

Introducción al lenguaje de programación lógica Prolog

Kathryn M. Jones Pérez

Universidad de Costa Rica, escuela de Ciencias de la Computación e Informática,
San José, Costa Rica
kathrynster@gmail.com

y

Gustavo A. Yong Morales

Universidad de Costa Rica, escuela de Ciencias de la Computación e Informática,
San José, Costa Rica
tavoyong@gmail.com

Abstract

Prolog is a programming language used to implement artificial intelligence and expert systems. A large part of its success is due to its convenience, being open source (easily modified) and it can be downloaded from the internet, not to mention the languages deduction capabilities for answering questions asked by the user. Prolog is a simple and easy language to program in, even for beginners, but its inference motors are not always efficient. Applications vary from environmental systems to resolution of automatist systems.

Keywords: Prolog, Programming, Logic, Artificial Intelligence, Expert Systems

Resumen

Prolog es un lenguaje utilizado para implementar inteligencia artificial y sistemas expertos. Gran parte de su éxito se debe a su conveniencia por ser código abierto (modificable) y se obtiene fácilmente en Internet, además de su capacidad de deducción de respuestas para las consultas realizadas. Prolog es un lenguaje simple y fácil de programar, hasta para principiantes, pero sus motores de inferencia no siempre son eficientes. Sus aplicaciones varían desde sistemas ambientales hasta la resolución de funciones automatizadas.

Palabras clave: Prolog, Programación, Lógica, Inteligencia Artificial, Sistemas Expertos

1 Introducción

Prolog es un lenguaje de programación diseñado para representar y utilizar el conocimiento que se tiene sobre un determinado dominio. Los programas en Prolog responden preguntas sobre el tema del cual tienen conocimiento.

Forma parte de lo que se conoce como programación declarativa. En lenguajes tradicionales se indica como resolver un problema, en la programación lógica se establecen hechos reales y reglas para así no saber *como* resolver el problema, sino *que hacer* para resolverlo.

La popularidad de este lenguaje se debe a su capacidad de deducción y además es un lenguaje fácil de usar por su semántica y sintaxis. Sólo busca relaciones entre los objetos creados, las variables y las listas, que son su estructura básica.

En este artículo se dará una reseña histórica, para entender los orígenes del Prolog, una pequeña descripción del programa, así como las estructuras básicas y algunos operadores. También se mencionaran algunas ventajas y desventajas de Prolog, así como casos de uso. Por último se mostrarán tres ejemplos para mostrar la sintaxis y funcionamiento de Prolog.

2 Historia

Los inicios de la programación lógica se dan gracias a los primeros trabajos de inteligencia artificial. Los cuales originaron el primer lenguaje de programación que contempla los mecanismos de inferencia necesarios para la demostración automática de teoremas.

El lenguaje de programación Prolog se originó del trabajo hecho por Robert A. Kowalski en la Universidad de Edinburgh y Alain Colmerauer en la Universidad de Aix-Marseille (Francia) en los años 70. La investigación de Kowalski en el área de deducción automatizada, llevó al desarrollo con Colmerauer al uso formal de lógica como un lenguaje de programación. Kowalski proporcionó la base teórica y Colmerauer inició la programación de Prolog. Colmerauer y Phillippe Roussel desarrollaron el primer intérprete, y David Warren de la Universidad de Edinburgh desarrolló el primer compilador Prolog. La mayoría de las implementaciones comerciales de Prolog usan la misma sintaxis desarrollada en Edinburgh. Su nombre proviene de las palabras en inglés "Programming in Logic". Desde su creación, Prolog ha crecido en popularidad en Europa, América y Japón. [2]

3 Descripción

Prolog es un lenguaje de programación simple, pero poderoso. Se basa en nociones matemáticas de relaciones de inferencia. Es un lenguaje declarativo e interpretado, esto quiere decir que el lenguaje se usa para representar conocimientos sobre un determinado dominio y las relaciones entre objetos de ese dominio. [1]

Un programa en Prolog consiste de una base de datos de relaciones lógicas y detalles que se cumplen para la aplicación. Dicha base de datos no tiene una estructura impuesta, ni un procedimiento o clase principal. Los datos y relaciones de un programa en Prolog se escriben en un único archivo, el cual es consultado por el programa cuando se le hace una pregunta.

