

Prolog - Sintaxis, Operadores y Herramientas

Sintaxis

Concepto y/o explicación	Sintaxis en Prolog
Comentarios	% (símbolo de porcentaje) /* comentario en varias líneas en el código */
Finalización de expresiones	. (el punto)
Términos: formados por átomos, números y variables	
Átomos: <ul style="list-style-type: none">Una secuencia de caracteres compuesta por letras en mayúsculas, letras minúsculas, dígitos y el carácter subrayado, que comienza con una letra minúscula.Una secuencia arbitraria de caracteres encerrados en comillas simples.	perro(canela). perro('Canela').
Variables: <ul style="list-style-type: none">Deben empezar siempre con mayúscula o también pueden empezar con el carácter de subrayado(underscore).En el caso de variables sin nombres las mismas deben utilizar el guion bajo	padre(A, B). padre(_elPadre,B). perro(_).

Tipos de cláusulas

Básicamente las cláusulas son los hechos y las reglas. También podemos considerar el objetivo como un tipo de cláusula particular.

Cláusulas	
Una <u>regla</u> es así: Cabeza :- Cuerpo. Un <u>hecho</u> puede decirse que es: Cabeza :- true. El <u>objetivo</u> es una cláusula sin cabeza, es la consulta ?- Cuerpo.	% hechos gato(michi). gato(mini). perro(scooby). % reglas animal(X):-gato(X). animal(X):-perro(X).

Algunas consultas que podemos hacer:

```
?- animal(michi).
true ;
false.

?- animal(X).
X = michi ;
X = mini ;
X = scooby.

?- perro(scooby).
true.

?- perro(X),gato(Y).
X = scooby,
Y = michi ;
X = scooby,
Y = mini.

?- 
```

Operadores

Tienen particularidades como que no existe el operador de asignación (porque no se usan variables en el sentido de los lenguajes imperativos)

Lógicos	
AND (conjunción y)	, (coma)
OR (disyunción o) [mejor usar not]	; (punto y coma)
NOT (negación lógica)	not
IF (implicación SI)	:- (dos puntos y guión)
Aritméticos	
Suma, resta, división y multiplicación	+ - / *
División entera	//
Módulo	mod
Potencia	^
Relacionales	
Mayor, menor, mayor o igual, menor o igual	> < >= <=

De igualdad	
Igualdad lógica	=
Desigualdad lógica	\=
Exactamente igual (compara términos sin evaluar expresiones)	= =
Igualdad aritmética	= : =
Desigualdad aritmética	= \ =
Operador de Asignación	NO EXISTE
<p><i>NO EXISTE algo como la asignación de los lenguajes imperativos. Lo que existe es la Unificación.</i></p> <p>Puede decirse que una variable es una incógnita que va tomando distintos valores debido a los mecanismos de unificación y sustitución, como han visto en la Teoría.</p> <p>En Prolog = significa Unir con en lugar de es igual a.</p>	
Operador is (evaluador)	
<p>Se utiliza para:</p> <ul style="list-style-type: none"> ➤ evaluar las expresiones aritméticas y funciones. ➤ evalúa la parte de la derecha y unifica a la parte izquierda. ➤ si no se usa el is, las expresiones se mantienen en su forma original 	<p>Ejemplo:</p> <p>X=3+3 se evalúa como X=3+3 X is 3+3 se evalúa X=6</p>

Unificación y Sustitución

```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
reydejuda@pegasus:~$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('/home/reydejuda/prolog/animales.pl').
true.

?- animal domestico(X).
X = 'Canela' ;
X = 'Melchor' ;
X = 'Dulce' ;
X = 'Sol' ;
X = 'Michi' ;
X = 'Kitty'.

?- animal salvaje(X).
X = lince1 ;
X = lince2 ;
X = leon1 ;
X = leon2.

?- 

```

usamos el ; punto y coma, hasta que Prolog nos muestra el último matcheo.

```
?- animal_salvaje(X).  
X = lince1 ;  
X = lince2 ;  
X = leon1 ;  
X = leon2.
```

```
?- animal_domestico(canela).  
false.
```

no puede unificar canela con 'Canela'

```
?- animal_domestico(Canela).  
Canela = 'Canela' ;  
Canela = 'Melchor' ;  
Canela = 'Dulce' ;  
Canela = 'Sol' ;  
Canela = 'Michi' ;  
Canela = 'Kitty'.
```

estamos usando Canela como variable

```
?- animal_domestico('Canela').  
true .
```

puede unificar

```
?- []
```

```
?- animal_domestico('lince1').  
false.
```

```
?- animal_domestico('lince 1').  
false.
```

```
?- animal_domestico(_lince).  
_lince = 'Canela' ;  
_lince = 'Melchor' ;  
_lince = 'Dulce' ;  
_lince = 'Sol' ;  
_lince = 'Michi' ;  
_lince = 'Kitty'.
```

variable _lince

```
?- animal_domestico('lince1').  
false.
```

átomos

```
?- animal_salvaje('lince1').  
true .
```

```
?- 'lince1' = lince1.  
true.
```

```
?- 'lince1' = Lince1.  
Lince1 = lince1.
```

```
?- 2 = 2.  
true.
```

numeros

```
?- '2' = 2.  
false.
```

no unifica átomo con número

```
?- []
```

```
?- 2=2.  
true.  
  
?- '2' = 2.  
false.  
  
?- 8 = 8.  
true.  
  
?- 8 := 8.  
true.  
  
?- 8 := 9.  
false.  
  
?- 8 \= 9.  
true.  
  
?- 8 \= 9.  
true.  
  
?- not(8 \= 9).  
false.  
  
?- 4 + 2 = 1 + 5.  
false.  
  
?- X = 1 + 5.  
X = 1+5.  
  
?- X is 1 + 5.  
X = 6.  
  
?- Y is 4 + 2.  
Y = 6.
```

Para ampliar estos temas en la web de swi-prolog

Libro online:

<https://lpn.swi-prolog.org/lpnpag.php?pageid=online>

Curso con diapositivas:

<https://cs.union.edu/~striegkn/courses/essli04prolog/>

Herramientas:

- swi-prolog (Linux, Mac y Windows): <https://www.swi-prolog.org/Download.html>
- SWISH (herramienta online de swi-prolog): <https://swish.swi-prolog.org/>