腦溝通，希望它能助我們一臂之力的時候，便需要一種十分精確的語言，俾使電腦能正確無誤地掌握到設計者的思惟。習慣上，這會令人直接地聯想到電腦的程式語言（computer programming language）。誠然，程式語言是電腦據以接受指令、執行運算的基本媒介；然而，從設計思路到程式寫作之間，畢竟存在著太大的差距！它需要一種中介性的、描敍性的語言來縮短此一溝通上的差距。底下，我們用圖1說明此一關係。

在圖1的最裡層（第零層）是計算機硬體，組成的成份是一些電子零件。它的外面一層（第一層）是作業系統（operating system），其本質是一些軟體程式的集合，用來協調、控制計算機主機與螢幕、印表機、磁碟機等週邊設備的運作。在個人電腦上，常見的作業系統有MS－DOS及PC－DOS；在工作站及大型電腦上，則有UNIX及VMS等。在作業系統外面的一層（第二層）是「程序性程式語言」（Procedural Programming Languages），包括了常見的BASIC, PASCAL, C等等。使用者（第五層）可以利用程序性程式語言（第二層）來撰寫電腦程式，以達到演算的目的。但是，通常這是一件煩人的事，使用者經常陷入瑣屑的程式寫作之中，而無法全心全力地貫注在其所關心的主要課題上面。於是乎，一些工具型軟體（第三層）被發展了出來，作爲使用者及程式語言之間的仲介。亦即使用者可以直接利用工具型軟體上的功能，達成其目的。常見的AUTOCAD, DBASE,LOTUS 1－2－3, PE－II等，均屬於此一層次。從語言的角度而言，第二以及第三層次的「程序型程式語言」和「工具型軟體」，均屬於一種計算機語言。使用者透過它們，控制計算機的運作，來達成演算的目的。但是，它們卻脫離不了低層次工具的色彩，就如同另一種新式的筆和紙一般，而與規劃設計的思惟過程無關。從圖一的架構來看，我們需要在第三層與第五層之間，加入一種新的語言，來精確地反映規劃設計者認知其研究對象及思惟推理的過程。具體地來說，第四層的語言應該是「宣告式」（declarative）的，它是一種描述問題特質的語言，而非像第二、三層的語言係以敍述解決問題之步驟爲導向。目前常見的Prolog語言，便屬於這一類。由於電子計算機的結構，本質上便是一種全然機械化的設備，使用者與它溝通時所運用的語言，必須要具備有（或能轉化爲）「可操作符號性語言」（operable symbolic language）的特質。此處所謂「可操作」，指的是它可以轉化成一系列的指令，用來控制計算機從事機械式的操作程序；至於「符號性語言」，不同於人類日常生活上所使用的語言，它是一種具有嚴謹語法（syntax）與語意（semantic）結構的語言（註2）。宣告式語言也必須滿足這些語法與語意上的條件。

這種需求並非今日始有。在哲學的論戰之中，「合理主義」便一直是一種強而有力的訴求。社會學家涂爾幹（Durkheim）在他著名的《社會學研究方法論》（The Rules of Sociological Method）一書中，便要求社會學家要向自然科學者學習，以嚴謹的態度，從事社會學領域的科學研究，而不要作形而上式的爭辯。結構主義學者李維・史特勞斯（Claude Levi-Strauss）更進一步地認爲：使社會科學達到像數學般地科學化，是一個值得追求的目標（李超宗，1989：236）。在建築學界，在三〇年代也掀起了在建築教育中拓展合理之設計程序的浪潮，希能平衡數世紀來被過度强調了的直感成分與偶然成分（Roe，1972：5）。當然，也有一些人對設計過程能否完全地機械化，提出强烈的質疑與批判。在介於全盤的肯定與否定之間，一般較爲穩健的態度則是認爲將電子計算機引進設計過程中，有助於把設計者從瑣碎的、制式化的工作中解放出來，讓設計者能更專心於具創造性的工作（李英明，1989：125－143）。

以下各節首先就實質空間的組成要素加以分析，然後以一些具代表性的宣告式語言爲例，介紹它們如何描述實質空間的方法。這些宣告式的語言包括了：多型代數語言（Many-sorted Algebraic Language），形式語言（Formal Language），邏輯語言（Logic Programming Language），限制式邏輯語言（Constraint Logic Programming Language）等。事實上，這些宣告式語言都還在繼續研究發展之中，進一步的應用性，也有待更深入的探討。

二、實質空間的組成要素

早在十多年前，密歇爾（William J. Mitchell）於他的《電腦輔助建築設計》（Computer-aided Architecture Design）一書中，便指出吾人需要一套符號性的描敍方法，來表達建築設計的理念。同時，他也指出了實質空間系統係由「元素」（element）、「屬性」（attributes）、「關係」（relations）及「階層」（hierarchy）等四個要素所組成（Mitchell, 1977）。符號語言的表達便是以這四個要素爲主要內容。

「元素」是實質空間系統的最小單位，相當於物理學中「原子」（Atom）的觀念。然而，實質空間的元素可以隨著應用目的的不同，由規劃設計師做適當的擇定。例如，一棟建築物的構成元素，在結構工程師的眼中，它們可能是各種的樑與柱；在空間配置規劃階段，它們則可能是各種不同的房間；在分析垂直動線系統時，設計師可能將每一層樓當做一個元素來處理。所以，實質空間的構成元素是可以隨著目的而有所變化的。