Escribir un programa en Prolog consiste en declarar el conocimiento disponible acerca de los objetos, además de sus relaciones y sus reglas. [6] En lugar de correr en un programa para obtener una solución, se hace una pregunta, el programa revisa la base de datos para encontrar la solución a la pregunta. [1] Si existe más de una solución, Prolog hace backtracking para encontrar soluciones distintas. El propio sistema es el que deduce las respuestas a las preguntas que se le plantean, dichas respuestas las deduce del conocimiento obtenido por el conjunto de reglas dadas. [1]

La ejecución de Prolog consiste en una búsqueda en profundidad de un árbol conteniendo todas las posibles soluciones. Para cada una de ellas se evaluará su validez. Más adelante se explicará el funcionamiento de un árbol de deducción (Fig. 8) relacionado con un ejemplo.

La estructura de un programa en Prolog es lógica y directa. Se explican cada una de sus partes y operadores disponibles a continuación.

3.1 Estructura de un programa en Prolog

La sintaxis Prolog incluye:

- Hechos: es un hecho del lenguaje cotidiano. En lenguaje normal, una proposición puede ser “hace calor”.
- Reglas: las capacidades del programa. Estas son consultadas al hacer una consulta para verificar su validez o invalidez.
- Variables: Representan objetos que el mismo Prolog determina. Una variable puede estar instanciada o no instanciada.
- Consultas: es la acción de hacerle una pregunta al programa sobre la información contenida en la base de datos. Una pregunta sencilla podría ser: ?- 'Hace Calor'. Una respuesta podría ser: si, seguido por el prompt: ?-. Un si significa que la pregunta es consistente con la información contenida en la base de datos. Si no existe suficiente información, responde: no. [1]

3.2 Operadores Aritméticos

- + : suma
- - : resta
- * : multiplicación
- / : división
- // : división de enteros (truncada)
- mod: resto de la división
- **: potenciación

3.3 Operadores Lógicos

- not : negación
- \+ : no probable
- , : conjunción lógica
- ; : disjunción lógica
- :- : implicación lógica
- -> : if-then-else

Ya que se conocen las generalidades de la programación en Prolog, mencionaremos algunas ventajas y desventajas de implementar un programa en este lenguaje.

4 Ventajas y Desventajas

4.1 Ventajas

- Una ventaja desde el punto de vista del usuario es la facilidad para programar ya que se pueden escribir programas rápidamente, con pocos errores originando programas claramente legibles, aun si no se conoce muy bien el lenguaje.
- Otra ventaja de este tipo de lenguajes frente a otros, como Java o C++, es que no hay que pensar demasiado en la solución del problema, ya que Prolog infiere sus respuestas basándose en las reglas declaradas dentro del programa.

4.2 Desventajas

- La resolución automática no siempre es eficiente, por lo que eventualmente se podría dar una respuesta incorrecta a una consulta.
- Ciertos problemas están ligados a la representación del conocimiento, que Prolog no posee.

- Prolog es algunas veces es incapaz de reconocer que un problema es (para su propio conocimiento) inaplicable o insuficiente. Si el programa no contiene suficiente información para contestar una consulta, es incapaz de reconocerlo y responde no. En esta situación sería más eficiente conocer que la respuesta no es negativa, sino que no es posible inferir un resultado.
- Los motores de inferencia poseen algunos límites.

5 Contexto de uso

Prolog se usa en diversos campos, incluyendo finanzas, defensa, telecomunicaciones, medicina, agricultura, ingeniería, manufactura y educación. [2]

Se utiliza en aplicaciones de inteligencia artificial, como interfaces de idiomas en los que busca en la base de datos para responder una pregunta. [2]

Otra área de su aplicación es en sistemas ambientales que son capaces de predecir el clima, analizar suministros de agua, entre otros. Un ejemplo es el MM4 Weather Modeling System, desarrollado en la Universidad Penn State y el Centro Nacional de Investigación Atmosférica (Co, EEUU), el cual es capaz de predecir condiciones meteorológicas a corto plazo. El sistema analiza imágenes de satélite y de acuerdo con los patrones de las nubes se puede pronosticar el tiempo de los siguientes días. [2]

Prolog se puede utilizar para representar autómatas. Al fijar los hechos (que generalmente consisten en las transiciones del autómata) se pueden hacer consultas para verificar si una expresión es o no aceptada por el autómata.

