



Tecnologías de Programación

Lenguaje de Programación Lógica

ProLog



Prolog

Lenguajes de Programación Lógica

Prolog

CLP - Constraint Logic Programming

- Basado en ProLog
- Capacidad adicional de constraints



Prolog

Utilidad Práctica

Generación de CGI's.

Acceso a bases de datos desde páginas Web.

Paralelización automática de programas.

Programación distribuida y multiagente.

Sistemas expertos e inteligencia artificial.

Validación automática de programas.

Procesamiento de lenguaje natural.

Prototipado rápido de aplicaciones.

Bases de datos deductivas.

Interfacing con otros lenguajes como Java y Tcl/Tk.

... y la lista sigue ...



Prolog

Entorno de Desarrollo

Compilador

Transforma el código fuente en código de byte

No existe un standard al respecto

Intérprete

Ejecuta el código de byte

Shells

Permiten probar los programas, depurarlos, etc.

Compatibilidad ISO-Prolog

El standard ISO dicta las normas de sintaxis y las bibliotecas básicas que se deben ofrecer



Prolog

Sintaxis

Comentarios

Se escriben comenzando la línea con un símbolo de porcentaje

Ejemplo: % Esto es un comentario

Variables lógicas

Comienzan siempre por mayúscula o subrayado

Ejemplos: Variable, _hola, _

El carácter '_' representa una variable anónima

Dos apariciones de la variable anónima (__) no hacen referencia a la misma variable



Prolog

Sintaxis

Términos

Datos que maneja un programa

se compone de un functor seguido de cero a N argumentos

Ejemplos:

`termino_cero_ario`

`1237878837385345.187823787872344434`

`t(1)`

`f(Variable)`

`muchos_argumentos(_,_,_,Variable,232,f,g,a)`

`terminos_anidados(f(g), h(i,j(7)), p(a(b)), j(1,3,2,_))`

`+(3,4)`



Prolog

Sintaxis

Operadores

Son funtores declarados como tales (predefinidos o programados)

Sirven para escribir en forma infija algunas operaciones

Tienen asociada una prioridad (precedencia)

Ejemplo:

-5 es en realidad $-(5)$

$a+b*c$ es en realidad el término $+(a,*(b,c))$

Nota: no es $*(a,+(b,c))$ porque $*$ tiene mayor prioridad que $+$



Prolog

Dando valor a las variables

Unificación

Mecanismo mediante el cuál las variables lógicas toman valor

El valor que puede tomar una variable consiste en cualquier término

Las variables que no tienen valor se denominan libres

Las variables que tienen valor asignado se denominan ligadas

Dos términos unifican cuando existe una posible ligadura (asignación de valor) de las variables tal que ambos términos son idénticos sustituyendo las variables por dichos valores.

Por ejemplo:

$a(X,3)$ y $a(4,Z)$ unifican dando los siguientes valores a las variables: X vale 4, Z vale 3



Prolog

Dando valor a las variables

Unificación

$K(Z, Z)$ y $k(4, H)$: por culpa del primer argumento, Z se liga al valor 4

$K(Z, Z)$ y $k(4, 3)$: no unifican

$K(_, _)$ y $k(3, 4)$: sí unifican ya que $_$ es anónima

$a(b(j, K), c(X))$ y $a(b(W, c(X)), c(W))$: unifican???

$a(b(\textcolor{red}{j}, K), c(X))$ y $a(b(\textcolor{red}{W}, c(X)), c(\textcolor{red}{W}))$

$a(b(j, K), \textcolor{red}{c(X)})$ y $a(b(j, \textcolor{red}{c(X)}), \textcolor{red}{c(j)})$

$a(b(j, \textcolor{red}{K}), c(j))$ y $a(b(j, \textcolor{red}{c(j)}), c(j))$

$a(b(j, c(j)), c(j))$ y $a(b(j, c(j)), c(j))$

Sí unifican



Prolog

Predicados y Objetivos

Predicados

Son elementos “ejecutables” en prolog

Una llamada concreta a un predicado en prolog se denomina objetivo

Todo objetivo tiene como resultado un valor de verdad:
True o False

Cuando un objetivo tiene éxito las variables libres que aparecen en los argumentos pueden quedar ligadas. Estos son los valores que hacen cierto el predicado. Si el predicado falla, no ocurren ligaduras en las variables libres.



Prolog

Predicados y Objetivos

Predicados

Ejemplo:

`son_hermanos('Juan', 'Maria').`

`son_hermanos('Juan', X).`

`son_hermanos(Y, X).`



Prolog

Predicados y Objetivos

Secuencias de Objetivos

Los objetivos se pueden combinar mediante conectivas: conjunción, disyunción y negación.

