

Artificial intelligence

人工智能-AutoLISP-认知与实践

Cognizance & practice

Character Quality Fined

—CAD 汉字谐和技术—

CQF 软件包 **揭密**

编著 江苏省电力公司 南通供电公司
南通电力设计院 陈启发



Artificial intelligence

人工智能—AutoLISP—认知与实践

Cognizance & practice

Character Quality Fined

— CAD 汉字谐和技术 —

CQF 软件包揭密

编著 江苏省电力公司 南通供电公司
南通电力设计院 陈启发



中国电力出版社

www.cepp.com.cn

内 容 提 要

AutoLISP 是当今广泛流行于工程设计行业和制造业、服装设计、教育部门等各界的一种人工智能语言。

本书分为上下两篇：认知篇就 AutoLISP 的特点和相关编程技术等，进行详尽分析介绍；实践篇公开《CQF 软件包》的全部程序、展示 CAD 汉字诸和技术。书末有附录 CAD 文献常见词汇。

本书可供设计师和大专院校师生研究 AutoLISP 时参考，更是程序员必备的工作手册。

图书在版编目 (CIP) 数据

人工智能—AutoLISP—认知与实践—CAD 汉字诸和技术—
CQF 软件包揭密/陈启发主编. —北京：中国电力出版社，2004
ISBN 7-5083-1958-3

I. 人… II. 陈… III. 计算机辅助设计—应用软件, AutoLISP
IV. TP391.72

中国版本图书馆 CIP 数据核字 (2003) 第 116919 号

中国电力出版社出版、发行
(北京三里河路 6 号 100044 <http://www.cepp.com.cn>)

汇鑫印务有限公司印刷

各地新华书店经售

*

2004 年 5 月第一版 2004 年 5 月北京第一次印刷

787 毫米×1092 毫米 16 开本 16.75 印张 393 千字

印数 0001—2000 册 定价 29.00 元

版 权 专 有 翻 印 必 究

(本书如有印装质量问题，我社发行部负责退换)



郑州大学

04010122444M

-48

前 言

10年前，CAD应用工程刚刚启动的时候，国内开发的科研成果往往还不能转化为商品，而从国外引进的CAD系统又不能快速地转化为效益。曾有很多分布在院校、科研所的课题组从事CAD技术的开发与研究，其后果多数是“交了学费”。根本原因在于人们对这种先进生产力的研制、开发规律和推广应用特点认识不足，同时由于程序开发对计算机专业技术要求较高，因而在开发过程中缺少生产一线的设计和制造人员的参与。

本书介绍的AutoCAD二次开发技术简单、易学，降低了技术人员参与开发的门槛，有利于CAD科研成果的商品化过程。CAD产业是典型的知识产业，它的主要资源和载体是掌握现代化知识的工程技术人员，技术人员通过简单的、智能化的程序语言，将长期积累的设计、工艺和制造经验表达为可重复利用的计算机程序，对于知识的继承是很有帮助的。从这个方面讲，类似于本书所述及的通用的、简单的工程程序的开发技术应该向企业中的技术人员推广，要抓好制造业信息化第一道工序——人才的培养和知识的积累工作。

从实际工程应用的角度出发，本书还是值得广大工程技术人员阅读的。

CAD/CAM与制造业信息化（前《计算机辅助设计与制造》）杂志社

副主编 朱辉杰

鸣 谢

本书能得以顺利出版,首先应感谢的,是国家科学技术部前工业科技司副司长、全国 CAD 应用工程办公室主任——陈贤杰先生。他与我素昧平生,却两次热心地为我牵线搭桥,联络专家给拙著作选题评审、并安排撰写前言予以推介;笔者相信,这是像我这样名不见经传的普通技术人员之大幸。

应该感谢《CAD/CAM 与制造业信息化》(前《计算机辅助设计与制造》)杂志社的副主编朱辉杰先生为我精心编撰的前言,把一个工程设计界作者的书,推荐给了制造业的广大工程技术人员。

