

专家系统工具 **CLIPS** 及其应用

吴鹤龄 编译



北京理工大学出版社

专家系统工具 CLIPS 及其应用

吴鹤龄 编译

北京理工大学出版社

CLIPS 是美国宇航局推出的一个崭新的基于规则的通用专家系统工具。本书根据 1988 年 4 月的新版本 V4.20 详细介绍了 CLIPS 的基本组成, CLIPS 函数, 模式匹配算法; 讨论了推理的控制技术, 推理的效率, 用 CLIPS 设计正向链、逆向链的专家系统, 具有不确定性的专家系统, 解决一般判定树及有学习功能的判定树问题等的方法与步骤, 给出了计算过程的实例, 是一本深入浅出的介绍高新技术, 有很大参考价值和实用价值的新书。可用作高等院校计算机专业及相关专业本科生和研究生的选修课教材, 也可供从事人工智能研究和开发的中、高级工程技术人员阅读、使用。

若需适用于 IBM PC 机及其兼容机的 CLIPS 4.20 版软件盘片(包括 CLIPS 目标码及一些示范性的、可运用的 CLIPS 专家系统程序)可与北京理工大学出版社联系。

专家系统工具 CLIPS 及其应用

吴鹤龄 编译

北京理工大学出版社出版

新华书店北京发行所发行 各地新华书店经售

北京理工大学印刷厂印刷

850×1168 毫米 32 开本 7.5 印张 194 千字

1991 年 5 月第一版 1991 年 5 月第一次印刷

ISBN 7-81013-398-5/TP·29

印数: 1-3000 册 定价: 4.00 元

前 言

集计算机科学、神经科学、心理学、语言学、认知学、思维科学、控制论、信息论等多种学科于一身的人工智能,是当代主要的高科技领域之一。自60年代正式提出人工智能的概念和开展研究工作以来,虽然有过波折,受过非议(其中最严重的一次当然要推英国著名的数学家詹姆斯·莱特希尔爵士于1973年发表的一份研究报告,报告认为人工智能是一种不切实际的幻想,甚至是一种骗局,从而导致英国政府取消了对人工智能研究项目的几乎所有投资),但总的来说,它在飞速发展。尤其是它的三个主要研究方向:自然语言理解、机器人和专家系统,取得了令人瞩目的成果。

在使计算机模仿人的智能解决实际问题方面,由于人的神经系统及其思维活动的极大复杂性,所以一些学者试图实现通用问题求解系统(GPS,即 General Problem Solver)的努力虽然有一定进展,但是遇到很大的困难。而另一些学者所走的另一条道路,即让计算机只模仿某一特定领域的专家去解决有限制的问题,即建立专家系统的努力却取得了极大的成绩。形形色色的专家系统已经实用化、商品化,进入了软件市场,在软件产业中占有了重要地位。根据1986年10月在伦敦举行的第二届国际专家系统会议上的资料估计,美国和加拿大两国商业和政府用户1986年在专家系统技术上的投资超过4亿美元,专家系统软件产品和服务市场达到2亿8千万美元,预计1992年将达到19亿美元。在1985—1986年间,美国最大的500家公司中就已有80%竞相采用专家系统,取得了巨大的效益。以DEC公司的专家系统XCON(eXpert CONfigurer)为例,这个系统把计算机配置专家的经验总结为四、五千条

规则,可以根据用户的特殊需求,提出最佳的(既满足需要,又最节省)的 PDP 或 VAX 机的配置方案。该系统投入使用以来,已经为十余万 DEC 用户确定了配置方案,据 DEC 公司自己的报告,这个系统使公司每月节省职员费用 20 万美元,生产费用每年减少 1800~2000 万美元。其它成功的专家系统,如斯坦福大学的 MYCIN,匹茨堡大学的 CADAUCEUS,麻省理工学院的 MACSYMA,斯坦福研究所(SRI)的 PROSPECTOR 等也都有类似的记录。我国在人工智能方面的研究工作虽然起步较晚,但在许多科技工作者的努力下,也取得了很好的成绩。例如北京理工大学人工智能研究所与中央气象局合作开发的暴雨预报专家系统 IMFOS,成功地在一些省市应用,相当准确地预报了一些灾难性暴雨,避免了人民生命财产的重大损失,经济效益和社会效益十分显著,荣获 1989 年国家科学技术进步三等奖。

