

Prolog

Programación lógica

Javier Loureiro Pérez

Historia

Prolog es un lenguaje de programación ideado a principios de los años 70 por Alain Colmerauer y Philippe Rousse, al principio su idea no era crear un lenguaje de programación, si no crear un proyecto para el tratamiento de algoritmos de lenguajes naturales. Más tarde, Rousse persuadió a Robert Kowalski, creador del método de resolución SL para que colaborase en el proyecto, dando así lugar en 1971 a una primera versión preliminar de prolog y más tarde en 1972 una versión final.



Prolog

En 1997, Prolog pasó a ser un lenguaje estandarizado por ISO bajo el nombre de ISO/IEC 13211 y se unió a la lista de lenguajes ya estandarizados. ISO Prolog define lo que hasta entonces era la evolución más común del lenguaje Prolog original. Por lo general, la licencia de Prolog es gratuita, aunque existe alguna implementación de pago.

Normalmente se considera un lenguaje ligado a la investigación sobre Inteligencia Artificial, aunque hoy en día se utiliza también en campos diversos, incluyendo finanzas, defensa o ingeniería.

Programación lógica

Prolog es el principal lenguaje de la programación lógica, este paradigma gira en torno al qué y no al cómo, esto se trabaja estableciendo relaciones entre entidades de forma descriptiva, lo que obliga a los lenguajes a tener un alto nivel de abstracción.

Es un paradigma que está basado en hechos y reglas, los cuales utiliza para hallar una solución. Los hechos serían la “base de datos” y las reglas, que sirven para afirmar la veracidad de un hecho en base a otro.

Hechos

Los hechos son los mecanismos con los que los programadores dictaremos las verdades a Prolog. Sirven para marcar relaciones entre los objetos que se están representado. Se pueden dividir en:

- Propiedades: Se caracterizan por llevar un solo argumento expresando una característica de un objeto.

- padre(juan) Denota la propiedad de juan de ser padre

- Relaciones: Llevan más de un argumento, estableciendo así una relación entre do objetos.

- padre('juan', 'maría') Declara que juan es padre de maría.

Reglas

Se usa una regla cuando la verdad de un hecho depende de la verdad de otro hecho. Consiste en combinar hechos para darle el valor verdadero a un predicado. La sintaxis para una regla sería “Cabeza :- Cuerpo”, quedando cabeza dependiente de cuerpo. “La cabeza es verdad si el cuerpo es verdad”.

-hijode(A,B) :- padrede(B,A). -- A es hijo de B si B es padre de A

-abuelode(C,A) :- padrede(C,B), nietode(A,C). -- C es abuelo de A si B es padre de A o A es nieto de C.

masgrande(elefante,caballo)

masgrande(caballo,perro)

muchomasgrande(A,C) :- masgrande(A,B),masgrande(B,C)

Términos

Prolog no es un lenguaje tipado puesto que no existen declaraciones de datos como tal, los términos son los elementos que conforman el lenguaje en prolog.

-Átomo o functor: Son nombres de objetos, propiedades, o relaciones. Estos deben empezar en minúscula.

predicado(atomo).

nombre(luis).

-Números: Valores que solo pueden ser entero o reales para realizar después operaciones aritméticas, pueden llevar el signo.

predicado(número).

edad(30)

-Variables: son cadenas de letras, dígitos y el signo '_'. Estas empiezan con una letra mayúscula o el símbolo '_':

predicado(variable)

predicado(X).

Términos

-Estructuras: Estos son términos compuestos por otros términos, donde la sintaxis que se tiene es la siguiente:

predicado(termino1, termino2, ...).

edad('luis', 30). -- Estructura formada por dos argumentos

Estructuras

-Registros: Son agrupaciones de datos en los que los datos se almacenan en los argumentos del término.

```
persona(nombre, apellido1, apellido 2, edad).
```

-Árboles: Son estructuras de datos en las que relacionamos datos entre si.

```
arbol(dato1(dato2),(dato3(dato4, dato5))).
```

-Listas: Es una representación de un conjunto de elementos, las listas en Prolog se representan entre corchetes.

```
lista[dato1, dato2, dato3...]
```

Programas en Prolog

La extensión por defecto para un programa prolog es .pl, sin embargo esta extensión puede originar conflictos en sistemas con archivos de lenguaje Perl, que utilizan esta misma extensión. En estos casos, la alternativa más común es utilizar .pro o también .p.