感谢我的老朋友,华东电力设计院 CAD 设计总工程师耿兆良先生的帮助,为我联系审题,并亲自审阅全书。

本书的出版,还受到中国科学院软件研究所主任研究员戴国忠先生、中国电力规划设计总院计算机处处长王聪生先生、上海交通大学计算机系教授何援军先生等的支持;此外,同济大学计算机学院副院长李启炎先生和清华大学计算机系教授苑春法先生也曾对拙著的初步提纲提出过宝贵意见,在此一并致谢。

笔者还深深感谢我的同事们——南通电力设计院(前南通供电局设计室)的全体工程师,多年来对笔者编制CQF软件包的支持、试用和许多改进建议,以及王家瑞院长对本书出版的有力支持。

最后,还要感谢我的两位同事,高级工程师孙晓威先生和李龙芳先生对本书实践篇英译简介的校勘工作。

笔者 2003.3.2

目 录

前言
鸣谢

导论..... 1

上 篇 — 认 知 篇

第 1 章 了解 AutoLISP	9
1 基本概念.....	9
1.1 ADS、ARX、Automation 与 AutoLISP	9
1.2 子程序传递结果参数的数量、位置及其他.....	11
1.3 局部变量限定的必要性.....	12
1.4 变量初始值的设置.....	12
1.5 数据记忆.....	13
1.6 函数与变量之异同问题.....	13
1.7 对字符串 string 长度的限制.....	15
1.8 关于反斜杠 \ 控制字符、其他符号及相关问题.....	15
1.9 AutoLISP 禁忌和函数的异常返回值.....	15
1.10 DCL 对话框.....	17
2 autload 之实践.....	18
3 command 相关问题.....	19
3.1 赋值方式.....	19
3.2 函数命令与 command 语句.....	19
3.3 拾取点须在当前屏幕内.....	19
3.4 Cmdecho 与带捕捉的 command.....	20
3.5 EXPLODE.....	20
3.6 关于 ZOOM_E.....	20
3.6.1 与 Extmin 和 Extmax 的残值问题.....	20
3.6.2 与 DEFPOINTS 层的关系.....	21
3.6.3 与 Windows 界面的关系.....	21
4 cond 与 if 之差别.....	21
5 defun.....	21
5.1 defun 定义函数 与 AutoLISP 函数的差别.....	21

5.2	defun 定义的函数命令	22
5.3	带执行语句的函数	22
6	entmod	23
7	entnext 与 Lwpolyline	23
8	error	24
8.1	致命错误	24
8.2	*error* 的关闭和恢复	26
9	exit & quit	26
10	get 函数	26
10.1	get ... “与汉字提示”	27
10.2	getint “aaa”	27
10.3	get 与空响应	27
10.4	getfiled 的返回值	27
11	initget 与 getkword	29
12	关于 load	29
13	nil, null 及其他	29
14	open、close 与文件类型	30
15	ssget	31
16	strcat	31
17	trace 的运用和程序调试	31
18	while	32
第 2 章	AutoCAD 版本和 CAD 经验杂论	33
1	AutoCAD 的版本	33
1.1	R10	33
1.2	R12	33
1.2.1	R12 的文本窗口	33
1.2.2	R12 中文版的环境设置	34
1.3	R14	34
1.3.1	R14 的文本窗口	34
1.3.2	R14 与低版本转换时的中文乱码问题	34
1.4	R20	34
1.5	单机与绘图机采用统一版本的问题	36
2	CAD 经验杂论	36
2.1	Color 颜色、色标及其他	36
2.2	DIM_HOM 与 DIMOVER_C	37
2.3	Extmin、Extmax 与 R14 的关系	38
2.4	INSERT 实体与阵列式 INSERT 实体	38
2.5	INSERT、MEASURE 与属性块的关系	38