专家系统的开发当前有三种方式,一种是采用一般的高级程序语言,如 PASCAL,FORTRAN,C 等。一种是采用人工智能语言 LISP,PROLOG。再有一种是采用专家系统工具。专家系统工具也叫专家系统开发工具或专家系统建造工具,也有叫专家系统“外壳”(shell)的。这是专为开发专家系统而创建的程序设计语言或其他辅助工具。显然,这三种方式各有优缺点。但从方便、简易、利于缩短研制周期等角度看,采用专家系统工具来开发专家系统当然具有较大优点。XCON 的开发成功与利用了专家系统工具 OPS5 及其前身 OPS4 是分不开的(OPS 是 Official Production System 的缩写)。因此,在我国普及若干比较成熟的专家系统工具,对于尽快地开发出更多更好的、有我国特色的实用的专家系统,是有意义的。本书就是基于这一想法而编译的。所介绍的专家系统工具 CLIPS (V4.20 版)是美国航天局 80 年代末新推出的。它主要总结和继承了 OPS5,ART(Automated Reasoning Tool)等著名的专家系统工具的优点和长处,又有所发展和创新,功能更加完善,语法更加简洁,更加易学好用。

本书只集中介绍 CLIPS。有关专家系统的一般概念与原理,请阅读有关的书籍。书末我们给出了参考书目。

本书主要根据 CLIPS 的主要开发者 J · Giarratano 和 G · Riley 所著的《Expert Systems: Principles and Programming》(PWS-KENT Publishing Co., 1989) 一书中有关对 CLIPS 的介绍以及我们在试用 CLIPS 中的体会编写而成。89 级研究生苏莉文和肖艳为把书稿录入计算机做了大量的工作。北京理工大学出版社为出版本书给予极大的支持。在此表示深切的谢意。

错误之处,欢迎读者指正。

编译者

1990 年 12 月

目 录

第一章 CLIPS 的基本组成

1-1 CLIPS 中的符号体系	(2)
1-2 CLIPS 的基本组成和语言符号	(3)
1-3 事实	(5)
1-4 进入和退出 CLIPS	(7)
1-5 加入事实和撤消事实	(7)
1-6 规则	(8)
1-7 待议事件表及其执行	(10)
1-8 用于规则的命令	(12)
1-9 printout 命令	(14)
1-10 有多个规则的情况	(14)
1-11 deffacts 命令结构	(16)
1-12 调试命令 watch	(18)
1-13 调试命令 matches	(19)
1-14 调试命令 set-break	(22)

第二章 模式匹配

2-1 变量	(25)
2-2 事实地址和撤消	(26)
2-3 变量的多次使用	(28)
2-4 字段通配符	(30)
2-5 移动积木问题的示例	(31)
2-6 多字段通配符和多字段变量	(36)
2-7 字段约束	(40)
2-8 函数和表达式	(42)
2-9 用规则完成值的相加	(44)
2-10 约束函数 bind	(47)

2-11	在 RHS 部分函数的使用	(48)
第三章 控制技术		
3-1	输入方法	(49)
3-2	谓词函数	(52)
3-3	测试模式	(53)
3-4	谓词字段约束	(55)
3-5	相等字段约束	(56)
3-6	特权(salience)	(62)
3-7	阶段与控制事实	(64)
3-8	模式的“逻辑或”、“逻辑与”和“逻辑非”	(70)
第四章 CLIPS 函数		
4-1	控制流函数	(76)
4-2	文件输入输出函数和逻辑名	(78)
4-3	其他 I/O 函数	(82)
4-4	字符串函数	(86)
4-5	多字段函数	(87)
4-6	扩充的数学函数	(90)
4-7	实用程序命令	(93)
4-8	gensym 函数	(95)
第五章 推理的效率		
5-1	里德算法	(97)
5-2	模式网络(pattern network)	(100)
5-3	连接网络(join network)	(103)
5-4	模式次序的重要性	(107)
5-5	模式应如何排序以提高效率?	(113)
5-6	多字段变量对效率的影响	(114)
5-7	通过测试模式检查效率	(116)
5-8	用内部模式匹配原语提高效率	(118)
5-9	特权对效率的影响	(118)
5-10	一般规则和特殊规则	(121)
5-11	简单规则和复杂规则	(123)
第六章 专家系统设计示例		

