Turbo Prolog 和 Turbo C 的互调用技术

王清毅　（安徽省软件技术公司）

摘要:本文在阐述了 *Turbo Prolog* 和 *Turbo C* 两种程序设计语言的主要特点之后,具体给出了它们相互调用的方法、注意要点及环境设置,最后给出了两个例子。

一、引言

人工智能型语言 Prolog 作为基于陈述式原理的非过程式语言,以匹配、回溯、递归等为其主要特色,尤其适宜于人工智能领域问题的求解。迄今为止,Prolog 语言已有 Arity Prolog、Micro Prolog 等版本。目前在我国应用较广的当属 Borland 公司推出的 Turbo Prolog,它具有优良的用户界面和丰富的谓词,其交互集成式调试运行环境为程序编制提供了极大方便。

通用程序设计 C 语言是一种结构化、模块化的过程式语言,广泛应用于系统程序和应用程序设计。Turbo C 作为 Borland 公司的又一系列产品,提供了优良的集成开发环境和丰富的数值运算和图形函数据库,具有浮点运算和编译速度快、内存分配快等优点。

我们知道,每种程序设计语言都有其特色和优点,但同时也存在着缺陷,而这些缺陷,又往往能被其它程序设计语言所弥补。

Turbo Prolog 长于逻辑推理,但由于本身语法约定和结构的限制,在数值计算和图形图象处理方面,显得不足;而 Turbo C 又恰好长于数值计算和图形图象处理,却不擅长逻辑推理。因此,在软件开发实践中,我们很自然地想到将二者结合在一起。具体表现在:在 Turbo Prolog 的人工智能应用中,将需进行大量复杂计算和图形图象处理的部分,分配给 Turbo C 完成,而在 Turbo C 的应用中,将需进行逻辑推理的部分,分配给 Turbo Prolog 完成。这就是 Turbo Prolog 和 Turbo C 的互调用。在大型的应用程序设计中,这显得十分必要。

二、Turbo Prolog 调用 Turbo C

Turbo Prolog 中的调用谓词要和 Turbo C 中的被调用函数同名,调用谓词的参数个数要与被调用的 C 函数参数个数相同,调用谓词的流模式数应与被调用的 C 函数数一致,C 函数本身不返回值,其函数原型定义为 Void,函数中返回参数为指针型,Turbo Prolotg 中的调用谓词的返回值由相应的谓词变量返回。调用谓词在全局谓词段说明,在专家系统设计中,需求出某一推理的可信度,将此过程分配给 Turbo C,则相应说明部分为:

```
Turbo Prolog 中:
.....
.....
global prediates
get__cf(data__type1,data__type2,...,data__typen)
--(i,i,...,0) language C
.....
.....
clauses
.....
.....
get__cf(const1,const2,...,Credit__factor)
.....
.....
Turbo C 中:
.....
.....
Void get__cf__o( data__type parm1, data__type parm2, . . . ,
data__ type *
credit__factor)
{
函数体
}
```

要注意的是,对于 Turbo C 来说,其 printf、puts 等和屏幕有关的函数,在与 TurboProlog 连接时不起作用,而代之以 zwf(Format string,Parm1,Parm2,...), 其中 Formatstring 为输出格式字符串,Parm1,Parm2,...,为输出参数。

三、Turbo C 调用 Turbo Prolog

Turbo C 以函数形式调用 Turbo Prolog 谓词,调用

函数名当然要和 Turbo Prolog 中被调用谓词名相应。Turbo Prolog 的被调用谓词在全局谓词段说明,形式为:

global predicates

predicate__called(data__ytpe1,data__type2,...,data__typen)

--(流模式)language C as"C__function"

目标段为:goal

prolog__goal__name

要注意:目标也应在全局谓词段说明,即: prolog__goal__name language C as"main",显然,Turbo C 的函数参数要和 Turbo Prolog 被调用谓词变量相对应。

四、环境设置

在上述两种调用中,当把 Turbo C 源程序编译成.OBJ 文件时,需设置如下环境:

O / C / Model、Large

O / C / Optimiation / Jump Optimization ... On

O / C / Code Generation / Generate Underbars ... Off

O / C / Optimization / Use Register Varibles ... Off

最后,采用如下格式,将.OBT 文件和.LIB 文件连接成.EXE 文件,即:

TLINK objfile,mapfile,libfile libfiles

其中,objfiles 为 init prolog 目标文件、C 目标文件、prolog 主符号表文件(.sym)

exefile 为指定的输出,EXE 文件,mapfile 为指定想使用的 MAP 文件名,可缺省(以空格代替),libfiles 为所用到的库,顺序为:

用户定义库+Prolog.Lib+Emu.Lib+Mathl. Lib+Cl. Lib。

本文讨论 Turbo Prolog 和 Turbo C 均为 2.0 版。

五、结束语

Turbo Prolog 和 Turbo C 的互调用,体现了非过程式程序设计思想和过程式程序设计思想的结合,一方面在 Turbo Prolog 的人工智能应用中可充分利用 Turbo C 成熟的算法和函数,另一面又可将人工智能技术溶进 Turbo C 的应用中以实现智能化,从而体现了事实世界和数据世界的统一。

附例:

1.Turbo Prolog 调用 Turbo C

```
global predicates
sum(integer,integer,integer)-(i,io)language C
goal
sum(10,20,x),
write("sum=",X).
 * * * * * *
extern void zwf(char * format,...);
void sum__o(int parm1,int parm2,int * pt)
{ * pt=parm1 * parm2;
zwf("parm1=%d,parm2=%d\n",
parm1,parm2);
}
```

2.Turbo C 调用 Turbo Prolog

```
void makewindow(int num,int attr1,int attr2,char * title,int row,int col,int hei,int wide);
void write__string(char * t);
void read__char(char * c);
void read__line(char * c);
void main()
{char dum;
char * Name;
makewindow(1,7,"How do you do ",2,2,15,70);
write__string("Hit ang beg");
read__char(& dum);
write__string("\n Enter your name:");
read__line (& Name);
wrile__string("\n Your name is:");
write__string(Name);
read__char(& dum);}
 * * * * * *
global predicates
prmakewindow(integer,integer,integer, string, integer, integer, integer,integer)
-(i,i,i,i,i,i,i,i)language C al"make window"
prwrite__string (string) - (i) language C as "write__string"
prread__char(char)- (o) language C as "read__char"
prread__line(string) - (o) language C as "read__line"
prmain language Cas "main"
goal
prmain
clauses
prmakewindow(Num,Attr1,Attr2,Prom,Row,Col,Hei,Wide)
:-make window(Num,Attr1,Attr2,Prom,Row,Col,Hei,Wide).
prwrite__string(S):-write(s).
prread__+char(C):-readchar (C).
prread__line(L):-readln(L).
```

参考文献:

*[1]*杨祥金 蔡庆生 <<人工智能>> 科学技术文献出版社重庆分社 *1988*

*[2]*杨冀宏等 <<用 *Prolog* 和 *Turbo Prolog* 语言开发专家系统>>航空工业出版社 *1990*

[3]Herbert Schildt<<*C* 语言大全>>电子工业出版社 *1990*