2.6	INSERT 与 XREF 的关系	38
2.7	Menu 和屏幕菜单定制	38
2.8	Scale 比例尺及其他	40
2.8.1	按 1:1 输入实体	40
2.8.2	实体比例、蓝图比例和打印比例	41
2.9	变更 DimScale, 实现连续尺寸标注	41
2.10	SELECT_ ALL 与 W 之间的差别	42
2.11	SHX 形文件空缺时的处理方法	42
2.12	中文乱码及其处置问题	43
2.13	STRETCH 对 HATCH 实体无效	43
2.14	用 WBLOCK 命令清理图形文件	43
2.15	OLE 对象	44
第 3 章	程序编辑、编程工具及其他	45
1	程序编辑	45
1.1	编程是否必须打草稿	45
1.2	关于空格、退格编程及其他	46
1.3	关于大小写	46
1.4	关于文件换名	46
1.5	编程专用程序及预制程序段	47
1.6	COPM.lsp 的使用	49
1.7	程序调试	49
1.8	调试程序与换名	50
2	编辑器的选择	50
2.1	几种编辑器之比较	51
2.2	关于 EDLIN	53
2.3	EDIT 软件与汉字	53
3	关于纯文本处理	54
3.1	Word 的纯文本处理方法	54
3.2	WPS 的纯文本处理方法	54
4	汉字输入和自然码	55
4.1	自然码的优异特点	55
4.2	自然码与 ACE 环境及其他	57
5	Word 与 AutoCAD 图形及其他	57
5.1	插入屏幕图像	58
5.2	插入图片	58
5.3	图片与 AutoCAD 色彩的关系	59
附 5.3	关于 Word 文字的色彩	61
5.4	图片尺寸与 AutoCAD 的关系	61

5.5	修改图片实体以及 VPOINT 的变换	62
5.6	图片中实体偏细、偏长的处理	62
5.7	防止损坏插入的图片	62
5.8	Word 图片小结	63
5.9	Word 中的通栏横线	64
5.10	过分敏感的编号及其他	64
5.11	关于目录	65
5.12	双面打印和其他	65
5.13	关于 Word 文件的容量	66
5.14	关于 WPS 的图形对象	67
第 4 章	平台环境及外部设备	68
1	WINDOWS 与 DOS	68
1.1	Windows 中有什么问题	69
1.2	关于虚拟盘的死机和异常退出 AutoCAD 问题	69
1.3	项目 BAT 文件的编制	69
1.4	关于文件名	73
2	屏幕形制及其背景色	73
2.1	文本编辑软件的背景色	74
2.2	AutoCAD 的屏幕	74
2.2.1	textscr 屏幕形制	74
2.2.2	ghaphic 图形屏幕	75
2.2.3	屏幕配置及其他	75
3	外部设备	85
3.1	关于硬盘损坏	85
附 3.1	更新计算机的烦恼	86
3.2	关于打印机	87
3.2.1	Windows 与打字机配置	87
3.2.2	软件锁和打印机的异常关机	88
3.3	关于鼠标器失灵	88
3.4	关于 UPS 与市电系统短路	88

下 篇 — 实 践 篇

引言	93
1 没有捷径——代序	93
2 CAD 汉字谐和技术	94
2.1 CAD 文字的生成	94

2.2 关于形文件 .SHX 及字体变形	95
2.3 解决途径	98
2.4 ACE 和空格	98
CQF 软件包简介	98
3C.lsp	99
ABC.lsp	100
ADDP.lsp	102
APLOT.lsp	103
ARCLINK.lsp	108
ATOML.lsp	108
BG.lsp	109
BH.lsp	111
BT.lsp	113
CHVC.lsp	114
附: README.hvc	118
CHVC.dcl	124
CHVC.txt	129
CLA.lsp	130
CLB.lsp	131
COMP.lsp	133
CQF.lsp	134
CQF.dcl	137
CSH.lsp	141
CTT.lsp	143
CTXT.lsp	146
DEFLIST.lsp	149
DIA.lsp	154
DINT.lsp	154
DWGP.lsp	155
DX.lsp	157
ECAD (屏幕菜单段)	157
ENT.lsp	159
ETXT.lsp	159
FLIST.lsp	162
FP.lsp	162
GFD.lsp	163
GJBZ.lsp	164
GMD.lsp	167