6-1 确定性因子和 CLIPS	(127)
6-2 判定树	(131)
6-3 有学习功能的判定树	(134)
6-4 逆向链接	(145)
6-5 一个监控系统的设计示例	(159)
附录 1 CLIPS 命令和函数汇总表	(185)
附录 2 CLIPS 中的编辑程序 MicroEMACS	(193)
附录 3 解“猴子和香蕉”问题的 CLIPS 程序及运行过程	(197)
参考书目	(202)

第一章 CLIPS 的基本组成

专家系统工具 CLIPS 是由美国国家航天局 NASA 的约翰逊空间中心 (JOHNSON SPACE CENTER) 的人工智能部开发的。其设计目标和特点是:

1. 可用于管理、工业、商业、文教等各种领域的实际专家系统的开发, 具有广泛的通用性。

2. 适用于各种型号的计算机, 从 IBM PC 及其兼容机, 到 VAX、HP、SUN、MACINTOSH 直到 CRAY, 都可以运行 CLIPS, 具有对机型的广泛适应性。

3. 与已有的普通软件系统易于集成, 同 FORTRAN、ADA 等高级程序设计语言均有接口, 为用户提供了方便。

4. 用 C 语言写成, 运行速度很快。

5. 具有在不确定性下进行推理的能力。

由于以上特点, 加之 NASA 对 CLIPS 采取公开化和鼓励推广的政策, 在美国可以以极低的价格获得 CLIPS 的执行码和原码, CLIPS 在商品化的专家系统工具中深受欢迎, 成为一枝独秀。CLIPS 研制小组也因而获得美国航天科技大奖。

CLIPS 是“C LANGUAGE INTEGRATED PRODUCTION SYSTEM”的缩写。实际上, NASA 后来也开发了一个与 C 语言 CLIPS 版本完全兼容的 ADA CLIPS 版本。

CLIPS 与比它早的专家系统工具 ART、OPS5 有很密切的关系, 继承与吸收了它们的长处与优点。但由于 CLIPS 追求简单实用、低价格、高度的可移植性, 所以 CLIPS 只提供正向链接能力。逆向链接、假设推理、面向对象的程序设计等技术在 CLIPS 中是没有的。但 CLIPS 的历史证明, 这并不影响它成为一个有效的专家系统

工具。

1—1 CLIPS 中的符号体系

在详细介绍 CLIPS 之前,先把 CLIPS 中描述各个命令和结构的语法所用的符号体系作一介绍。

CLIPS 的符号体系有三种类型。第一种类型是需要完全照样输入计算机的字和字符,即不被<>、[]、{}所括住的所有内容。例如:

(example)

见到这个符号,意味着先输入左括号“(”,然后依次输入“e”、“x”、“a”、“m”、“p”、“l”、“e”,最后输入右括号“)”。

如果有方括号,表示其中的内容是任意的,可有可无的。例如:

(example [1])

方括号中的 1 是任选的,即输入(example)和输入(example 1)是一样的。

如果有尖括号,表示其中的字所表示的意义要用一个相应的任意值代替。例如:

(example <integer>)

表示 example 后要跟一个任意整数,即应输入(example 1)或(example 5)或(example -20)……。

第二种类型是用双尖括号括住的描述。这表示任意多个(包括 0 个)与其中的字相应的值。例如:

<<integer>>

见到这一描述,可输入 1,或 1 2,或 1 2 3,或 1 4 8 -15,...,也可以根本不输入什么内容。

至于用三对尖括号括住的描述,这表示任意多个(但不包括 0 个)与其中的字相应的值。对这一符号,<<<integer>>>相当于<integer><<integer>>。

