

PROLOG

SINTAXIS Y TERMINOLOGÍA DE PROLOG

- Los predicados y constantes empiezan con minúscula
- Las variables con mayúscula
- Son variables lógicas, no memoria que puede cambiar su contenido.
- Si una unificación vincula una variable con un valor, eso impide vínculos con otro valor.
- La variable anónima es: _

Términos: no admiten funciones como argumentos:

Expresiones: se pueden formar con operadores comunes pero son muy especiales. Prolog no es un lenguaje de expresiones sino de pred.

Reglas: Cada línea de Prolog es la definición de una regla.

El $:-$ es la implicación hacia la izda. (\Leftarrow).

El ";" que separa antecedentes es un and (\wedge).

La regla termina siempre con un punto.

En el consecuente y antecedentes pueden aparecer variables.

Se sobreentiende que todas las variables están cuantificadas universalmente en cada regla.

Puede haber reglas sin antecedentes: $es_divisor(X, X)$.

Las reglas que se admiten son cláusulas de Horn.

Recuerdo: HORN

Una cláusula de Horn es un conjunto no vacío de literales en el que no hay más de un literal positivo.

Cláusula de Horn:

- restricción: cláusula sin ningún literal positivo
- hecho: un único literal positivo
- regla: una cláusula con un único literal positivo junto con, al menos, un literal negativo.

Se pueden escribir como una implicación (con reglas)

LISTAS EN PROLOG

- Se forman con corchetes y comas, o también con "|".

$$[a,b] \equiv [a|[b]]$$

- Todas estas listas son equivalentes

$$[a,b,c] \equiv [a|[b,c]] \equiv [a|[b|c]] \equiv [a|[b|[c|[]]]]$$

- Pero:

$$[a,b,c] \neq [a,[b,c]] \text{ (lista de dos elems., uno de ellos es una lista)}$$

$$[a,b,c] \neq [a|[b|c]] \text{ (lista impropia)}$$

- Las listas se pueden anidar
- Las listas se pueden usar como argumentos de predicados:
- Correspondencias listas prolog \longleftrightarrow intuitivamente:

$$[a,b] \sim (a,b)$$

$$[a|b] \sim (\underbrace{a}_{\text{first}}, \underbrace{b}_{\text{rest}}) \sim (\underbrace{a}_{\bar{a}}, \underbrace{b,c}_{\bar{b}}) \stackrel{\text{disp}}{\sim} (a,b)$$

$$[a,[b,c]] \sim (a, (b,c))$$

$$[[a,b]|c] \sim ((a,b), c)$$

$$[] \sim ()$$

$$[c|[]] = [c] \sim (c)$$

$$[b|[c]] = [b,c] \sim (b,c)$$

$$[a|[b,c]] = [a,b,c] \sim (a,b,c)$$

UNIFICACIÓN EN PROLOG

- Variables para unificación: simétrica ($[X, _, Y]$)
- $p([A|B])$ unifica con $p([1,2,3])$ si $A=1$ $B=[2,3]$ ("vínculos")
- $p([A|B])$ NO unifica con $q([1,2,3])$
- $p([A|_])$ unifica con $p([1,2,3])$ si $A=1$
- $p([_ |B])$ unifica con $p([1,2,3])$ si $B=[2,3]$
- $p([_ _])$ NO unifica con $p([1,2,3])$
- $p([_ 1 _])$ unifica con $p([1,2,3])$
- $p([_ 1 _])$ SÍ unifica con $p([1])$

OPERADORES

• El operador "=" unifica la expresión a su izquierda con la de la derecha, introduciendo vínculos nuevos. Si es imposible \rightarrow return Fallo.

Los operadores ">", "<", ">=", "<=", "\=" dan fallo si no se cumplen. No introducen vínculos nuevos.

El operador "is" realiza dos cosas:

- \rightarrow 1) evalúa la expresión de la derecha
- \rightarrow 2) unifica el resultado de dicha evaluación con lo que hay a su izquierda.

Si la expresión de la izquierda del "is" contiene alguna variable no vinculada, dará error de ejecución. Ejemplos de operaciones a la derecha del "is": $M+N$, $X-Y$, $A*B$, C/D , $E \bmod F$