GMR.lsp	172
GP.lsp.....	174
HBLX.lsp	179
HNT.pat	180
INSTAL.lsp	181
JSTJ.lsp.....	184
JSTS.lsp.....	185
JZTXT.lsp.....	186
lightN.lsp.....	186
LL.lsp	187
LOD.lsp.....	188
MB.lsp	189
ML.lsp	190
N.lsp	193
PLC.lsp.....	194
PLCD.lsp.....	195
PLV.lsp.....	196
QB.lsp.....	199
QWM.lsp.....	201
QZ.lsp	202
STLIM.lsp	204
TEXTF.lsp	206
TJ.lsp	207
TK.lsp.....	210
VERSION.lsp.....	216
WTT.lsp.....	227
WTXT.lsp.....	228
XG.lsp.....	229
XT.lsp	230
YCFH.lsp.....	232
ZDX.lsp	232
ZK.lsp.....	236
附录 CAD 文献常见词汇	238
跋.....	250
附《AutoLISP 资源检索手册》简目	252
参考文献.....	255

导 论

一、人工智能研究和 AutoLISP

许多读者始终将人工智能研究视为一个很难企及的神秘领域。而实际上，每一个使用 AutoCAD 的普通设计者，都早已在接触、使用或在开发人工智能。

按照现在的观点，所谓人工智能 (artificial intelligence)^①，就是研究解决某些通常认为要用智能才能解决的问题之计算机技术^[1]。如所周知，已经出现并且还将出现许多种计算机语言，诸如 ALGOL、BASIC、C++ 等等。应当说，每一种语言都可以成为人工智能研究的工具，但其中的 LISP (LISt Processing language)^②，即表处理语言，是目前人们公认的特别适合于人工智能应用的一种语言^[2]。

1959 年，麻省理工学院 (Massachusetts Institute of Technology) 约翰·麦卡锡 (John McCarthy) 研制成功 LISP。经过几十年的发展，今天，这种语言已经繁衍成包括 EVALQUOTE-LISP、EVAL-LISP、MacLISP、InterLISP、ZetaLISP 和 CommonLISP 等多种语支的一个语系。

由戴维·贝茨 (David Betz) 以其开发的 XLISP 为基础，加以发展并嵌套在 AutoCAD 软件包中的 AutoLISP，是 CommonLISP 的一个小小的子集 (当然也有若干专用的 CAD 函数)。从而使得这一通常只有科学家们研究的玄妙无比的人工智能语言，走出象牙之塔，变成许多普通工程师手中得心应手的工具；可以说，AutoLISP 的诞生，开创了人工智能研究的新纪元。尽管它只是一个很小的子集，但其强大的功能早已令许多设计行家吃惊：“AutoLISP 在设计领域，几乎无所不能！”

二、AutoLISP 研究的现状

今天，全世界有成千上万的工程师在用 AutoLISP 从事开发程序的工作，在所有已知的科学研究领域中，这是一个非常特殊的群体。

首先，他们大多是没有受过专门的有关人工智能开发培训的普通工程师。事实上，还没有这样的专业学校；而即使有这样的学校 (或培训班)，除了能对掌握 LISP 稍有帮助外，在别的任何方面均将一无所能，因为人工智能可能研究的领域无所不包，这种学校当然也就无可包容了。所以，他们只有通过自学来获取相关的知识。

其次，他们都具有极强的事业心，不仅有将自己日常反复从事的某一单项工作智能化，让繁琐的操作交由计算机自动完成的愿望；而且能不辞劳苦、废寝忘食地将愿望变成现实，编制出一个又一个 LISP 程序，除了自己使用以外，还经常无偿地提供给别人使用，使本部门、本单位以致更多单位相关工作的效率，都能成倍、甚至数十倍地提高，却从来也不指望得到任何形式的报酬或表彰，用户说一声好用，就会使他们感到心满意足。

① 人工智能的研究，可以追溯到 1834 年美国查尔斯制造的机械计算机。1956 年在美国召开的达特默思会议、1977 年斯坦福大学爱德华·费根鲍姆提出的知识工程，为人工智能的应用奠定了基础^[3]。