每一種元素都有其「屬性」。它們可以是幾何性的（如：長、寬、高），物理性的（如：重量、透明度），或者經濟性的（如：成本、收益）等等。例如，柱子有高度、截面積、承載力等屬性，傢俱有長、寬、高、重量、價格等屬性。如果稍加深思，我們便可以發覺，如欲完整不遺地描敍任何一種構成元素，其所需的屬性項目，幾乎是無限多的。除了部分的屬性，是可由某些基本屬性所導出來以外（例如，面積屬性可由長與寬二屬性求得），屬性的項目也必須依據目的之不同而有所選擇。

元素與元素之間存在著各種空間性和非空間性的「關係」。這些關係可以是命題的、邏輯的、或者是數值的。例如，設甲、乙、丙爲實質空間的三個構成元素，「甲和乙相鄰」是一種空間性的命題關係；「甲和乙相距m公尺」則是空間性的數值關係；「如甲比乙重，則甲比丙重」是物理性的邏輯關係。同樣地，元素與元素之間存著無限多種可能的關係；在某一特定的應用當中，關係的選擇必須視目的而定。

最後，吾人如依據元素的特性，把它們「階層化」的組織起來，將可簡化描敍事物的複雜度，並可確保各資料間的一致性。例如，吾人可視某一建築物係由數部門所組成，每一部門由數辦公室所組成，每一辦公室由數牆面所組成，每一牆面由數樑、柱、磚塊所組成。當吾人進行數地計劃時，可逕以該建築物爲處理單元；當進行動線分析時，則以各辦公室爲處理單元；而結構分析則是以樑、柱爲基本單元。由於這些不同的分析及設計過程中，其處理的單元之間有階層的從屬關係，可以使這些分析設計的成果，獲得某一程度的一致性。

密歇爾所提的上述觀念，至今仍然是一正確的方向。但是，當時的資料結構（data structure）、程式語言及人工智慧等電腦理論俱尚處於萌芽的階段，所以未能有進一步的發揮、闡釋。本文以下各節即是依據近十年來電子計算機語言學的進展成果，來重新檢視密歇爾的實質空間構成要素之觀念。

三、多型代數語言（Many-sorted Algebraic Language）

代數是一門很古老的數學，它兼具有嚴謹的語法和語意結構。所以，它是理論電子計算機學家心目中，最完美的程式語言候選人之一。雖然目前它仍未能達到實用的階段，但是它具有極大的發展潛力。從實質空間規劃應用的角度來看，它幾乎和上述 Mitchell 的四項要素完全相呼應，是故它是一種值得吾人深入探討的語言。

一個代數系統（algebraic system）是由元素的集合（A set of elements）、一組函數（A family of func-

tions）、及一組公理（A family of axioms）所組成。例如，「自然數系」及「加法」構成了一個代數系統（設稱之爲S），其元素的集合N＝ {0, 1, 2, ...} ，函數F即是加法本身，F＝{+}，公設A包含了x=x（反身律），x＋y＝y＋x（交換律），x＋0＝x（加法單位元素），．．．。進一步地，吾人將之表示爲S＝＜N，F，A＞；亦即，代數系統S是由N，F及A三個要素所組成。

值得吾人注意的是，在傳統的代數系統裡，其元素的類型（type）是均質的。舉例來說，整數系代數系統的元素，全都由整數組成，而不涉及無理數、實數等。但是，這種限制卻無法滿足電子計算機程式語言的要求。在程式語言發展的過程中，由於實際的需要，有了資料型態（data type）的設計。它的用途正如數學上「單位」的觀念一樣，在於避免數字的錯用，不同的資料型態也不能混淆亂用。於是乎，一個程式裡，常常需要運用到好幾種不同的資料型態。然而，這種實際的現象，並不能用傳統的代數系統來表達。計算機科學的理論學者爲解決此一問題，遂引進多元類型的觀念，把傳統的單一類型代數學擴充爲「多元代數」（Many-sorted Algebra）的理論。從而更進一步地發展出了「抽象資料型態」（Abstract Data Type）以及「事物導向程式設計」（Object-oriented Programming）的新觀念。

在多元代數的理論架構下，系統元素不僅包含了各種不同的屬性型態，也嚴格地規定了各種運算的定義域與值域。這些抽象的數學觀念又如何可以被應用到實質空間的描敍上面呢？我們試以建築物爲例說明之。

爲了方便起見，我們把「建築物」定義爲一組房間的集合（註3）。每一房間又分別有空間尺寸屬性（以數字表示，如長、寬、高等）以及機能屬性（以文字表示，如客廳、臥室、餐廳、廚房等）。此外，建築物的營建行爲，包括了「新建」、「增建」與「改建」等方式。在這裡，我們假設增建都是垂直方向的增建。從多型代數的理論來看，每一種營建方式相當於一種函數關係。茲以NEW,ADD, MODI三個函數來分別表示。底下我們先以程式設計的方式將建築物的多型代數系統加以敍述，然後再進一步地詳細說明。

```
DATA TYPES：
    ROOM=RECORD OF
        length, width, height： REAL；
        usage：STRING；
        END；
    BUILDING=SET OF ROOM；
FUNCTIONS：
    NEW：ROOM——>BUILDING；
    ADD：BUILDING×ROOM——>
    BUILDING；
    MODI：BUILDING——>BUILDING；
    BHGT：BUILDING——>REAL；
    a,b：ROOM；
VARIABLES：
    R      ：ROOM；
    S      ：BUILDING；
AXIOMS：
    BHGT（NEW（R））=R.height；
    BHGT（ADD（S,R））= BHGT（S）＋R.
    height；
    BHGT（S）＜＝20；
```

