

Introducción a Prolog Lógica Computacional 2017-2

Lourdes del Carmen González Huesca
Roberto Monroy Argumedo
Fernando A. Galicia Mendoza

Facultad de ciencias, UNAM

Miércoles, 29 de marzo del 2017



Breve historia

Las raíces de la programación lógica tienen como base la resolución automática de teoremas, cuyo principio recae en la deducción.

En los años 30 Gödel y Herbrand estudiaron la noción de computabilidad basada en encadenamiento de razonamiento, Herbrand estudió la manipulación algebraica de ecuaciones sobre términos, base matemática de lo que conocemos hoy como programación lógica.

En 1965 Robinson brindó las bases para la deducción automática y finalmente en 1974 Kowalski creó una forma de cómputo donde las fórmulas pueden ser vistas como un programa computacional.

El paradigma lógico pertenece al paradigma declarativo, ya que, el código es un conjunto de fórmulas lógicas donde aparecen relacionados los términos.



Aspectos básicos

Las tres características que hacen diferente a la programación lógica respecto a otros paradigmas son:

- Los cálculos se realizan en el dominio de todos los términos definidos sobre un *lenguaje universal*.
- Los valores son asignados a variables, generados automáticamente por sustituciones a través de **unificadores más generales**. Estos valores son llamados *variables lógicas*.
- El mecanismo de cómputo básico es una mezcla de **unificación** y **backtracking**.



Ventajas y desventajas

Ventajas

Código simple y conciso.

Soporta *programación interactiva*, es decir, el usuario puede realizar varias consultas al mismo programa.

Desventaja

Lenguajes de programación lógica (como Prolog) no son fuertemente tipados.



Prolog está basado en un subconjunto de la lógica de primer orden, donde gracias a su sintaxis se puede realizar programación de alto nivel y meta-programación.



Prolog está basado en un subconjunto de la lógica de primer orden, donde gracias a su sintaxis se puede realizar programación de alto nivel y meta-programación.

Los cálculos son ejecutados a través de unificación de ecuaciones, las cuales son llamadas *acciones atómicas*.

A diferencia del paradigma imperativo, las variables podrían no ser utilizadas, es más, los valores de variables son términos.

La programación lógica debe realizar sustituciones para hacer el cómputo de los programas.

Como se mencionó el cómputo de la programación lógica es tratar de llegar a ecuaciones iguales, esto gracias al unificador más general.



Historia de Prolog

Creado por Phillipe Roussel en Algol-W e interpretado por la máquina abstracta creada por David H. Warren (conocida como la máquina WAM). Prolog es la aplicación de los programas lógicos, inició como un lenguaje de programación lógica, sin embargo, después fue utilizado como lenguaje de programación de propósito general.

Actualmente se han creado frameworks a Prolog para aplicaciones de inteligencia artificial por su mecanismo de resolución rápida.



SWI-Prolog

Interprete libre para el lenguaje de programación Prolog, utilizado normalmente para la enseñanza y aplicaciones web semánticas. Goza de ciertas bibliotecas como: programación lógica, restrictiva, multihilos, pruebas unitarias, interfaz gráfica para usuarios, interfaz para Java, etc.



Claúsulas simples

En Prolog contamos con relaciones para definir ecuaciones que definen al programa.

Los símbolos se dividen en:

- Aridad 0, entonces llamados símbolos proposicionales.
- Aridad n , entonces dados t_1, \dots, t_n y R una relación n -aria, se dice que $R(t_1, \dots, t_n)$ es un átomo.

Ejemplo:

`nat(0).`

`nat(suc(x)) :- nat(x).`



Sintaxis

A cada expresión:

$$R(t_1, \dots, t_n) \text{ :- } T_1, \dots, T_n.$$

Es llamada meta, donde $R(t_1, \dots, t_n)$ es la cabeza de la meta y T_1, \dots, T_n es cuerpo de la meta, donde cada T_i es una cláusula.

Cada meta debe terminar con punto (.) y cada cláusula debe ir separado por una coma, es decir, estamos escribiendo en **forma normal conjuntiva**.

Abran su editor de texto favorito y en terminal ejecuten `prolog`. A teclear.