② 此外，还有 SAIL 和 PROLOG 等，前者在机械手控制程序研究中获得应用，而后者在第五代计算机研究中受到重视，被人们比作人工智能领域中的高级语言；相对而言，LISP 则被喻为人工智能中的汇编语言^[4]。

在我国,除了几个软件公司曾组织一二百人专业从事过这一开发以外,自发进行 LISP 开发的,估计不下 5 万人^①,但大多只在极小的范围内运作,很少交流。每年通过各种书籍、杂志等公开发表的 AutoLISP 程序,最多时约达 50 个左右;近年来,由于 ADS、ARX 及 Automation 等的先后出现,数量逐渐萎缩,每年不足 20 个。在美国, CADALYST 杂志每年大约发布 100 个程序(迄今总共 1500 个左右),另外还有 Autodesk CompuServe 论坛等,发布情况不详。

三、傻瓜设计系统与 AutoLISP

就 CAD 技术本身而言,可以有三个发展层次的理解:

- (1) 计算机辅助绘图 (Computer Aided Drawing)。
- (2) 计算机辅助设计 (Computer Aided Design)。
- (3) 计算机自动设计 (Computer Automatic Design)。

目前,许多设计人员应用 CAD 的水平,基本上停留在计算机辅助绘图的层面上,先画一条线、再来一个圈……,这样一点不拉地用计算机进行绘图,即所谓甩图板。不少人感到 CAD 最方便的地方,莫过于复制和修改,这是初级层次。

但是显然,初级层次的 CAD 存在两大遗憾:

其一、CAD 图纸的差错率高于手工图纸。

大家知道,设计者的出手质量是第一位的;而校对、审核者对图纸质量的把关作用,就如世界级的顶尖守门员也会漏球一样,总是非常有限。

有手工绘图经历的设计师都知道,每次新画一个细节时,只需稍稍移动一下视线,就可以看清楚已经绘就的部分,对照比较、读数检查,既方便又准确;因此,过去有许多老工程师都能够作到无差错出手。

CAD 设计则不然,屏幕尺寸再大,也无法在同一平面中看到图形之全部,而反复的 ZOOM 和 PAN 又往往令人心烦,所谓挂一漏万便在所难免;因此,即使是一位设计经验极其老到、耐心超乎寻常的工程师,也无法保证其 CAD 图纸的质量达到 100%,即免检的水平。

其二、CAD 标注文字和尺寸时有点烦。

CAD 特别适宜于 1 : 1 输入实体,然后根据出图比例的要求,标注尺寸、文字、套上图框,最后用 PLOT 命令绘成底图。

也正因为如此,往往使设计者在标注尺寸、文字时感到为难,以致手忙脚乱,不知所措。经常看到这样的成品,在同一张图内,标注文字的高度大相径庭,大者可超过 25mm,而小者却不足 1.8mm,根本无法分辨。

许多人至今不肯采用 1 : 1 输入实体的原因,很多都与此相关。

① 《江苏勘察设计》2002-1 载:1998 年全省 443 个勘察设计单位拥有微机 1.26 万台,技术人员平均 1.3 人 1 台微机;至 2001 年,单位发展为近 800 家,职工总数达 2.6 万人。工程设计行业使用的 CAD 软件以 AutoCAD 或以其为母体的软件为主,按此估计,江苏省勘察设计界目前使用 AutoCAD 的人员超过 1.5 万人,制造业、服装设计、教育部门等各界使用 AutoCAD 的总人数可能与此相近。

据全国 CAD 应用工程办公室主任陈贤杰先生介绍(载《计算机辅助设计与制造》2001-6、7):“九五”计划期间全国累计培训 CAD 应用人材 118 万人次。一般认为,AutoCAD 用户约占 CAD 软件用户总数的 1/2,按此推测,我国使用 AutoCAD 从事设计工作的工程师总数应在 50 万以上。