第三种类型是用大括号括住的描述。这表示选用其中的任意一项(必需一项,但也仅一项)输入。例如:

{all, none, some}

则可以输入 all,或输入 none,或输入 some。

1—2 CLIPS 的基本组成和语言符号

CLIPS 是用于开发基于规则的专家系统工具,其基本组成有三:

1. 事实表,即事实库。用于存贮数据(事实)。
2. 知识库。包括所有规则。
3. 推理机。用于控制推理程序的执行。

CLIPS 的表示和推理能力同 OPS5 类似,其语法则同 ART 的一个子集相似。

在建造知识库时,CLIPS 需要从键盘或文件读入以执行命令和给程序加载。CLIPS 把从键盘或文件读入的一组有特定意义的字符叫作一个“令牌”(token)。令牌是 CLIPS 的基本语言符号单位。有些令牌只有一个字符,如左括号“(”和右括号“)”。

在 CLIPS 中特别重要的语言符号单位是“字段”或叫“域”(field)。字段是由一组令牌组成的,有三种类型。

第一种类型叫“字”(word)。一般来说,字以可打印的 ASCII 字符开始,后随零个或多个字符。但下列字符及其组合不得作为字的开始字符:

< | & \$? + - () ;

此外,字中间也不能包含下列任意字符:

< | & () ;

这是因为以上 6 个字符在 CLIPS 中有特殊用途,即作为字的限界符(delimiter),表示字的结束。根据以上规定,以下均为有效的 CLIPS 字:

emergency
fire
emergency-fire
activate-sprinkler-system
notify-fire-department
shut-down-electrical-junction-387
!? # \$ ^ *

CLIPS 维持令牌中的大、小写字母,即 CLIPS 认为同一字母的大、小写是两个不同的字符。因此,在 CLIPS 中下面是三个不同的字:

FIRE fire Fire

第二种类型叫“字串”(string)。字串以双引号开始和结束,双引号本身是字串的一部分,其间可有零个或任意个字符。例如:

"Activate the sprinkler system."
"Shut down electrical junction 387."
"! ? # \$ ^ "
"<-;()+-"

在 CLIPS 中,空格一般用作字段(诸如字)和其它令牌的分界。在这种情况下,多个连续空格只当作一个空格,多余空格舍弃。但在字串中,空格作为有效字符有一个算一个,因此,以下字串在 CLIPS 中是不同的:

"fire" "fire" " " fire" " fire "

如果去掉以上四个字串的引号,那么它们变成四个相同的字。

因为双引号是用来定界字串的,所以字串中不能再用双引号。

例如:

""fire""

将被 CLIPS 当作是""、fire、""三个令牌。

如果在字串中一定要用双引号,那怎么办呢?这可以通过使用后斜杠“\”来解脱后随的双引号作为定界符的作用来实现。例如:

`"\fire\"`

在 CLIPS 中将被当作一个字符串`"fire"`。

由于后斜杠有这样一个作用,因此当字符串中需要用到后斜杠时,必须用连续两个后斜杠。例如:

`"\\fire\\"`

在 CLIPS 中被认为是字符串`"\fire\"`。

第三种类型叫“数值字段”或简称“数值”(number),它表示浮点数。在 CLIPS 中,所有数,包括整型值的数,都以浮点数形式存储。数值字段包括三个部分:符号、值、指数。但符号和指数这两部分是任选的。符号或为+或为一。值包括一个或多个数字以及一个任选的小数点。指数包括字母e或E,后随任选的+或-,其后是一个或多个数字。根据以上规定,以下都是 CLIPS 中有效的数值字段:

1 1.5 .7 +3 -1 65 3.5e10

1—3 事 实

为了解题,CLIPS 程序必须有据以推理的数据或信息,这在 CLIPS 中叫做“事实”(fact)。事实由配对的左、右括号括住的一个或多个字段所组成,例如:

`(single-field) (tow fields) (speed 38 mph)`

`(cost 78 dollars 23 cents) (name "John Doe")`

虽然 CLIPS 可以接受任意的字段组合作为事实,但从好的程序设计的角度出发,当然希望事实能够表达确定的意义。例如:

`(fire) (flood)`表示出现火警和洪水;