Un ejemplo de la vida real del uso de Prolog en una aplicación Java es una compañía que brinda servicios para manejar el financiamiento de propiedades. Es una aplicación web que informa a los usuarios de la mejor solución para un préstamo hipotecario. El módulo fue desarrollado usando 5000 líneas de código de Java y varias tablas de una base de datos. EL problema de esta aplicación es que era necesario cambiarla constantemente para adaptarse a nuevas reglas y factores de tasación. En este caso se precisaba una solución con menos líneas de código para facilitar todos los cambios necesarios y minimizar los errores. Se construyó un modelo lógico con Amzi! Prolog, con lo que las 5000 líneas de código y las tablas de base de datos se redujeron a simplemente 500 líneas Prolog. “La base lógica resultante estaba casi libre de errores, y el ciclo de modificación/prueba se redujo en gran forma.” [7]

Con este ultimo ejemplo vemos el poder este lenguaje y su utilidad no solo para programas escritos en Prolog, sino también para migrar y mejorar aplicaciones implementadas en otros lenguajes. A continuación se explica donde se puede conseguir y como se instala el compilador utilizado en la ultima aplicación mencionada.

6 Donde conseguir el compilador

Por ser opensource, se puede descargar gratis de varias páginas. Algunos ejemplos son:

<http://www.visual-Prolog.com/vip6/product/default.htm>

http://www.amzi.com/download/Prolog_download_sites.htm

<http://win.allbestsoft.com/freedownload-2633-visual-Prolog.html>

7 Como instalar Amzi! Prolog

Instalación de Amzi! Prolog en Windows: La instalación inicia con la ejecución de amzi_N-N-N_win.exe, el cual instalará Amzi! en el directorio escogido. Luego seguir los siguientes pasos:

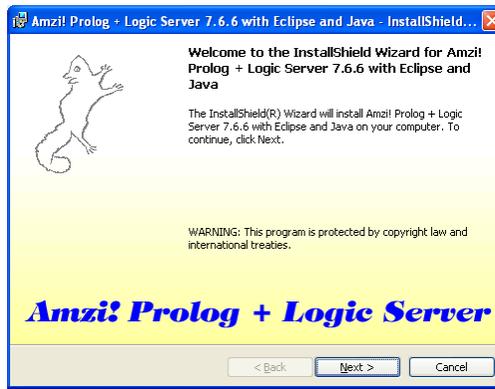


Figura 1: Primero aparece la pantalla de Bienvenida. Se selecciona el botón marcado “Next” para seguir.

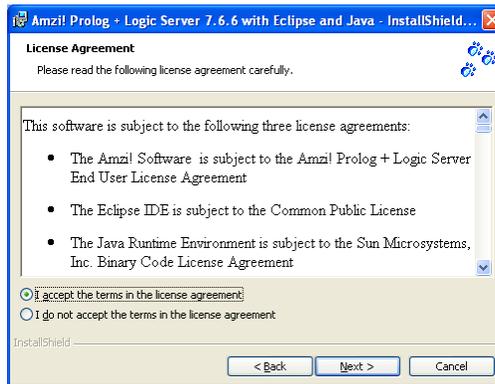


Figura 2: Se acepta los términos en el acuerdo de licencia (SE selecciona el botón “Next”).

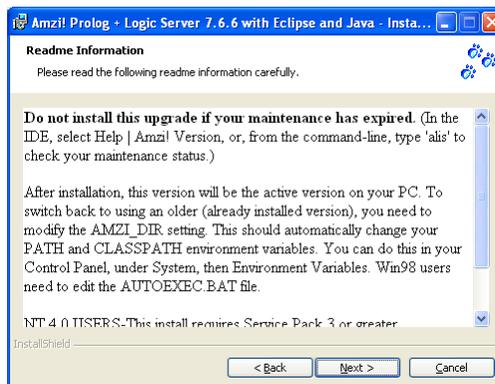


Figura 3: Se presenta información sobre el programa que se está instalando.