按编者所在部门估计,在 22 位用 AutoCAD 从事设计工作的工程师中,有 3 位能熟练应用 AutoLISP 编程,超过 10%;则全国使用 AutoLISP 编程的人数应逾 5 万人;从全球 400 万注册用户推测,数字将更大。

上述两点，已为大量的实践所充分证明。

从软件业界考虑，CAD 虽然提供了日益友善的交互输入界面，但并不希望用户始终停留在计算机绘图的水平，因而不断翻新二次开发技术接口，让用户能比较方便自在地编制出各种程序进行参数设计，即得到普遍认同的计算机辅助设计，这是目前所达到的层次。

如所周知，设计本身可以划分为两个等级，创新设计和应用设计。

前者是在基本上无范本借鉴、无规范约束的情况下实现的，通常只有少数先知先觉者凭借其超人的灵感来实现，即创造和发明，在人工智能尚未完成自动学习和自动推理的条件下，不可能或许永远也不可能实现创新设计的自动化。

后者，即应用设计，则大多像服装师量体裁衣那样，只是根据已有的模式按照具体项目的实际需要适当调整尺寸而已，通常很少或根本没有什么创新，所以有人将设计师称为打样师傅或设计匠。从这样的层面看，实现自动化设计就完全可能。

人类设计了许多自动化系统，从面包生产线到飞机自动着陆系统……，而应用设计本身却没有实现自动化。现在，人类所积累的知识总量，已经到达能够向这个课题进行冲击的水平。

显然，计算机自动设计是借助计算机完成应用设计的最高层次，是今后 CAD 技术进步的必然结果。

严格意义上的自动设计，应当像业主提出定单那样，只向计算机输入几个简单的指令，诸如工程规模、基本参数和若干特殊要求等就可以了，剩下的工作全由计算机完成，并输出全套的图纸和说明书文件，无需任何人再次介入；至多只是在方案比选上交互动征询一下设计者的意向而已。判断一个软件是否达到了自动化设计的水平，可用‘傻瓜型’作为界定的标准，其理由不言而喻，无需赘述。

举例来说，如果能将早已实现的零部件参数设计，像某些齿轮设计程序，加入方案比选和参数计算等所谓的前处理程序，联结在一起，就是齿轮自动设计系统。设计者只需将要求制造的齿轮之传递功率（或力矩）和转速、联结类型等最原始的数据要求输入计算机，就可以完成设计工作；因而操作这个系统的人员可能只是一位对设计知之不多的销售经理。这类零部件自动设计系统估计在几年之内就会出现。

当然，具有独立功能的产品（工程设计也是一种产品）完全实现自动设计，则大约要在二三十年之后才能出现。其中最大的困难，绝对不是技术水平，而是资金！因为一种产品的自动设计，不再能简单地被称之为‘专家系统’，而应是一个需要许多人合作的极其庞大的系统工程。从某种角度而言，它所需要投入的人力资源，可能不亚于曼哈顿计划，因此从商业或经济的角度看，近 10 年之内大概不会有风险投资者愿意^①斥巨资去招募一大批专家来从事这类产品的开发工作。

但是，自从人类诞生以来就有的这一不断演进的梦、工程师们长期存在于脑海中的理想——完全的自动设计系统——作为人类智慧的巨大结晶，终将会呈现在世人的面前。

毋庸讳言，自动化设计系统所能给你的，一定不会是最优秀的作品，因为它只能总结和沿用人类已有的知识和体验，不可能出类拔萃；但是，它绝对比那些初出茅庐者的设计强千倍，无证设计者的成品当然更难望其项背。就像傻瓜相机受到非专业的摄影者喜欢一

^① 风险投资家最担心的，可能是软件的独享（或即保密）问题；只在开发中心保存一份工作软件（当然还有备份），而对外发行的仅为数据收集软件，并通过网络传递委托数据和电子成品文件，可能是一条途径。