多型代數系統的宣告，共可分爲四個部份：型態宣告（DATA TYPES），函數宣告（FUNCTIONS），變數宣告（VARIABLES）及公設宣告（AXIOMS）。

本例中計宣告了二個資料型態：ROOM和BUILDING。"RECORD OF"及"END"是保留字（RESERVED WORD），夾於此二保留字之間的是資料型態ROOM的屬性項目；其中，length, width及height是空間屬性，其值均爲實數，以保留字REAL示之；usage是機能屬性，其值是一字串，以保留字STRING示之。接著，我們由此基本資料型態定義出了另一個資料型態BUILDING。保留字"SET OF"指出了BUILDING是一組ROOM的集合。

在宣告了資料型態之後，我們緊接著宣告了六個函數的名稱及其定義域與值域。值得再强調一次的是，每一營建活動可用一函數來描敍之。函數NEW表示（新建的）房間本身便是一棟建築物。函數ADD表示一棟建築物再增建一間房間，結果仍然是一棟建築物。函數MODI表示一棟建築物改建之後，還是一棟建築物。在實務上，如果某些地區可能禁止新建或增建，只能改建。那麼，在其多型代數系統中，便不許有函數NEW及ADD的存在，而只能有MODI了。除了這三個定義營建活動的函數之外，函數BHGT定義了建築物的高度，至於高度值應如何推算，則另於公設宣告時，再予界定。最後，函數a與b是二個「常元函數」，他們代表了二種基本的房間型式。（當然，如果系統裡有n個房間基本型，我們便以n個常元函數來表示之。

在繼續介紹變數及公設的宣告之前，我們有必要先檢視一下由前述的函數所定義出來的「建築物」。在此一多型代數系統裡，最基本的建築物是NEW（a）及NEW（b），分別表示由房間型式a及b所構成的單一房間建築物。接著，雙房建物共有四種型式，分別是ADD（NEW（a），a），ADD（NEW（a），b），ADD（NEW

（b），a），ADD（NEW（b），b），它們各自代表了一、二樓是 aa, ab, ba, bb 等型式的建築物。同理，ADD（ADD（NEW（a），b），a）是三層建築物，它的一至三樓房間型式分別是 a, b 及 a。

多型代數系統的第三部份是變數的宣告。變數宣告是爲了方便公設宣告時代數式的表達，而定義了一些變數。在本例之中，我們使用了二個變數 R 和 B，它們的資料型態分別是 ROOM 和 BUILDING。是故，變數 R 可以是房間型式 a 或 b，而變數 S 則可以代表任何一種由 a 及 b 構建而成的建築物。

最後，公設宣告的部份包括了三條公設。第一條公設說明了新蓋的單房建築物 ADD（R）的高度〔即BHGT（ADD（R））〕等於該房間 R 的高度 R.height（注意，我們用 A.x 代表 A 事物之 x 屬性值）。第二公設說明了增建的建物 ADD（S,R）的高度是原建物 S 的高度 BHGT（S）加上新增頂樓房間 R 的高度 R.height。第三公設則敍述了建築物高度不得超過 20 單位長度的高度限制。

至此，我們完成了建築物多型代數系變的宣告工作。然而，必須再次强調的是，在上述的宣告中，吾人僅就房間的空間與機能屬性及建築物如何由房間所組成等性質加以定義而已，而尚未眞正就實際尺寸及使用機能加以明確敍述。一旦吾人進一步地敍明其空間大小及使用機能，它便代表了某一特定的房間或建築物了。是故，吾人尚需一實體敍述（STATEMENTS）的步驟。舉例如下：

```
STATEMENTS：
        a.length＝10；
        a.width＝8；
        a.height＝4；
        a.usage＝“Office”；
        b.length＝10；
        b.width＝8；
        b.height＝5；
        b.usage＝“Store”；
```

上面的實體敍述界定了房間型式 a 的長、寬、高各是 10, 8, 4，其使用機能是辦公室；而房間型式 b 的長、寬、高各是 10, 8, 5，其使用功能是商店。

根據上述的多型代數系統及實體敍述，我們可以利用公設一及二來計算建築物的高度。設若某一五層樓的建築物是 ADD（ADD（ADD（ADD（NEW（b），b），a），a），a）。其高度之計算如下：

BHGT（ADD（ADD（ADD（ADD（NEW（b），b），a），a），a））
＝BHGT（ADD（ADD（ADD（NEW（b），b），a），a））＋a.height
＝BHGT（ADD（ADD（ADD（NEW（b），b），a），a））＋4
＝BHGT（ADD（ADD（NEW（b），b），a））＋a.height＋4
＝BHGT（ADD（ADD（NEW（b），b），a）＋8
＝BHGT（ADD（NEW（b），b））＋a.height＋8
＝BHGT（ADD（NEW（b），b））＋12
＝BHGT（NEW（b））＋b.height＋12
＝BHGT（NEW（b））＋17
＝b.height＋17
＝22.