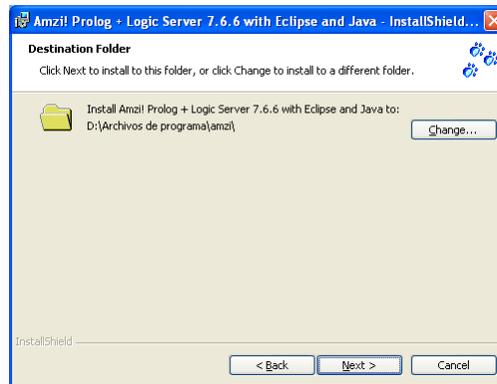


Figura 4: Se escoge el destino de instalación.



Figura 5: Se confirma la información de instalación para iniciar el proceso.

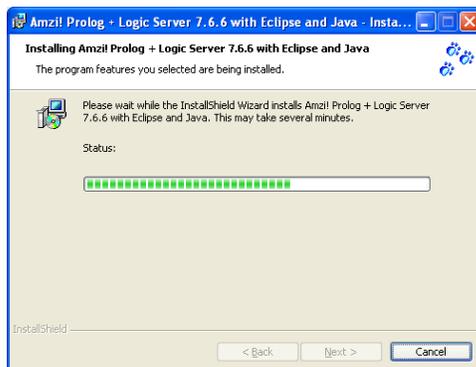


Figura 6

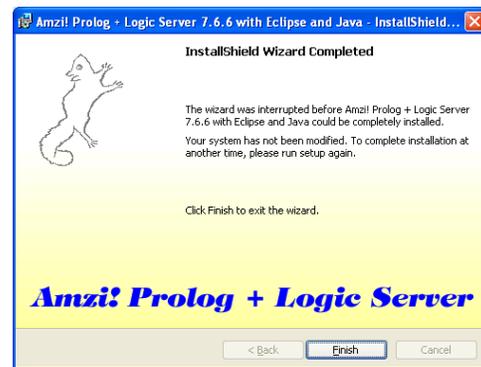


Figura 7

Luego de finalizar la instalación, se puede iniciar la programación en Prolog.

8 Ejemplos de programación en Prolog

- **Ejemplo #1:** se muestra el uso de la instrucción lógica “if” en Prolog.

```
%%
%% Hechos
%%
```

```

padrede('juan', 'maria').    % juan es padre de maria
padrede('pablo', 'juan').    % pablo es padre de juan
padrede('pablo', 'marcela'). % pablo es padre de marcela
padrede('carlos', 'debora'). % carlos es el padre de debora

%
%Relaciones
%

hijode(A,B) :- padrede(B,A).          % A es hijo de B si B es padre de A.
abuelode(A,B) :- padrede(A,C), padrede(C,B). % A es abuelo de B si A es padre de C y C es padre B.
hermanode(A,B) :- padrede(C,A) , padrede(C,B), A \== B. % A y B son hermanos si el padre de A es también
% el padre de B y si A y B no son lo mismo.

% A y B son familiares si A es padre de B o A es hijo de B o A es hermano de B
familiarde(A,B) :- padrede(A,B).
familiarde(A,B) :- hijode(A,B).
familiarde(A,B) :- hermanode(A,B).

%%
%% Consultas de prueba
%%

?- hermanode('juan', 'marcela'). % juan es hermano de marcela?
yes

?- hermanode('carlos', 'juan').   % carlos es hermano de juan?
no

?- abuelode('pablo', 'maria').    % pablo es abuelo de maria?
yes

?- abuelode('maria', 'pablo').    % maria es abuela de pablo?
no

[4]

```

La parte del ejemplo que ilustra el uso del 'if' son las relaciones: hijode(), abuelode(), hermanode() y familiarde(). Las relaciones no se cumplen a menos de que se cumpla la condición después del operador de implicación lógica (:-)

- **Ejemplo #2:** Se dispone de una base de conocimiento compuesta de reglas sobre clasificación de animales y hechos sobre características de un animal.

Regla 1: Si un animal es ungalado y tiene rayas negras, entonces es una cebra.

Regla 2: Si un animal rumia y es mamífero, entonces es ungalado.

Regla 3: Si un animal es mamífero y tiene pezuñas, entonces es ungalado.

Hecho 1: El animal es mamífero.

Hecho 2: El animal tiene pezuñas.