样，傻瓜设计系统也一定会受到在数量上占绝对多数，对经济性要求很高、对工期要求较紧的业主们的欢迎，这便是此类系统的社会价值。

如果有位对建筑设计一窍不通的人，能凭借自动设计系统亲手为自己设计一座小别墅，得到完全随心所欲的满足，难道有什么不好吗？难道你不认为在中间，正蕴含着巨大的商业机会和市场价值吗？

当然，工程师们根本不需要担心会因此而失业。总有许多业主钟情于富有创新意识、标新立异的设计师，就像许多大款、明星愿意出天价请声誉卓越的能工巧匠手工定制衣帽皮鞋，而不屑在市场上购买随处可见的、便宜得多的成品服饰一样。同时，自动化程序的编制者，也只能从他们中间产生，正如最适用的牙医工具总是由牙医师首先开发出来，而不是医疗器械厂的工程师一样；由一群刚从大学毕业、毫无实际经验的计算机专业博士生从事自动设计系统的开发，是难以想像的。再者，傻瓜设计系统一旦真的问世，肯定也会受到这些工程师们的喜爱，因为从此可以毫不费力地获得设计方案的毛坯（姑且这么说），然后稍事修饰，加入若干可称为完全手工设计的点睛之笔，像今天动辄加上‘绿色、天然’的标记一样，去满足某些业主的需要。

以上所说的一切或即傻瓜设计系统的出现，毫无疑问地需要使用人工智能语言，这便是学习 AutoLISP 的诱人前景。

四、AutoLISP 的特点

这种语言的最大特点，是简便易学。

首先，任何别的程序语言，都要作变量和变量类型定义，且不能变动，这就需要预先苦苦思索出一大堆变量，变一个角色换一个名，既累又烦。而 LISP 的语法结构简单，变量和变量类型完全由赋值 (setq) 结果确定，什么时间需要便在什么时候赋值，变量类型在运行中间可任意变更，必要时还可以测试 (type)。例如开始时给变量 abc 赋一个字符，随即又根据这个字符把 abc 变成相应的实数，极其灵活；就如有位先生，在学生面前是老师、在子女面前是父亲，而在一群京剧票友之中，可能又是琴师一样。这样处理的结果，有效地减少了定义的变量名，既节约内存，又便于记忆和学习。这是 LISP 的一大特色——易学。

其次，别的语言，例如当前最走红的 C++，需要先找到与平台相应的头文件，然后编译、连接，缺一不可，R12 ADS 程序就必须在 Borlan C++3.1 平台编译，换了别的平台，就用另外的一套头文件，常常因此而构成困难。而 AutoLISP 属于解释性程序，并被直接嵌套在 AutoCAD 内部（有人认为这是其致命伤^[36]，而笔者的观点刚好相反），解释平台就是 AutoCAD 本身，因此象 BASIC 语言那样，随编随读随执行，使其优点得以充分发挥，调试和运行都很方便。这是 LISP 的又一个特色——简便。

正因为有以上两个特色，才能使 LISP 成为人工智能领域中最具影响力的语言，她在设计界所拥有的研究者人数，其他任何语言都难以与之匹敌。当然，LISP 也有缺点，主要是速度慢、保密性差及对外部文件的操作能力弱。但是，与交互输入相比，速度并不是一个重要问题；至于保密问题，除了 AutoCAD 本来使用的保护模式之外，还有多种加密方法可以采用^[46, 47]。

由于 AutoLISP 的 read 函数读文件时以 ‘1A’ 为终止判据，因此无法读写二进制文件。也使某些类型的文件，如数据库的 .DBF 等的读入产生困难，为此，数据库需规避结构部分

出现字段位置在‘1A’的可能，有时因此要调整字段结构的长度；当然，自从 R12 提供结构查询语言 SQL (Structured Query Language) 以后，这个问题或许已经消除。

最后，作为一种人工智能语言，AutoLISP 的功用绝不仅仅局限在计算机辅助设计，如参数绘图、建模和图块处理、机器运动仿真^[32, 33, 37, 42]等的范围之内；