由於公設三規定了建築物的高度限制爲 20，故知該五樓建築物不可以存在於本系統中。

從上面的例子裡，我們介紹了多型代數系統描敍實質空間環境的方法。事實上，它十分嚴謹地界定了一個系統的活動模式；然而，由於它剛起步不久，在實際的應用上仍然有許多問題尚待探索、解決。例如，對於任一組公設之間是否有矛盾存在的問題，數學家已經証明吾人並無一演算法則（Algorithm）可資用來檢定（註 4）；此外，如何發展出一套迅速、有效的方法來自動地求解（如上例中求建築物高度的問題），也是一個值得努力的課題。如果這些瓶頸能夠突破，將可以把人工智能的應用提昇到一個更高的境地。

四、形式語言(Formal Language)

關於語言的結構，很早便引起了語言學家的興趣。在五〇年代末期，强史基（Chomsky）開始以數學的方法，展開了對語言結構的研究。從此，語言學家們有了一項新的工具，可以把語言共通的基本要素，用符號來代表，而要素之間的關係，則用數學形式來表達。此一研究的領域便被稱之爲「形式語言學」（Aho, 1979）。由於形式語言具有嚴謹的數學結構，是一種適合機械式操作的語言，所以計算機程式語言便借用它爲語法及語意分析的主要工具。此外，它也被應用於影像處理（image processing）及空間組合及判定（spatial synthesis and verification）等領域。本節將就形式語言的基本理論及在空間設計上的應用，做一個簡單的介紹。

語言的語法結構可以透過「文法」（grammar）或者「剖析器」（parser）來加以表達。這兩種方法剛好是一體之兩面，文法是用來產生「句子」（Sentence），而剖析器則用來分析句子，判定它是否合乎既定的語法結構。底下分別介紹之。

一套文法係由四個要素 Σ，N，P，S 所組成，其中：

(1) Σ是一組由「字彙」所構成的有限集合（Finite Set）。舉例來說，'橘子'、'蘋果'、'吃'、'美麗'等，皆是字彙。

(2) N是一組由「字彙變數」所構成的有限集合。'形容詞'、'名詞'、'動詞'等皆是詞彙變數，因爲它們代表了一組相對應的字彙。

(3) P是一組由「規則」所構成的有限集合。規則的形式爲 α ——> β，其中 α 和 β 是字彙及字彙變數的混合串列，但 α 之中必須至少存在一字彙變數。舉例來說，下面的二條語法規則分別說明了'句子'可以由'主詞'及'動詞'所組成，也可以由'時間副詞'，'動詞'及'空間副詞'所組成。

'句子'——>'主詞' '動詞'

'句子'——>'時間副詞' '動詞' '空間副詞'

接著，以下的六條規則把字彙變數轉換成字彙。

'主詞'——>我

'主詞'——>你們

'動詞'——>走

'動詞'——>拜訪

'時間副詞'——>明天

'空間副詞'——>這裡

(4) S是「起始字彙變數」，它單獨出現在語法規則的左方。由S開始，根據P的規則，吾人便可產生一合乎此文法定義之語句。根據上例，如果S是'動詞'，那麼便可產生'走'及'拜訪'兩個語句。如果S是'句子'，則可以產生以下的一些句子：

我走

你們走

明天拜訪這裡

.....

形式語言也可以用來描敍實質空間結構。在1976年，Stiny應用形式語言的理論於實體形態上，定義了「形態文法」（shape grammar），並舉了一例，以梯形爲基本單元（字彙），根據組合規則，產生出各式各樣圖形（語句）的變化（Mitchell, 1977）。近年來，這一方面的研究仍然不斷。Radford應用形態文法於定義建築物之細部設計上，並企圖發展出一套專家系統（expert system）以組合出各種可能的建築設計（Radford, 1985）。由於形式語言與人工智慧語言，在理論上有其共通之處，是故它們的進一步發展，是可以預期的。

一句合乎文法的語句可能是含混的（Ambiguous），也可能是毫無意義的（Semantically meaningless）。尤其後者涉及語意學的範疇，雖然在純形式語言學裡已有一些初步成果，但仍有待更深入的探討。由於相關的內容已超出本文範圍，不再細述，此處僅舉一簡單的例子加以說明（註5）。

屬性文法（Attribute Grammar）是形式文法的一種延伸，它是在語法規則中加上了屬性資料，用以查核語意的適當性。圖2的塔狀結構物分爲「物體」及「支架」兩個部份，並且可用以下的「屬性化規則」表示之。

'塔（w,s）'——>'物體（w）' '支架（s）'；w<s.

'物體（w）'——>鐵球（w=8）

'支架（s）'——>紙架（s=2）

'支架（s）'——>鋼架（s=10）

在上述的規則中，每一個字彙或字彙變數均附有一個或一個以上的屬性，w代表物體重量，s代表支架的支撐力。根據這些規則，吾人可以導出'鐵球在紙架上的塔'之形態語句。圖3的語法及屬性剖析圖，查核出了此一結構事實上不可能存在（即犯了語意上的錯誤）。字彙變數'物體'的重量屬性w可以由字彙'鐵球'的屬性 =8 得知（圖中灰色路徑）；字彙變數'支架'的支撐力屬性s則可由字彙'紙架'屬性s=2得知。這兩個屬性值分別傳至字彙變數'塔'，由於他們不符合w<s的條件，故知其不可能存在。同理，語句'鐵球在鋼架上的塔'則因符合該條件，而可以實際存在。

形式語言提供了實質空間設計在語法與語意表達的方式上一個十分嚴謹的方法。然而，它有一項限制，使用者必須事先完全瞭解空間結構的語法規則。這項限制使得目前形式語言在設計初期，整體意念尚在構思的階段，很難被有效地派上用場。但是，這並不代表著絕望。計算機科學家目前所致力研究的「人類自然語言」（natural language）也有許多類似的問題本質存在。這些基礎的研究將來或許可以爲其它的應用領域打開出一條更寬廣的道路來。

五、邏輯語言——PROLOG

在二千多年前，亞里斯多德便開始講述邏輯學。由於一般人均相信，邏輯中一切重要的發現早已爲亞里斯多德所完成，同時，在十七世紀以前，各種數理科學也尚未發達，造成了古典邏輯學的一直停滯不前。

