

Contents

Basics	1
Operators	2
Atoms	2
Variables	2
Arity	2
Functor	2
Cuts	3
Responses	3
Arithmetic	3
Comparison	3
Summary	4
Past Papers	4

Basics

Fact:

- `happy(yolanda).`
- `party.`

Rule:

- `listens2music(yolanda):- happy(yolanda).`

Predicates (facts that are always true):

- `happy.`
- `listens2music.`
- `playsAirGuitar.`

Operators

- Implication: :-
- Conjunction: ,
- Disjunction: ;

Atoms

Starts with a lower case letter

- `butch.`
- `playGuitar.`

Enclosed in single quotes

- `'Vincent'`
- `'Five dollar shake'`

Variables

Starts with an upper case letter

- `Var`

Arity

The numbers of arguments a complex term has is called its arity

- `woman(mia)` has arity 1
- `loves(mia, vincent)` has arity 2
- `hide(X, father(butch))` has arity 1

Functor

- Things like the `playsAirGuitar` above. Functors must be atoms. The arity of a functor is the number of arguments it takes.
- Functors can be defined with the same name and different arities, though Prolog makes no assumption about whether those functors are related.

Cuts

Green cuts:

- Cuts that do not change the meaning of a predicate
- Adds efficiency

Red cuts:

- Everything else
- Programs containing red cuts
 - Are not fully declarative
 - Can be hard to read
 - Can lead to subtle programming mistakes

Responses

Arithmetic

- `+`, `-`, `/` and `*` do not carry out any arithmetic
 - Functors with arity 2
- To force Prolog to actually evaluate arithmetic expressions, we have to use `is`
 - We are free to use variables on the right hand side of the `is` predicate

Comparison

- `==/2` does not instantiate variables
 - Behaves differently from `=/2`
 - `\==/2` succeeds where `==/2` fails

```
?- a==a.  
true  
?- a==b  
false  
?- a=='a'  
true  
?- a==X  
X=_443  
false
```

Summary

=	Unification predicate
=	Negation of unification predicate
==	Identity predicate
==	Negation of identity predicate
:=	Arithmetic equality predicate
==	Negation of arithmetic equality predicate

Past Papers

```
?- X=1.  
X = 1.  
  
?- X==Y.  
false.  
  
?- 0+1=1+0.  
false.  
  
?- 0+1 := 1+0.  
true.  
  
?- 3+2=5.  
false.  
  
?- 3+2=X.  
X = 3+2.  
  
?- 3+2=2+3.  
false.  
  
?- 3+2 is X.  
ERROR: Arguments are not sufficiently instantiated  
ERROR: In:  
ERROR:    [8] 3+2 is _7292  
ERROR:    [7] <user>  
?- X is 3+2.  
X = 5.  
  
?- X \= f(X).  
false.
```

```

?- f(X) \= g(Y).
true.

?- [1|[2,3]] = (1,(2,[3])).
false.

?- [a|[b,c]]=[a,[b,c]].
false.

?- [a,b|[c]]=[a|[b,c]].
true.

?- [[a]]=[a|[]].
true.

?- X==f(X).
false.

?- X>0.
ERROR: Arguments are not sufficiently instantiated
ERROR: In:
ERROR:      [8] _6278>0
ERROR:      [7] <user>

?- findall(X, X \= 1, L).
L = [].

```