它也可以被用于 CAM 数控编程^[43]，还可实现与 FORTRAN 等接口^[44]；

除此之外，它对文件排版处理（如 PB.lsp^①）、编程（如 COMP.lsp、DEFLIST.lsp）以及上面提及的数据库管理等等，都有用武之地。

五、升级到 Visual LISP 的问题

1998.3, Autodesk 公司推出了 AutoLISP 的升级版— Visual LISP，即所谓可视化 LISP 编程软件，并于次年嵌入 AutoCAD 2000 内部。

毫无疑问，Visual LISP 的功能要比 AutoLISP 强大得多。首先，它能将明码的 LSP 程序编译成 ARX 格式文件，一举解决了功能、速度和保密等问题；其次，它融入了所谓面向对象的 ActiveX 接口技术，除了 AutoLISP 能访问的实体和符号表对象等之外，还可以对 AutoCAD 图形及其应用程序本身进行操作，使其成为在 CAD 领域中与 Visual Basic 等难分伯仲的崭新的软件开发工具。

Visual LISP 的最大特点，可能在于关联 Related 和反应器 Reactor 两方面。大家知道，现有图形中的所有实体，除了标注尺寸和属性块之外，相互之间几乎都没有关系，只是一个集；而建立反应器机制之后，实体与实体之间会相互关联，改变其中之一，其他实体会随之变更，使图形类似于一个有生命的机体，从前瞻的角度看，这无疑十分诱人。

但事物总是有其两重性，功能强大通常会带来技术的复杂化，比较[5-P18]和[11-P289]可知，同为花园路径的绘制程序，后者要比前者复杂得多（当然，后者加入了对话框等功能）。对于一般的简单应用而言，这种复杂化可被视为多余，俗称杀鸡用牛刀；但对研制功能更加丰富多彩的商业软件来说，采用 Visual LISP 编程则是顺理成章的结果。

无论是否升级到 Visual LISP 编程，掌握 AutoLISP 总是必由之路。因为除了 Visual LISP 新增的 vl-、vla-、vlax-、vlisp-、vlr-等五类约 1390 个 (R20) 函数之外，大量使用的，还是 AutoLISP 基本函数（本书所述，也仅限于 AutoLISP；如果你在 AutoCAD 2000 平台上，当然会使用 Visual LISP 编程器工作，但只要不打算使用上述新增函数，那么本书所述，足敷应用；而编译成 .arx 文件等问题，则属于解决速度和保密的范畴，应另当别论）。

六、入门并不难，得道唯有不懈

让我们也先来炒作一个热门新词 DIY，作为本节的引子，一切都靠你 Do It Yourself——自己动手。

学会 AutoLISP 并不难，凡是能读懂绘制花园路径 PATH 程序段^[5]的人，均可入门。大体说来，一个悟性平常的工程师，学习 127 个函数（指 R10 版基本函数，其实常用的函数不到 50 个，R12、R14 增加了若干函数，但不常用）二至三遍，便可基本理解，自学时间不会超过一个月，再花同样的时间将学会的函数名称记得很熟至能背诵的程度（这样比较

① 该程序系将 AutoLISP 程序文件直接排成方正系统版面的论文，读者如有兴趣，可联系。本书所述及其他程序，凡无说明者，均见下篇——实践篇。

容易入门), 便可参考花园路径 PATH 程序的样式着手编程, 先试着从一个很小的题目开始, 只有几行 (此时无需考虑实用价值, 能理解就可以), 运行成功以后你会感到信心倍增, 然后编稍长些的程序。当你由于成功而感到欣喜时, 便已经入门了; 这个过程, 不会超过三个月, 对许多人来说, 可能只要半个月时间就完成了。

至此, AutoLISP 编程的神秘面纱已经被你剥去, 剩下的问题, 无疑就是持续不断的努力。有道是, 世上无难事, 只要肯登攀。

曾有无数人登上过泰山之颠, 其中的许多人会情不自禁地联想起杜工部的千古绝唱。你也想在 AutoLISP 这一人工智能的领域中品味 “会当临绝顶, 一览众山小” 的美妙感受吗, 愿您成功。