

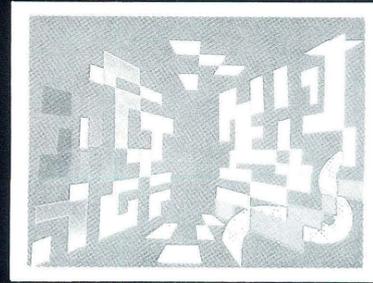
ANOTACIONES TÉCNICAS

PROLOG

Y LOS SISTEMAS

EXPERTOS

Camilo Parra Ramírez*



Prolog es el primer lenguaje de programación que ofrece una base para una programación "no algorítmica". Prolog significa programación lógica, permite la especificación de "que" en vez de "como", razón por la cual ha apoyado el desarrollo de la Ingeniería del Conocimiento (sistemas expertos) y de la inteligencia artificial, entre otros.

Anotemos brevemente la definición de Naylor sobre sistemas expertos: "...un sistema computacional compuesto de una base de conocimiento extraída de un(os) experto(s) de tal forma que el sistema pueda ofrecer un "consejo inteligente" o pueda tomar un "decisión inteligente" acerca del funcionamiento de un proyecto. Este sistema deberá justificar su propia línea de razonamiento lógico de una forma inteligible al usuario..."

Luego de que los japoneses declararan Prolog como lenguaje base para el desarrollo de los computadores de quinta generación, se ha fomentado su estudio fuera de los círculos de la lógica y la inteligencia artificial. La característica de estos lenguajes debe ser el procesamiento de conocimiento teniendo una base de conocimiento y mecanismos de solución con estrategias dadas o aprendidas. Para ello Prolog ofrece mecanismos de deducción e inferencia.

Las raíces de Prolog están dadas en un aporte de Robinson sobre me-

canismos de inferencia automática (1965), el desarrollo de un sistema de resolución de Green (1969) y un trabajo sobre efectividad y "goal-directness" de Kowalski (1970).

Prolog fue desarrollado en 1972 por Colmerauer y sus colaboradores en la Universidad de Marsella (Francia). En Europa y Japón ha tenido una gran acogida y se han impulsado otras versiones en Inglaterra, Alemania, Hungría, Bulgaria y Portugal entre otros.

Los estadounidenses, por su parte, han preferido desarrollar sus sistemas expertos en Lisp y otros lenguajes, aunque ya despertaron al mundo de Prolog, lo que señala un reconocimiento internacional de su importancia.

Prolog ha demostrado su versatilidad en diversos campos, entre los que se cuentan: bases relacionales de datos, lógica matemática, entendimiento del lenguaje natural, decisión automática, diseño automático, bioquímica y, por supuesto, los sistemas expertos y la inteligencia artificial.

En la feria Compuexpo 86, realizada en Bogotá el año pasado, varias empresas mostraron herramientas para construir sistemas expertos y prototipos de esos sistemas que demuestran comportamientos inteligentes. La pregunta que resulta es si tenemos personal para tra-

bajar como ingenieros de conocimiento o adiestrados en estas áreas.

Los sistemas expertos han demostrado bastante eficiencia en tareas de diagnóstico, interpretación, predicción, diseño, instrucción, control y otras. Las aplicaciones que podría tener en el país serían numerosas y si vemos su factibilidad presenta aspectos muy positivos: no se requieren, por ejemplo, computadores especiales, los que tenemos serían suficientes; existen expertos nacionales en diversas áreas y en temas claves para el desarrollo científico nacional. Faltaría, eso sí, personal adiestrado en estas nuevas tecnologías y aquel dedicado al desarrollo de este campo que muestra una proyección muy grande dentro del contexto informático mundial.

La importancia de tener familiaridad con nuevos lenguajes de programación es de vital importancia si se desea tener un desarrollo aceptable en la industria nacional de software.

Veamos rápidamente la estructura de lenguajes como Prolog el cual tiene una notación simple y compacta. Su principal elemento es el "término" que puede ser:

átomo/functor: es_abuelo_de,
"ALGO", "Nada, plantas, recursos"
número: 1, 1986, 20

variable: Quien, Como, Variable, K estructura
 pantalla (resolución, tamaño, color, técnica de graficado).

donde la estructura consta de: func-
 tor (termino,.....,termino).

