

# 人工智能技术基础 实验指导书

人工智能课程组编写

北京交通大学电子信息工程学院

2019 年 3 月

## 实验一 PROLOG 语言编程练习

### 1 目的与要求

实验目的：

加强学生对逻辑程序运行机能的理解，使学生掌握 PROLOG 语言的特点、熟悉其编程环境，同时为后面的人工智能程序设计做好准备。

实验要求：

- (1) 程序自选，但必须是描述某种逻辑关系的小程序。
- (2) 跟踪程序的运行过程，理解逻辑程序的特点。
- (3) 对原程序可作适当修改，以便熟悉程序的编辑、编译和调试过程。

### 2 实验内容或题目

实验题目：PROLOG 语言编程练习

实验内容：

在 Turbo PROLOG 或 Visual Prolog 集成环境下调试运行简单的 PROLOG 程序，如描述亲属关系的 PROLOG 程序或其他小型演绎数据库程序等。

### 3 实验步骤与源程序

实验步骤：

- (1) 启动 Windows 操作环境。
- (2) 下载、安装 Turbo PROLOG，了解它的运行模式及功能。
- (3) 打开文件目录，执行 prolog 应用程序，启动 Turbo prolog，并按空格键 (SPACE) 进入集成开发环境。
- (4) 选择 Setup 项，打开下拉菜单，选择 Directories 项，进行工作目录修改，按 Esc 键退出，选择 Save Configuration 项，保存修改。
- (5) 选择 Files 项，打开下拉菜单，选择 New file 项，进入源程序输入和编辑，或选择 Load 项，选择要打开的示例程序，再选择 Edit 项，可以进行编辑源程序。
- (6) 编辑之后，可以选择 Run 项，执行程序，可以在 Dialog 窗口进行询问，即外部目标的执行，查看程序运行结果，分析程序之功能。
- (7) 仿前例，可以选择其他程序并运行，分析程序功能。
- (8) 退出，选择 Quit 项，可以退出 Turbo Prolog 程序，返回到 Windows 环境。

源程序：

验证题 1：

```
domains
    d=integer
predicates
```

```

    not_(D, D)
    and_(D, D, D)
    or_(D, D, D)
    xor(D, D, D)
clauses
    not_(1, 0).
    not_(0, 1).
    and_(0, 0, 0).
    and_(0, 1, 0).
    and_(1, 0, 0).
    and_(1, 1, 1).
    or_(0, 0, 0).
    or_(0, 1, 1).
    or_(1, 0, 1).
    or_(1, 1, 1).
xor(Input1, Input2, Output):-
    not_(Input1, N1), /* (1, 0) */
    not_(Input2, N2), /* (0, 1) */
    and_(Input1, N2, N3),
    and_(Input2, N1, N4),
    or_(N3, N4, Output).

```

#### 验证题 2:

```

predicates
    student(integer, string, real)
    grade
goal
    grade.
clauses
    student(1, "zhang", 90.2).
    student(2, "li", 95.5).
    student(3, "wang", 96.4).
    grade:-write("Please input name:"), readln(Name),
        student(_, Name, Score),
        nl, write(Name, " grade is ", Score).
    grade:-write("Sorry, the student cannot find!").

```

#### 验证题 3:

```

domains
    n, f=integer
predicates
    factorial(n, f)
goal
    readint(I),
    factorial(I, F),

```

```

    write(I, "!=" , F).
clauses
    factorial(1,1).
    factorial(N,Res):-
        N>0,
        N1=N-1,
        factorial(N1,FacN1),
        Res=N*FacN1.

```

## 验证题 4:

```

domains
    s=symbol
predicates
    p(s) p1(s) p2(s) p3(s) p4(s) p5(s,s) p11(s) p12(s) p31(s)
goal
    p(X),write("the x is ",X).
clauses
    p(a1):-p1(b),p2(c).
    p(a2):-p1(b),p3(d),p4(e).
    p(a3):-p1(b),p5(f,g).
    p1(b):-p11(b1),p12(b2).
    p3(d):-p31(d1).
    p2(c1).
    p4(e1).
    p5(f,g).
    p11(b1).
    p12(b2).
    p31(d11).

```

## 验证题 5:

```

domains
    name=symbol
    age =integer
predicates
    player(name,age)
    match(name,name)

clauses

    player(peter, 9).
    player(paul, 10).
    player(chris, 9).
    player(susan, 9).

    match(X,Y):- player(X,9), player(Y,9), X<>Y.

```

```

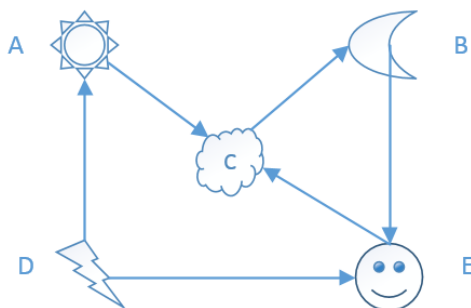
match(X,Y):- !, player(X,9), player(Y,9), X<>Y.
match(X,Y):- player(X,9), !, player(Y,9), X<>Y.
match(X,Y):- player(X,9), player(Y,9), !, X<>Y.
match(X,Y):- player(X,9), player(Y,9), X<>Y, !.

```

在外部 Goal: match(X,Y)条件下，“分别”采用以上各 match 谓词，观察对应的执行结果。

附加题：

编写 prolog 程序，寻找下图某个节点出发所有路径（提示：利用递归思想，参考 ppt2.2.7）。



### 实验中易出错的地方

- (1) PROLOG 程序中大小写字母分别代表不同的含义，一般大写字母开头表示变量，小写字母开头表示常量，在写程序时要注意区分。
- (2) 子句段中规则和事实可连续排列在一起，但同一谓词名的事实或规则必须集中排列在一起。
- (3) Goal 段可在程序运行时临时给出，但必须是在开发环境中运行程序时方可给出。若要生成一个独立的可执行文件，则在程序中必须包含 goal 段。

## 4. 实验报告要求

实验报告应至少包含如下内容：

- 实验题目
- 实验目的
- 实验环境
- 实验内容
- 实验结果（要求附上源程序）
- 实验中出现的問題
- 对问题的解决方案
- 实验总结