邏輯從古典的形式發展爲現代的符號形式，主要是靠著數學家的努力，萊布尼茲（Leibniz, 1646－1716）、布爾（Boole, 1815－1864）、摩根（Morgan, 1806－1871）、耶芳斯（Jevons, 1835－1882）及斐士（Peirce, 1839－1914）等人是其中主要的代表。到了1910年的時候，羅素（Russell）和懷海德（Whitehead）合作，苦心地創立了符號邏輯（Symbolic Logic）這一門學問（註6）。

計算機科學家把符號邏輯進一步地轉化成一種可以自動操作、運算的語言形式。由於各種事物及其關係均以符號與邏輯關係來表示，其蘊含關係的推導，也正如數學定

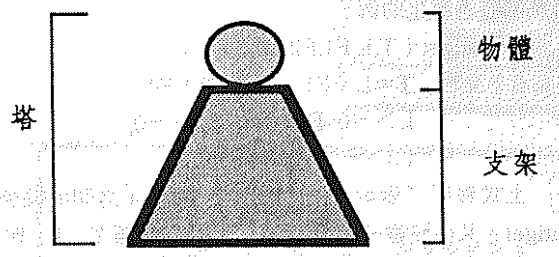

圖 2　語法結構

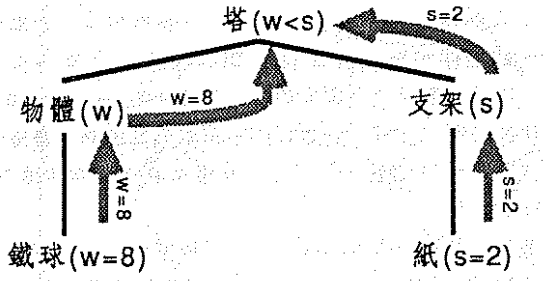

圖 3　語意解析圖

理之証明過程一般，這個領域便被稱之爲「定理証明學」（Theorem Proving）（Cheng, 1973）。

1981 年，日本政府宣佈了「第五代電腦計劃」，預備在往後的十年內，由政府編列預算四億五千萬美元，加上民間產業界一倍以上的配合款，全力推動人工智慧的研究。它們的目標是在 1990 年代發展出新一代的智慧型電腦，可以用一般語言與人類交談，有視覺及聽覺的能力，並且具有學習、聯想、推理、做決定等人類智能範圍內的行爲方式（Feigenbaum，1983）。這一項計劃的提出，震驚了全世界，尤其是執計算機科學牛耳的美國，第五代電腦計劃中所選定採用的程式語言——PROLOG 更引起了一陣熱烈的研究風潮。其實，PROLOG 可視爲定理証明學裡一項特例的實際應用，它讓電腦能根據定理証明學裡的一些理論來証明某一假設的正確性。雖然，它並未完全發揮定理証明學裡的理論功能，但是它的確是目前最有效、最被廣爲採用的邏輯程式語言之一。

PROLOG 程式語言主要由三個部份所組成：「事實」、「規則」和「假設」。其主要之目的即在依據已知之事實，以及事物關係之規則，來証明其假設。

所謂「事實」，指的是事物的性質或多項事物間的關係。敍述事實的形式爲

relation（object1, object2, ..., objectN）

當事物（object）只有一項，其關係項（relation）即代表了它自身的特性。例如：

father（john, mary）.

girl（mary）.

分別敍述了 John 是 Mary 的父親以及 Mary 是個女孩的事實。必須注意的是，PROLOG 規定事物的名稱一律應以小寫字母開頭，而變數（variables）則以大寫字母開頭的名稱來代表，所以在上面的兩項事實敍述裡，John 和 Mary 的名字均以小寫開頭。

在 PROLOG 裡，一組事實敍述的集合，亦稱之爲「資料庫」（data base）。在學理上，它也確與關聯性資料庫（relational data base）系統的結構相當。最近，有一些學者把 PROLOG 和關聯性資料庫系統加以進一步地結合，並稱該領域爲「邏輯式資料庫系統」。

所謂「規則」，即是事實之間的因果關係。其形式爲

A：－ B, C,…N.

其中的 A, B, C,…, N 各代表了一個事實陳述。此一規則的意義可被理解爲'如果 B, C,…, N 等事實存在時，事實 A 也存在'。

最後，「假說」的形式是 ？－ A, B,…, N. 它的意思可以被理解爲'事實 A, B,…, N 是否同時爲眞？'

由於下節將介紹的「限制式邏輯語言」，在語言的形式上中可視爲是 Prolog 的衍伸；是故，在此處，我們不特別舉例子，讀者可以參考下一節的實例（註 7）。

如前所述，Prolog 雖是一種廣被採用的邏輯程式語言，但是它並非是唯一的一種選擇（註 8）。在諸多的努力之中，嚐試排除 Prolog 現有的一些限制，是一個當今研究工作的大方向。下一節所將介紹的，是其中的一支。

六、限制式邏輯語言(Constraint Logic Programming Languages)

在計算機的發展過程之中，「數理計算」與「邏輯推理」一直是被當做兩個不同的研究領域（註 9）。雖然這幾乎是發展過程中不得不然的事實，但是卻與外界的眞實世界不符。在人類追求合理化的目標下，數理計算和邏輯推理一直是交互運用，相輔相成的。所以，近幾年來，計算機科學家也開始致力於整合這兩個被割裂的領域。其中有一些人所嚐試的方法是將 Prolog 語言的處理能力予以加強，把一些數學限制式（constraint）納入 Prolog 之中，並稱此一類型的語言爲「限制式邏輯語言」。

