Programación lógica

Pedro O. Pérez M., PhD.

12-2025

Implementación de métodos computacionales
Tecnológico de Monterrey
pperezm@tec.mx

Contenido i

Contenido ii

Paradigma de la programación lógica

Paradigma de la programación lógica

- Perteneciente al grupo de los **lenguajes declarativos**.
- Planteamiento de la solución a un problema basado en un QUÉ y no en un CÓMO.
- Paradigma de **mayor abstracción**.
- Basado en el **pensamiento lógico del ser humano** (Lógica como área de estudio: Aristóteles, Boole).
- Formalmente basado en el **cálculo de predicados de primer orden**.

Características de la programación lógica

- Abstracción de control basada en la **recursividad y decisiones implícitas**.
- Abstracción de datos basada en la estructura de la **lista** y en describir **relaciones** entre datos. Tipos de datos dinámicos.
- **Modularización** natural a través de **cláusulas** y parámetros que se instancian según la ejecución.
- Programas **no determinísticos**.
- Ambiente de traducción basado en **interpretación**.
- Internamente se generan **estructuras jerárquicas para la búsqueda automática de soluciones**. Ejecución optimizable.

Aplicaciones principales

- Inteligencia Artificial.
- Bases de datos relacionales.
- Procesamiento de lenguaje natural.
- Sistema expertos, representación del conocimiento.

También conocida como Lógica de Primer Orden

- Operadores:
  - Negación.
  - Conjunción.
  - Disyunción.
  - Implicación.
  - Equivalencia.
- Cuantificadores.
  - Universal.
  - Existencial.
- Cláusulas.
- Axiomas.
- Términos.
- Inferencias.
- Resolución.
- Unificación.

Esencia de la programación lógica (Cláusula de Horn)

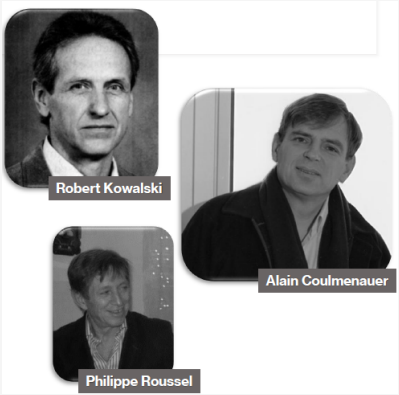
$$a_1 \text{ y } a_2 \text{ y } a_3 \text{ y } \ldots \text{ y } a_n \rightarrow b$$

- b es verdad si todas las a_i son verdaderas. Representa una implicación.
- Es la base para la construcción de programas lógicos por la facilidad de interpretación procedural.
- Un programa es un conjunto de axiomas que definen las relaciones entre objetos. La ejecución consiste en obtener la deducción de consecuencias utilizando las reglas definidas, dándole así un significado al programa.

Conceptos básicos

Lenguaje Prolog

- Surge a inicios de los años **70´s** en la Universidad de Marsella en Francia, con los antecedentes de investigaciones en la Universidad de Edimburgo en Reino Unido.
- Sus creadores son Alain Coulmenauer y Philippe Roussel, basados en los estudios previos de Robert Kowalski.
- Significa: "**Programming in Logic**".



Un programa en Prolog se compone de...

1. Declaraciones de **HECHOS** acerca de los objetos y sus relaciones.
   - Son cláusulas de Horn sin cuerpo $\rightarrow$ Axiomas, Verdades.
2. Declaración de **REGLAS** acerca de los objetos y sus relaciones.
   - Son cláusulas de Horn con cuerpo $\rightarrow$ Implicaciones, decisiones implícitas
3. Consultas, interrogaciones o queries.
   - Provocan el proceso de resolución y unificación de reglas y/o hechos para instanciar variables y dar **resultados**.

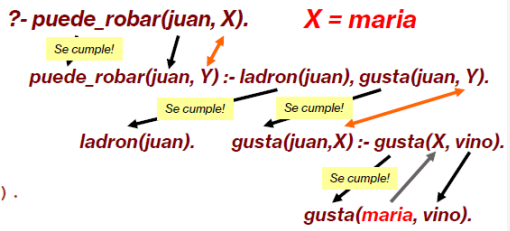
Formato general para las cláusulas (estatutos) de Prolog

?- CABEZA :- CUERPO.