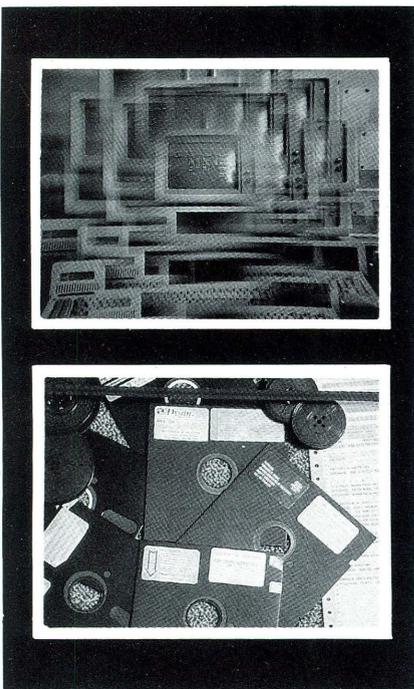
Un programa Prolog consta de las
 siguientes cláusulas:

a) hechos (factos): que dicen algo
 sobre el objeto o su relación:

pedro es padre de carlos.
 o
 es padre de (pedro, carlos).

b) reglas: dicen como se interrela-
 cionan los hechos:

X es abuelo de Y si
 X es padre de Z y
 Z es padre de Y.
 O
 es abuelo de (X,Y):—
 es padre de (X,Z),
 es padre de (Z, Y).



c) preguntas: las formula el usuario
 al interpretador y éste intenta
 responderlas con base en sus
 hechos (conocimientos) y en sus
 reglas (estrategias):

? quien es padre de quien
 o
 ? es padre de (X, Y).

Prolog tiene muchas otras ayudas
 (predicados, listas) y mecanismos
 de programación basados en la re-
 cursión (rutinas que se llaman a ellas
 mismas varias veces sin perder el
 control de la secuencia) y en el lla-
 mado "backtracking" en el cual el
 interpretador Prolog cuando no en-
 cuentra una solución se devuelve al
 último estado correcto e intenta en-
 contrar una nueva solución por otro
 camino, todo automáticamente, lo
 que hace de Prolog una herramienta
 de programación poderosa. Para
 búsqueda de errores y ayuda al
 aprendizaje ofrece predicados para
 ver en cualquier momento el conte-
 nido de la memoria y ejecutar el
 programa paso a paso (debugg, tra-
 ce).

La arquitectura de los computa-
 dores de quinta generación se basa
 en esta idea de búsquedas automá-
 ticas. Se dispondrá de un computa-
 dor con varios procesadores (multi-
 procesador), en el cual cada uno de
 los procesadores, seguirá una ruta
 de búsqueda y otros se encargarán
 de administrar todo el proceso. Este
 modo llamado "procesamiento
 en paralelo" se está estudiando
 a fondo ya que los problemas de sin-
 cronía son relativamente difíciles.
 Así el Prolog no funcionará como
 hasta ahora en el momento en que
 encuentra un error se devuelve y
 busca por otro camino, si no que se
 encontrará rápidamente la respues-
 ta siguiendo todas o varias de las
 posibilidades al mismo tiempo.

Los amigos de la programación
 "clásica" pueden seguir el siguien-
 te programa el cual suma todas las
 cifras desde 1 hasta N y da el resul-
 tado en R:

sume (1, 1): — !.
 sume (N, R) : —
 N1 is N - 1,
 sume (N1, R1),
 R is R1 + N.

La recursividad, el backtracking y
 el estilo de programación hacen un
 poquito difícil la comprensión a
 simple vista; el "cut" que se denota
 con el signo de admiración (!), inte-
 rrumpe el backtracking, lo que hace
 que los programas sean más veloces
 cuando de antemano se conocen las
 partes donde no vale la pena conti-
 nuar buscando.

El unir Prolog con un paquete
 gráfico proporciona nuevas técnicas
 para el diseño, ya que aprovechando
 su forma de programación (re-
 cursividad, backtracking) se pueden
 generar figuras en forma diferente a
 la convencional. Esto ya se ha hecho
 con la versión Ifprolog y el GKS
 (Graphical Kernel System) habién-
 dose desarrollado prototipos de sis-
 temas expertos con interfases gráfi-
 cas.

En Prolog existen numerosos pro-
 totipos de sistemas expertos y los
 llamados "Schell" que son estruc-
 turas básicas para la construcción y
 desarrollo de estos sistemas exper-
 tos con bases de conocimientos muy
 bien logradas.

Uno de los principales problemas
 es la representación del conocimien-
 to, o sea el poder interpretar el
 modo como están relacionados los
 conocimientos en nuestro cerebro
 para que Prolog puede "deducir"
 de la misma manera.

Como todo lenguaje de progra-
 mación, Prolog tiene sus amigos y
 enemigos; nosotros no queremos
 afirmar que sea la última palabra en
 el área pero deseamos resaltar su
 importancia ya que este tipo de len-
 guajes ha partido en dos grandes
 grupos el desarrollo histórico de
 lenguajes de programación y ha ge-
 nerado nuevos estilos para enfocar
 problemas. □

* Ingeniero de Sistemas. Jefe Departamento
 de Información, Empresa de Acueducto y
 Alcantarillado de Bogotá, Calle 22C No.
 40-99, Bogotá.