數學限制式在眞實世界的意義，反映了各種生產、消

費資源的有限性。從整合數理計算及邏輯推理的角度來觀察，把一般數學式及統計方法也一併與邏輯語言相結合，似乎是一條免不了的路子，尤其是在社會科學的研究當中，統計分析、數理模型、法則歸納、演繹推理等俱是不可或缺的工具（註 10）。然而，當吾人企圖將「迴歸分析」（Regression Analysis）及「最佳化求解」（Optimization Problem）與邏輯語言相結合時，程式語言的語意問題便有了很深刻的變化（註 11）。在此，我們並不做計算機科學上學理的探討，而是就其應用面來介紹它如何描敍一個影響實質空間之住宅市場的運作系統。

住宅市場是一個相當複雜的系統。由然本文係針對描敍該系統之語言做一介紹，是故對系統結構有所簡化，唯仍以能反映出系統的複雜性爲要。在本例中，我們除了納入了住宅市場的需求面及供給面等經濟模型之考慮外，也將政府的公共設施投資政策、獎勵投資融資政策、均衡地區發展等因素納入系統之中，加以描敍。

首先，我們假設研究範圍共涵蓋了四個地區，分別稱之爲 area1, area2, area3, area4。這四個地區各自有其地區性之住宅市場，且彼此之間有市場互動的關係。在政府部門方面，主要有兩項政策工具可供運用。其一是政府在各地區的公共設施投資建設，此一因素的影響範圍僅限於當地之住宅需求面，亦即會吸引更多的人到該區購置住宅。另一種政府的政策工具是以獎勵興建住宅，提供營建融資爲手段，此一因素會同時影響到四個地區的住宅供給面。此外，由於政府的預算是一定的，所以預算之分配於各地區之公共設施投資及整體貸款融資的比例，將影響住宅市場供需面的結構。

除了經濟層面的考慮外，政府還有一些社會目標必須達成。這些目標至少包括了以下數項：

1. 提高房地產總值：此一目標之追求有助於增加稅源之豐裕，使政府有進一步從事教育建設、社會福利工作之可能。
2. 投資之回收：公共設施建設之投資，應能從房屋稅中回收，俾再轉投資於其它地區，達成社會之公平性。
3. 控制房地產於一合理價格範圍之內。
4. 均衡地區人口分佈：人口過度集中，不僅造成公共投資之浪費，也增加因交通擁塞等而造成的社會成本。

這些目標之間，有些是協調一致，有些則是相互矛盾、牴觸的。

圖 4 表示了此一住宅市場的系統結構。簡單地來說，政府的預算可分爲五部份：興建融資貸款及四個地區的公共投資金額；融資額度同時影響了四個地區的住宅供給面，四個地區的公共投資則分別影響當地之住宅需求面。由供需均衡點所決定的均衡價格及住宅單位，分別構成了合理價位、人口分佈、房地產總值的指標因子。這些結構性的關係可以用限制性邏輯語言來描敍（註 12），底下舉幾個經過簡化的例子。

```
budget ( T,L,F1,F2,F3,F4 ) : -
        T=L+F1+F2+F3+F4,
        L>=0, F1>=0, F2>=0,
        F3>=0, F4>=0.
```

上式敍明了變數 T, L, F1, F2, F3, F4 之間的關係爲 budget，其中變數 T 是總預算，L 是貸款額度，F1 到 F4 分別代表四個區的公共設施投資。符號：－右邊則是這些變數之間必須滿足的條件（註 13）。

```
equi ( A, L, F, P, H ) : -
        formula ( D, S, L, F, PD, PS, P, H ),
        P=PD,
        P=PS.
```

上式敍述了地區 A 的融資 L、公共投資 F、均衡價格 P、均衡住宅單位數量 H 之間的關係 equi 應符合符號：—右邊的關係式及限制條件。關係式 formula 代表了該區供給及需求曲線，此處我們不再明列其細節。最後的二條式子 P=PD及P=PS表示均衡價格 P 是供需曲線 PS 及 PD 的交點。

最後，問題或假說的提出，可以根據前述規則關係之定義來陳述。例如，area1 地區設無任何融資及公共設施之投資，欲知其均衡價格 P 、均衡住宅單位 H 及人口分佈指數 E 時，其問句之形式如下：

```
? - equi ( area1, 0, 0, P, H ),
    popu ( area1, 0, P, E ).
```

規劃者只要用此簡明扼要之問句，計算機便可自動求出其 P, H, E 值。是故，用這種語言來描敍問題有下列潛在的優點：㈠ 程式的撰寫符合人類的邏輯思考，規劃者可以全心全力把問題描敍清楚，不必像撰寫 FORTRAN 程式，有太多程式設計的細節必須分心。㈡ 規則之間相互獨立，故其增加、刪減、修改，均十分容易。㈢ 規劃者可提出關鍵性的限制條件，而由電腦依據邏輯規則，將相關的細節條件一一找出，不但可以減輕規劃師的工作負擔，也可使問題的細部不致遺漏。㈣ 由於問題（或假說）的陳述與系統規則的描述是分開的，規劃者很容易提出各種問題或假說來加以驗証，不必修改程式本身。

