LISP基礎

LISPcraft by Robert Wilensky

南 石 譯

儒林圖書公司 印行

序

1. LISP程式語言

LISP 是目前還在使用的少數古老的程式語言之一，約翰・麥卡錫（John McCarthy）和他的學生在一九五〇年代末期首度發表了這種語言。LISP 這個名字，是取 LISt Processing（串列處理）的頭幾個英文字母組成的，以表達隱含於大部份 LISP 程式和 LISP 資料之下的基本結構：串列（List）。LISP 語言過去及現在的主要用途仍然在於人工智慧領域。

在這個計算領域快速變遷的時代，LISP 是少數能夠幸存下來的古老程式語言之一。從它首度發表及實作成功至今，LISP 語言經歷了無數次的演化，增加了許多新的性能，其實作方法也有了長足的進步，和使用者之間的介面也做了明顯的改善。這種語言已經建立起屬於自己的程式設計環境。此外，LISP 更藉著硬體的協助，以具有高度工作效率的 LISP 機器的形式出現。

這些改良通常都是爲了改善早期版本在觀念上或實作上所呈現的缺點而進行的。到了今天，LISP 某些令人難以接受的部份可說已不存在。但是，LISP 過去殘留在人們腦海中根深蒂固的印象常使得人們仍然認爲它是種有趣但却不切實際的語言，不過由於近年來電腦科技的快速進展以及日趨完善的軟體發展環境，使得 LISP 的這種形象已大爲改觀。許多 LISP 團體，已

如雨後春筍般大量的出現。

要探討 LISP，首先面臨的困難就是無法針對 LISP 語言究竟是什麼做個明確的定義。和別種語言不同的是，標準的 LISP 語言並不存在。LISP 語言擁有許多不同的發展方向，這導致 LISP 有個龐大的家族。LISP 語言的這些不同版本雖然具有相同的基本性質，但在某些方面又完全不同。有種 LISP 版本，稱爲 Common LISP，嘗試著替 LISP 程式設定更具統一性的標準，但是目前仍無一種 LISP 的解譯方式得到所有 LISP 團體的支持而公認爲是一種標準。

目前最常用的 LISP 版本是 MacLISP，它是在 MIT 發展出來的，還有就是 INTERLISP，由 Bolt, Beranek and Newman and Xerox Palo Alto研究中心所發展。這些版本已廣泛應用於特殊用途的 LISP 機器上。Portable Standard LISP 是另一種 LISP 版本，其可輕易的在不同的機器上做轉換。至於 Scheme LISP 版本則由 MIT 的 Sussman 和 Abelson 所推出，其對於許多基本概念的解譯方式和時下大多數的 LISP 語言版本截然不同。

2.範圍及目標

本書所用的 LISP 語言是 Franz LISP。Franz 是種外觀上近似於 MacLISP 的 LISP 版本，由加州大學柏克萊分校所發展，可在柏克萊 UNIX* 作業系統上使用。當然，本書儘量超然於各種不同版本之外，爲達這個目標，除了在各種版本的差異點上指出其它版本的不同用法外，並加上超然於版本特色（意即，較一般性）的註解。此處所引用的哲學基礎是：與其從頭學習一種新的版本，不如透過其它的版本比較學習來進行。

本書編排的目標在於希望讀者能對 Franz LISP 的工作方式具有較完整的認識，因此不對語言本身做任何的形式定義，而完全採取實用主義的觀點。除了對 LISP 的基本特性做個說明以外，程式設計的慣常用法、各種程式設計風格的基本元素，以及系統的介面等論題也佔有同等的份量。根據作者

* UNIX 是貝爾實驗室的註册商標。

個人的經驗，這些論題對於程式設計語言的影響不下於這語言的基本性能。

探討程式設計風格時已先進行過濾工作，以便提供某些有用的程式語言設計風格給讀者參考。對於好的和壞的程式設計風格也都加上完善的註解。這些註解完全是根據於對 LISP 程式設計實務所做的觀察，而非根源於程式風格上的偏好。

LISP 在人工智慧領域內的應用範圍很廣，但是這不是本書所要探討的重點，此處著重的是語言本身，而不是它的應用。等到熟悉這種語言的基本特性之後，讀者可以查閱其它的教科書，以進一步了解何謂人工智慧。

雖然本書的內容具有自我擴充的特質，但是仍然無法完全描述 Franz LISP 的每一部份。讀者可以查閱 J.K. Foderaro 和 K.L. Sklower 所著的 The Franz LISP Manual一書，以瞭解所有的細節。那本手册（Manual）著重於說明 Franz LISP 能做什麼，而非 LISP 解譯程式是什麼。

3.格　式

用電腦語言來設計程式就和從事其它的工作一般，需要特殊的技巧：只擁有專業的知識而無實務經驗也是枉然。即使是最好的程式設計語言教科書，也無法讓讀者熟練的運用這種語言，除非讀者實際動手去使用這種語言。有了這種認識，本書的格式即設計成 LISP 解譯程式的交談式章節。若想善用本書，讀者需根據各節所列的對話實際動手執行 LISP，依照對話的內容輸入運算式，然後注意 LISP 系統會產生什麼樣的反應。

本書與其說是本教科書，不如視爲旅遊指南來得恰當。若不實際動手試試，就無法眞正的學到東西。讀者首先須熟悉書中前面部份的章節所描述的語言基本性能，這可讓讀者擁有足夠的知識，以充分應付未來章節中學習上所會遭遇的任何狀況。

最後，作者鼓勵讀者儘量嘗試各種不同的用法。並非讀者所碰到的每件事，在一開始時就能看出它的意義和價值。雖然要了解一個國家的很好方式是依循大衆化的路線而行，但是也不要忽略了“雖小道，亦有可觀者焉”這

句話。這也是事物總能引發人們的興趣和好奇心的主要原因。

祝各位一帆風順。

LISP基礎

易懂、實用和高吸引力是這本書的特色——想要了解LISP 的學生和想要深入發掘LISP 語言特色的程式設計師，都應該閱讀這本書。本書的作者，威廉斯基教授，從最基本的觀念談起，把在LISP解譯程式上執行過的活生生例子呈現在讀者面前，鼓勵讀者動手去做，然後介紹了幾近於全部的LISP 特性，例如LISP 內部結構、罕用的資料型態和函數、偵錯問題、錯誤處理問題以及系統函數等等。這本書最精華的部份在於語言的工作環境之探討，包括常用的程式設計慣用法、繁雜的LISP 結構和無數的程式設計風格，這些都以使用者的觀點做爲討論的基礎；所有的“技巧和手腕”都得自於LISP 程式設計經驗的累積。

威廉斯基教授把重心放在Franz LISP 上。這個LISP版本可在柏克萊UNIX 上使用。書中凡是和其它版本的LISP 不同之處，都會附上詳細的說明。

這本書完全以語言的介紹爲主，對於想研究語言架構的人而言，這是本最佳的教科書。書中最後兩章也分別針對樣式配對以及聯結式資料庫管理發展了兩個應用程式。

羅伯特・威廉斯基是加州柏克萊大學的計算機科學副教授。他在耶魯大學拿到博士學位，並參加了耶魯大學的人工智慧計畫。他的研究領域著重於常識理解和自然語言處理這兩方面。

儒林圖書公司 印行

目 錄