`(Tuesday) (Wednesday)`表示日期。

`(emergency-fire) (emergency-flood)`这两个事实由于第一部分都用了 emergency,使人一眼能看出其间有联系——两者都表示灾害。但由于规则不能只处理字段的前缀部分,所以不能识别这种相

似性。如果采用以下形式：

(emergency fire)

(emergency flood)

则第一个字段可用来描述其后字段的关系。这样的第一个字段称为关系名(relation name),其后字段用来规定事实的特定的值,即灾害的类型。一般说来,这可以用“事实样板”(fact template)来描述：

(emergency <type>)

更一般地,可以用一个以上样板来表示一组事实。例如,对以下事实：

(action activate fire-alarm)

(action notify fire-department)

(action shutdown-electrical-power)

可以用下列三个样板来表示：

(action activate <device>)

(action notify <group>)

(action <specific-action>)

用一个样板还可以表示一个以上的字段。例如：

(computer-components CPU disk-drive terminal)

可以用以下样板描述：

(components <<component-list>>)

其中<<component-list>>表示零个或多个字段。

上述样板再稍微修改一下就可以表示更多的部件表。例如以下两个事实：

(components computer CPU disk-drive terminal)

(components car tires engine body gas-tank)

可以统一用以下样板描述：

(components <item> <<components-list>>)

当然,样板只是为了编制文档方便而建立的,它不能作为命令被

CLIPS 接受。

1—4 进入和退出 CLIPS

进入 CLIPS 要在安装它们的机器上发出适当的运行命令。CLIPS 的提示符形式如下：

```
CLIPS>
```

这时，就可以直接向 CLIPS 输入命令。这叫做“最高级”(top-level)。

退出 CLIPS 用 exit 命令，即在 CLIPS 的提示符下键入(exit)。注意：命令必须用括号括住。

对于在 IBMPC 机(MS-DOS)上用驱动器 A 工作的情况，进入和退出 CLIPS 的命令序列如下。为了简化，每个命令后面以及提示响应输入后面的回车符号一律省去，请读者注意。

```
A>CLIPS
```

```
CLIPS(V4.20 4/29/88)
```

```
CLIPS> (exit)
```

```
A>
```

1—5 加入事实和撤消事实

CLIPS 所承认的事实都保存在事实表(fact-list)中。新的事实用 assert 命令加入事实表：

```
(assert <<<fact>>>)
```

作为例子，我们假定有一个监视工厂活动的专家系统。这个专家系统检查工厂可能发生的各种灾害，并采取相应的措施，如火灾。表示发生火警的事实可通过如下命令输入：

```
CLIPS> (assert (emergency fire))
```

为了检查火警这个事实是否已正确输入，可以用 facts 命令来

显示事实表中的事实：

```
CLIPS> (facts)
f-1 (emergency fire)
CLIPS>
```

(以后为了简化,在输入和响应后出现的 CLIPS 提示符也不再写出)。

响应中的 f-1 是 CLIPS 赋给事实的标识符 (fact identifier)。f 后的数字叫“事实索引号”(fact index)。索引号不一定顺序。

CLIPS 不接受重复输入的事实。用一个 assert 命令可以输入多个事实。例如：

```
CLIPS> (assert (emergency fire)
              (emergency flood))
```

在不需要检查事实表中的所有事实的情况下,可用如下形式的 facts 命令：

```
(facts [<start>[<end>[<maximum>]]])
```

其中的<start>、<end>、<maximum>均为正整数。分别表示要查阅事实的起始索引号,终止索引号和极大值。

撤消事实用 retract 命令：

```
(retract <<<fact-index>>>)
```

若给出的事实索引号在事实表中不存在,则给出出错信息。例如,事实表中仅有 f-1、f-2、f-3,但要撤消 f-4,则将有如下响应：

```
CLIPS> (retract 4)
Fact 4 does not exist
```

用一个 retract 命令可以一次同时撤消多个事实,如(retract 1 2 3)将同时撤消 f-1、f-2、f-3 三个事实。

1—6 规 则

现在我们来讨论为了使专家系统能够工作必须有的规则。