雖然限制式邏輯語言有上述的這些潛在優點，但是我們必須強調的一點是，它畢竟尚在理論形成的階段，它的實用性還有待進一步的驗証。

七、結　論

以上各節，我們就多型代數語言、形式語言、邏輯語言、及限制式邏輯語言如何用來描敍實質空間的方法，分

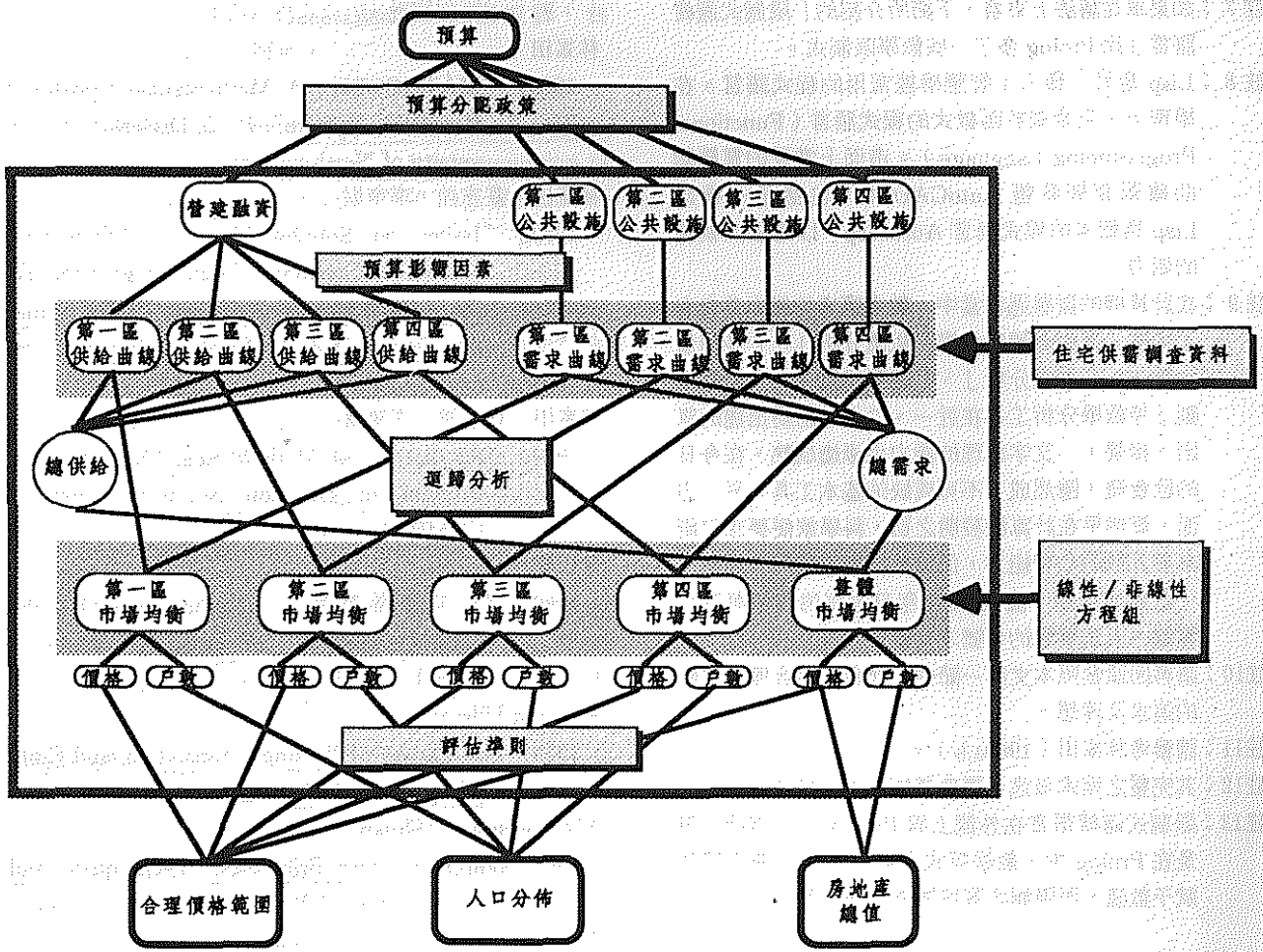

圖 4　住宅市場結構

別做了介紹。這些語言理論在計算機科學的領域裡，都有了一些或多或少的發展，但是在描敍實質空間的應用上，仍然十分少見，本文嚐試性地舉了一些實例，其進一步的實用性值得吾人再深入探討。此外，可用於描敍實質空間環境的語言，並不僅限於本文所介紹的這一些而已。在諸多的語言中，那一種語言最適合於那一種應用的場合，也是值得繼續研究的地方。

註釋：

註 1 ：參見 Eastman（ 1989 ），Ohsuga（ 1989 ），王佑仁（ 1989a,b ）等文的討論。

註 2 ：嚴格地來說，由於語意的理論仍然沒完全成熟，目前大部份計算機程式語言的語意大多不夠精確。用一般的慣用語來說，以這些語言寫的程式，具有先天性的陷阱（ 也稱之爲「 虫 」）。一旦執行程式時掉入陷阱，便造成錯誤的結果。

註 3 ：Mitchell 曾經提到過，在電腦輔助建築設計的領域裡，至少有五種主要描敍建築物的方法。分別是：規則網格法（ Regular Grids ），可變尺寸網格法（ Variably-dimensioned Grids ），多邊形法（ Polygon Representation ），對偶圖法（ Dual Graph Representation ），史密斯圖形法（ Smith Diagram Representation ）。

註 4 ：這類的問題有一種特性：如果公設之間是存在矛盾的（ inconsistent ），那麼，此一矛盾是「 可能 」被檢驗出來的；反之，如果它們是具一致性的，卻可能永遠無法被証明。

