基本编程

加载文件

```
['D:/GK.2010.83a.pl'].
```

退出

```
halt.
```

注释,结束

```
/* print hello world */
go :- write("hello world").
```

注释:/* */,或行注释:%,结束用“.”来表示。

运行结果

```
?- go.
hello world
true.
```

术语

```
dog(jack). dog(cathy).
cat(rose). cat(mike).
animal(X):-dog(X).
```

语句(**clauses**)包括事实(**facts**)或规则(**rules**)两类。

例如：dog(fido)是事实, animal(X):-dog(X)是规则。

在dog(fido)中，dog是谓词(**predicate**), fido是实参(argument).

在animal(X):-dog(X)中，X是变量(variable).

Prolog通过问问题(query)来执行。

```
?- animal(jack).
true.
```

“?-”后面就是目标问题 (**goal**). 这里prolog会回答两种结果：true或false.

也可以带着变量问

```
?- animal(X).
X = jack ;
X = cathy.
```

变量用首字母大写的串来表示，事实首字母必须小写。

Prolog找到一个答案后，会回头重新找下一个答案, 多个答案用分号隔开。

有些内置的谓词可直接用作goal.

```
?- listing(dog).
dog(jack).
dog(cathy).

true.
```

目标中可以含多个询问，用“,”隔开表示“与”的关系。

```
?- dog(X),cat(Y).
X = jack,
Y = rose ;
X = jack,
Y = mike ;
X = cathy,
Y = rose ;
X = cathy,
Y = mike.
```

数据类型

数: 234, 23.4

字符串: ‘Hello World’, nl是回车换行

函数: likes(jack,rose)

变量: X,Y,Z

List: [a,[b,c,d],e,f]

有一些内置的常数，如： pi, e, inf.

操作符

操作符的定义

```
 :- op(500,xfx,'is_parent').
```

尽管一般是这么使用

```
 m is_parent t.
```

但也等同于谓词:

```
 ?- is_parent(m,t).
 true.
```

常见的算术操作符有:

+,-,*,/, //, ^,abs(X),max(X,Y),sqrt(X).

```
?- 1 =:= sin(pi/2).
true.
```

常见的逻辑操作符有:not, “;”, “,”

输入与输出

输入:

```
sayhi :-
    write('What is your name ?'),nl,
    read(X),
    write('Hi '),tab(2), write(X).
```

运行结果是

```
?- sayhi.
What is your name ?
|: 'Hui Li'.
Hi   Hui Li
true.
```

又如

```
square :-
    write('Please input a Num: '),
    read(Num),
    do(Num).
do(stop) :- !.
do(Num) :-
    S is Num * Num,
    write('Square of
'),write(Num),write(': '),write(S),nl,
square.
```

运行结果是

```
?- square.
Please input a Num: 5.
Square of 5: 25
Please input a Num: |: stop.

true.
```

输出: write

输出ASCII码:

```
?- put(37),nl.
%
true.
```

List与字符串操作

读取每一项

```
writeall([]).
writeall([A|L]):- write(A),nl,writeall(L).
?- writeall([i,'love','Beijing',[tian,an,men]]).
i
love
Beijing
[tian,an,men]
true.
```

内置谓词

```
?- append([a,b,c],[1,2,3],L).
L = [a, b, c, 1, 2, 3].

?- member(a,[a,b,c]).
true.
```

```
?- length([a,b,c],X).
X = 3.

?- reverse([a,b,c],X).
X = [c, b, a].
```

字符串与List的转换

```
?- name('helloworld',L).
L = [104, 101, 108, 108, 111,
119, 111, 114, 108|...].


?-
name(X,[104,101,108,108,111,119
,111,114,108,100]).X =
helloworld.
```

借助List操作实现字符串操作

```
joinNew(Str1,Str2,Str3):-
    name(Str1,L1),name(Str2,L2),
    append(L1,L2,Result),
    name(Str3,Result).
?- joinNew('hello',' world',X).
X = 'hello world'.
```