- ?- Se utiliza sólo si se trata de una consulta.
- **CABEZA** corresponde a una estructura simple.
- **CUERPO** puede contener 0, 1 ó más estructuras separadas por comas o puntos y comas.
  - Cuando se omita esta parte la cláusula corresponde a un **HECHO**.
  - En caso de tener 1 o más estructuras , la cláusula corresponde a una **REGLA**.
  - Las **comas** indican **conjunciones**; los **puntos y comas** indican **disyunciones**.

¿Cómo funciona Prolog?

```
ladron(juan).
gusta(maria, comida).
gusta(maria, vino).
gusta(juan, X):- gusta(X, vino).
puede_robar(X, Y):- ladron(X), gusta(X, Y).
```



- La ejecución de la consulta provoca la búsqueda de la solución a través de "**pattern matching**", generando un **árbol de búsqueda de soluciones** en el que hay **backtraking**.

Operaciones relacionales

Conforman cláusulas en formato infijo y por si mismos arrojan un valor de verdad.

- Igual que: =
- Diferente: ¯
- Menor que: <
- Mayor que: >
- Menor igual que: =<
- Mayor igual que: >=

Operadores aritméticos

Conforman cláusulas en formato infijo cuyo resultado debe instanciarse a una variable a través del operador `is`.

- Suma: `+`
- Resta: `-`
- Multiplicación: `*`
- División: `/`
- Residuo: `mod`

Es requisito utilizarlo cuando se requiere hacer una evaluación aritmética.

Formato: **`Variable is expresión_aritmética`**

Instancia a la variable con el resultado de la expresión, y la cláusula es VERDADERA por default.

Reglas como módulos

- Bajo un enfoque de la abstracción modular, las reglas son MÓDULOS, y las variables de la regla , son parámetros de entrada y/o salida según el caso

Reglas recursivas

- Reglas cuyo cuerpo , tienen términos que corresponden a la cabeza de la propia regla.
- Ejemplos:
  - abuelo(X,Y) padre(X, Z), padre(Z,Y).
  - bisabuelo (X,Y) padre(X, Z), abuelo (Z,Y).
  - tatarabuelo (X, padre(X, Z), bisabuelo (Z,Y).

PENSAMIENTO RECURSIVO

- Se aplica de la misma manera.
- La implementación involucra tener al menos una regla para el caso base, y al menos una regla recursiva (que se llama a sí misma).
- La decisión de evaluar un caso o el otro, está implícita en la manera en que trabaja el intérprete.

Ejemplos

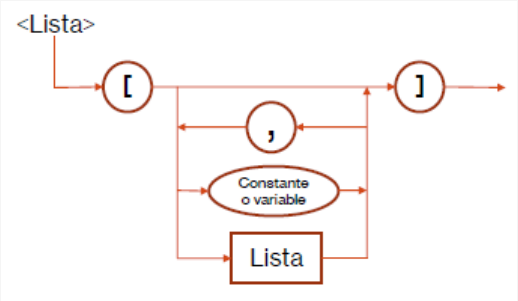
- Calcular el factorial de n.
- Calcular la cantidad de dígitos de un número entero n.

Revisar actividad en Canvas.

Listas

Listas en Prolog

Sintaxis



Ejemplos

- `[ ]`
- `[1, 2, 3, 4, 5]`
- `[juan, maria, pedro]`
- `[A, B, C]`
- `[1, a, X, 58, luis]`
- `[x [M , N], U, [[a, b, c], d, [e]]`
- `[a+b, a*b, c-d]`

Notación especial para el manejo de listas

Ejemplos:

- `[juan, maria, pedro]` <=> cabeza = juan, cola = `[maria, pedro]`
- `[A, B]` <=> cabeza = A, cola = `[B]`
- `[x]` <=> cabeza = x, cola = `[]`

Este formato sólo puede utilizarse como descriptor de la parte final de una lista

Ejemplos

- Verificar si un elemento es miembro de una lista.
- Contar los elementos de una lista.
- Unir dos listas.
- Invertir una lista.

Revisar actividad en Canvas.