Aquí predicamos que perro y gallina son mamífero ovíparo, y que palmera es un árbol y margarita una flor, esto serían los hechos.

```
mamífero(perro).
```

```
ovíparo(gallina).
```

```
arbol(palmera).
```

```
flor(margarita).
```

Programas en Prolog

Y aquí predicamos las reglas:

```
vegetal(X) :- arbol(X)
vegetal(X) :- flor(x)
animal(X) :- mamifero(X)
animal(X) :- oviparo(X)
```

Las consultas a prolog se formulan de la siguiente manera:

? -animal(perro).	?-vegetal(palmera)
true	true
?-vegetal(gallina)	?-animal(flora)
false	false

Librerías

Las librerías de Prolog contienen una serie de paquetes que son útiles en varias aplicaciones, para usar uno de estos paquetes, se ingresará usando “`use_module(library(Package)).`”. Algunas de las librerías más comunes son:

-aggregate: Proporciona un operador de consultas para bases de datos.

-fyle_systems: Da acceso a los archivos y directorios del sistema.

-assos: Implementa listas asociadas término a término.

-lists: Añade operaciones básicas en listas.

Librerías

-random: Proporciona un generador de números aleatorios.

-types: Analiza los tipos de datos.

-assos: Implementa listas asociadas término a término.

-objets: Proporciona un paquete para programación orientada a objetos

-between: Implementa algunos medios para generar números enteros.

Versiones de Prolog

Uno de los lenguajes que más éxito ha tenido en programación lógica es Prolog. Por tal razón son muchas las compañías de software que han creado sus propias versiones del mismo. La diferencia entre versiones es mínima, ya que su sintaxis y semántica es la misma, la mayor variación es el cambio de plataforma para las que fueron diseñadas.

PROLOG 1: Esta versión tiene un mejor manejo de los tipos de datos entero y real, y además posee más predicados integrados por lo cual el programador ya no requiere de definirlos.

PROLOG QUINTUS: Es una versión avanzada del lenguaje. El objetivo de sus diseñadores era producir una versión que pudiera ofrecer velocidades rápidas de ejecución, así como la capacidad de comunicación con otros programas. Ahora mismo la versión más actualizada es la 3.5.

MACPROLOG: Esta versión está diseñada totalmente para correr en máquinas MAC, esta versión combina técnicas de programación de inteligencia artificial en ambientes Macintosh.

Implementaciones

Como todo lenguaje, la gente no se quedó con Prolog a secas, sino que ha habido modificaciones respecto a la versión de 1972. Algunas de las implementaciones más importantes de Prolog son:

SWI Prolog:

SWI Prolog es una implementación comunitaria de Prolog, aunque su principal desarrollador, Jan Wielemaker, está respaldado por la Universidad de Amsterdam. SWI Prolog se centra en ser práctico y usable como un lenguaje de programación más, altamente productivo. Prueba de ello lo tenemos en la inmensa cantidad de librerías, herramientas y documentación que tiene el proyecto.



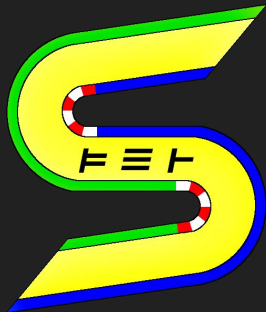
SICStus:

SICStus es una implementación de ISO Prolog de manos de RISE, unos laboratorios de investigación del gobierno de Suecia. Tiene fama de ser una de las implementaciones más fieles al estándar y una de las más rápidas. Pero este intérprete es de pago por lo que tiene menos usuarios de los que debería tener por su calidad técnica

Implementaciones

GNU Prolog:

GNU Prolog es la versión del proyecto GNU de Prolog. Se trata de otra implementación fiel de ISO Prolog y rápida, ya que es capaz de generar ejecutables nativos. Esto último no es nada habitual en los sistemas Prolog.



Scryer Prolog:

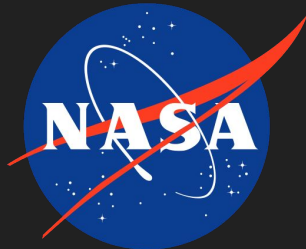
Scryer Prolog es una implementación, todavía en desarrollo, de un entorno fiel a ISO Prolog. Está implementado en Rust y pretende ser un entorno completo con gran cantidad de librerías. Al ser un desarrollo desde cero, pretende alcanzar gran rendimiento usando técnicas modernas.

Software desarrollado en Prolog

Algunos programas importantes desarrollados en Prolog son:

Clarissa

Clarissa es un software de la NASA que recibe instrucciones vía comandos de voz, facilitando así otras tareas de los astronautas



Configuración Wizard de red de Windows NT

El sistema operativo Microsoft Windows NT utiliza un intérprete interno de Prolog para configurar sus sistemas de red de área local (LAN) y de la red de área amplia (WAN).