De estas la más usada es la conjunción y se utiliza la coma “,” para denotarlo

Ejemplo:

`edad(luis, Y), edad(juan, Z), X > Z`



Prolog

Predicados y Objetivos

A tener en cuenta

- Los objetivos se ejecutan secuencialmente en orden de lectura

- Si un objetivo falla el resto ya no se ejecutan

- Si un objetivo tiene éxito, algunas o todas sus variables quedan ligadas, y por tanto, dejan de ser variables libres para el resto de objetivos en la secuencia

- Si todos los objetivos tienen éxito, la conjunción tiene éxito y mantiene las ligaduras de los objetivos que la componen.



Prolog

Predicados y Objetivos

Volviendo al ejemplo

$\text{edad}(\text{luis}, Y), \text{edad}(\text{juan}, Z), X > Z$

Supongamos edad de Luis = 32 y de Juan = 25

Por el primer objetivo, "Y" que antes estaba libre ahora queda ligada al valor 32 y el objetivo tiene éxito

Por el segundo objetivo, la variable "Z" hasta ahora libre, queda ligada al valor 25 y el objetivo tiene éxito

Al evaluar el tercer objetivo, sus variables ya no están libres.
Como el valor de "Y" es mayor que el de "Z" la comparación tiene éxito

Como todos los objetivos han tenido éxito, la conjunción tiene éxito, y deja las variables "Y" y "Z" ligadas a los valores 32 y 25 respectivamente..



Prolog

Predicados y Objetivos

Backtracking

Se usa cuando los objetivos tienen más de una solución

Esto consiste en recordar los momentos de la ejecución donde un objetivo tenía varias soluciones (puntos de elección), para posteriormente dar marcha atrás y seguir la ejecución utilizando otra solución como alternativa.

Cada vez que se ejecuta un punto de elección, se elimina el mismo para no encontrarlo en nuevos backtrackings

Ejemplo: Supongamos que disponemos de dos predicados $p/1$ y $q/1$ que tienen las siguientes soluciones:

$p(1)$, $p(2)$, $q(2)$ tienen éxito

No hay más soluciones que éstas



Prolog

Predicados y Objetivos

Backtracking

Ejecuta $p(X)$ y sabe que existen dos soluciones, se anotan dos puntos de elección

Ejecutamos $p(X)$ usando el primer punto de elección (el cual es descartado). El objetivo tiene éxito y X se liga a 1

Se ejecuta $q(X)$ que solamente tiene un punto de elección (el cual se descarta), pero la variable ya no está libre. El predicado falla porque $p(1)$ no es solución

Comienza el backtracking, recorriendo los objetivos en orden inverso hasta encontrar un punto de elección anotado

Encontramos el objetivo $p(X)$. Se deshace la ligadura de la variable X , es decir, X vuelve a estar libre



Prolog

Predicados y Objetivos

Backtracking

Se encuentra un punto de elección. La ejecución sigue de nuevo hacia adelante

Ejecutamos de nuevo $p(X)$, pero esta vez se liga la variable X al valor 2. El punto de elección se elimina en el proceso

Hay que ejecutar $q(X)$ de la misma forma que con el primer punto de elección, solo que ahora X está ligada al valor 2. El objetivo tiene éxito esta vez

La conjunción tiene éxito manteniendo la ligadura de la variable X al valor 2



Prolog

Predicados y Objetivos

Predicados predefinidos

Existen muchos y están disponibles en todo momento

El más importante es la igualdad

`=/2` tiene éxito si sus dos argumentos unifican entre sí

Algunos predicados aritméticos

`</2` menor que

`>/2` mayor que

`=</2` menor o igual que

`>=/2` mayor o igual que

`:=/2` igualdad aritmética

`=\=/2` desigualdad aritmética



Prolog

Cláusulas

Los predicados se escriben como un conjunto de cláusulas

clausula1

clausula2

...

clausulaN

Las cláusulas tienen la siguiente forma:

cabeza :-

objetivo1,

objetivo2,

...,

objetivoN.



Prolog

Cláusulas

Todo gira en torno al operador ":-". Lo que aparece a la izquierda se denomina cabeza y la secuencia de objetivos que aparece a la derecha se denomina cuerpo

Cuando se invoca un objetivo, el mismo se unifica con las cabezas de las cláusulas. Cada cláusula que unifique constituye un punto de elección

A continuación se ejecuta el cuerpo de la primera cláusula seleccionada. Para ello se mantienen las ligaduras que ocurrieron en el paso anterior y se ve con que cabeza unifica cada objetivo del cuerpo, volviendo a repetir el proceso

Si el cuerpo de la cláusula falla, se utiliza backtracking, si no hay más puntos de elección, el predicado falla



Prolog

Ejemplo

edad(juan, 30).

edad(pedro, 80).

es_anciano(Individuo) :- edad(Individuo, Valor),
Valor > 70.

nuestro objetivo será: es_anciano(juan).

Unificaciones:

es_viejo(juan) :-

edad(juan, Valor),
Valor > 60.

como edad(juan, Valor), solo puede unificarse a edad(juan, 30) y ésto no satisface el segundo objetivo del cuerpo, el predicado falla