註 5 ：進一步之學理討論，可參見 Aho（ 1979, 1985 ）。

註 6 ：參見吳定遠（ 1981：1－6 ）。

註 7 ：如果單從語法上來看，下節所介紹的「限制式邏輯語言」比 Prolog 多了一些數學限制式。

註 8 ：Lisp 是另一種人工智慧學裡常用的程式語言。在學理上，它是屬於函數式的程式語言（Functional Programming Language）。市面上常見的電腦輔助繪圖套裝軟體 AutoCAD 中，包括了一種以 Lisp 爲藍本的程式語言 AutoLisp，以增強其軟體的能力。

註 9 ：在計算機的發展過程當中，數字運算的能力首先有了突破，這也是就爲什麼「數值分析」、「多變量分析」、「系統分析」、「線型規劃」、「動態規劃」等數學分析工具在近一、二十年紛紛出籠的原因。接著， 文字處理的能力也漸趨成熟，在今日的社會裡，隱然成了不可或缺的基本工具。另一方面，雖然早在計算機發明之初，科學家便夢想它能具有人類的心智能力，但由於技術上的尚未成熟，一直無法突破，直到 1981 年日本宣佈發展第五代電腦，人工智慧的口號才又被炒熱。

註10：請再回頭參照本文第一節中，社會家對合理化運動的追求及理想。

註11：請參考林峯田（1989a,b）。

註12：其完整之程式敍述，請參考林峯田（1989c）。

註13：限制式邏輯語言在外觀上與 Prolog 十分相近，但是在 Prolog 中，數學等式之右方在執行前均需被賦予數值，而限制式邏輯語言則無此要求。

參考文獻

王佑仁　譯

1989a 《電腦運用與建築教育》（How the Schools Are Teaching the Uses of Computers）20－21, 台北：空間雜誌社。（原作刊載於 Architecture, 1989 年 8 月號。）

王佑仁

1989b 《爲什麼不？「電腦運用與建築教育」譯後記》22－23，台北：空間雜誌社。

李英明

1989 《科學社會學》社會學新境界叢書 3：125－143，台北：桂冠圖書公司。

李超宗

1989 《新馬克思主義思潮——評介「西方馬克思主義」》桂冠政治學叢書 12：236，台北：桂冠圖書公司。

吳定遠

1981 《符號邏輯入門》水牛大學叢書 37，台北：水牛出版社。

林峯田

1989a "MATHCORE：A Mathematical Constraint Resolution System", Ph.D. Dissertation University of Northwestern.

林峯田，莊志洋，李德財

1989b "Issues on Solving Many－mathematical Models in Constraint Logic programming", the Annual Conference of the Rocky Mountain Society for Artificial Intelligence, Denver, U.S.A.

林峯田，莊志洋，李德財

1989c "Application of Mathematical Constraint Resolution to Decision Support System", COMPSAC.

曾仁文

1986 《人工智慧語言——Turbo PROLOG, the natural language of artificial intelli－gence》，台北：松崗。

Aho and Ullman

1979 The Theory of Parsing, Translation, and Compiling, Vol. I：Parsing.

Aho, Sethi and Ullman

1985 Compilers —— Principles, Techniques and Tools, Addison-Wesley Pub. Co.

Aiba, etc.

1988 "Constraint Logic Programming Language CAL", Proc. of the Intern. Conf. on 5th Generation Computer Systems, Tokyo.

Cheng and Lee

1973 Symbolic Logic and Mechanical Theorem Proving, Academic Press.

Clocksin and Mellish

1984 Programming in Prolog, 2nd edition, New York：Springer－Verlag.

Durkheim（涂爾幹）

1989 《社會學研究方法論》（The Rules of Soci－ological Method），黃丘隆譯，台北：結構羣出版。

Eastman

1989 "Architectural CAD：a Ten Year Assessment of the State of the Art", CAD, 21（5）：289－292.

Feigenbaum and McCorduck

1983 《第五代——日本第五代電腦對世界的衝擊》

（The Fifth Generation），尉騰蛟譯，台北：長河出版社。

Goguen, J. A., etc.

1976 An Initial Algebra Approach to the Specification, Correctness, and Imple mentation of Abstract Data Types, IBM T. J. Watson Reserach Center Report.

Hoare, C. A. R.

1987 "An Overview of Some Formal Methods for Program Design", Computer, Sept. 1987：85－91.

Horowitz, Ellis and Sahni, Sartaj

1976 "Algrbraic Representation of Data Struc－ture", in Fundamentals of Data Struc tures , Computer Science Press, Inc.

Mancarella, P., etc.

1988 "An Algebra of Logic Programs", in Logic Programming Proc. of the 5th Intern. Conf. and Symp., ed. R. A. Kowalski and K. Bowen, MIT press.

McLoughlin

1972 《都市及區域之系統規劃原理》（Urban & Regional Planning —— A System Approach），倪世槐譯，台北：幼獅書店。

Mitchell, William J.

1977 Computer－aided Architectural Design；台北：虹橋書局。

Ohsuga

1989 "Toward Intelligent CAD Systems", CAD, 21（5）：315－337.

Radford and Gero

1985 "Towards Generative Expert Systems for Architectural Detailing", CAD, 17（9）：428－435.

Roe, Soulis and Handa

1972 《合理的設計原則》（The Discipline of Design），漢寶德譯，台中：境與象出版社。

Wos, etc.

1984 Automated Reasoning —— Introduction and Applications, Argonne National Lab., Argonne, Illinois, U.S.A.