Hecho 3: El animal tiene rayas negras. [5]

```

%
%Reglas
%

es_cebra :- es_ungulado, tiene_rayas_negras. % Regla 1
es_ungulado :- rumia, es_mamífero.          % Regla 2
es_ungulado :- es_mamífero, tiene_pezuñas.  % Regla 3

%
%Hechos
%
```

```

es_mamífero.      % Hecho 1
tiene_pezuñas.   % Hecho 2
tiene_rayas_negras. % Hecho 3

```

```

%
%Consulta de prueba
%
```

```

?- es_cebra
Yes

```

Prolog para haber llegado a responder “yes” a la pregunta de si “es cebra?” realizó un árbol de deducción

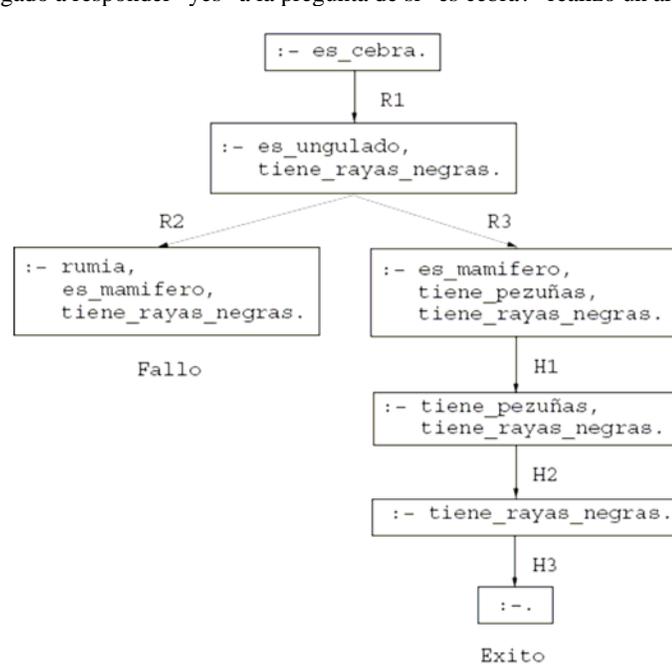


Figura 8: árbol de deducción del problema de los animales [5]

8 Conclusiones

Las bases de Prolog se remontan a la necesidad de automatizar ciertos procesos, como el de la resolución de autómatas deterministas finitos.

Prolog es una herramienta sencilla, potente y útil para el desarrollo de sistemas expertos e inteligencia artificial. Su popularidad se debe a sus aplicaciones en sistemas complejos de alta demanda en el mercado actual. Su crecimiento seguirá mientras exista necesidad de este tipo de sistemas.

Prolog tiene la ventaja que el programa revisa la base de datos para encontrar soluciones a las preguntas, infiriendo el mismo la respuesta basándose en las reglas del programa. La desventaja que esto posee, es que la resolución automática no siempre es eficiente.

Algo que le ha dado auge a Prolog, es su capacidad ser bajado fácilmente de la Internet, ya que este programa es de código abierto, por lo que es gratuito. Al ser de código abierto, se pueden modificar las versiones Prolog, e incluso hay muchas versiones en la web.

9 Bibliografía:

- [1] Aaby, Anthony. [Prolog Tutorial http://cs.wvc.edu/KU/PR/Prolog.html](http://cs.wvc.edu/KU/PR/Prolog.html), 5 de Febrero del 1997
- [2] Roth, Al. The Practical application of Prolog, *Dr. Dobbs Portal* <http://www.ddj.com/hpc-high-performance-computing/184405220>, 10 de Diciembre 2002
- [3] Amz1! Inc. Requirments and Instalation. <http://www.amzi.com/manuals/amzi7/install.htm>, 2004
- [4] Wikipedia. <http://es.wikipedia.org/wiki/Prolog>, 2 de Noviembre 2007
- [5] Alonso Jiménez, José. Introducción al programación lógica de Prolog http://www.cs.us.es/~jalonso/publicaciones/2006-int_Prolog.pdf, 17 de Junio 2006
- [6] Correas Fernandez, Jesús. Programación lógica Prolog. http://clip.dia.fi.upm.es/~jcorreas/Prolog/master_Prolog_2.pdf, 17 de Junio 2002
- [7] Rossel, Gerardo. Programación lógica. http://www.amzi.com/articles/code07_whitepaper.pdf, 15 de Marzo del 2004