



Universidad Tecnológica Nacional
FACULTAD REGIONAL CORDOBA

PARADIGMAS DE PROGRAMACION

Unidad V
Paradigma Lógico
Parte I

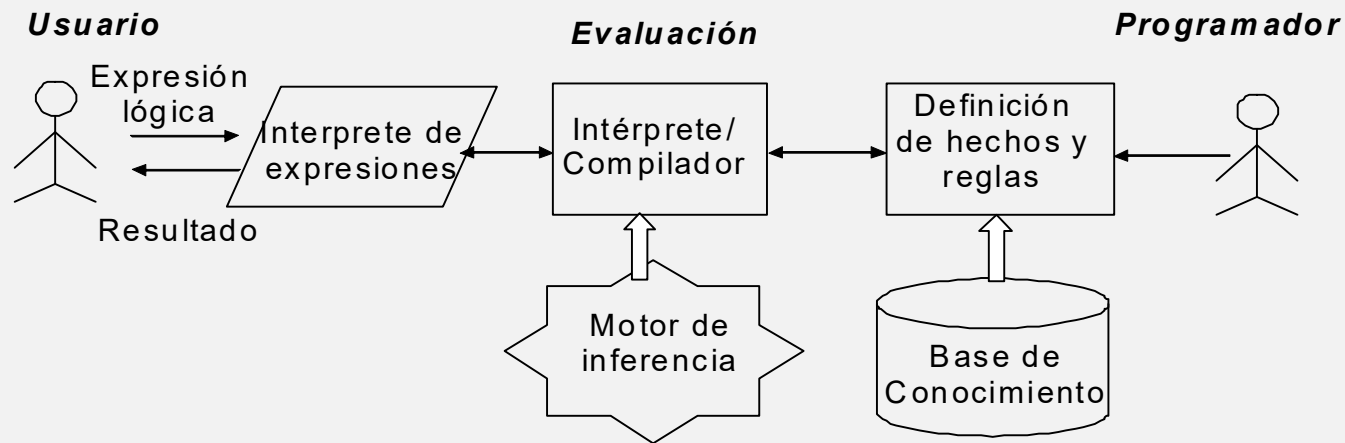
CONTENIDOS ABORDADOS

- Introducción al paradigma, ventajas, desventajas, aplicaciones.
- Lógica proposicional o de primer orden.
- Prolog: introducción, elementos, cálculo de relaciones, estructura de los programas.
- Ejemplificación.

Paradigma lógico: Introducción

- Es un paradigma declarativo que se basa en la definición de reglas lógicas, que se evalúan a través de un motor de inferencias lógicas y generan conclusiones aceptadas como válidas.

Lógica + Control = programa



Paradigma lógico: Introducción

Aplicaciones

- Inteligencia artificial, sistemas expertos y robótica.
- Sistemas basados en conocimiento.
- Procesamiento de lenguaje natural.
- Demostración automáticas de teoremas.

Lógica Proposicional

- La lógica proposicional o lógica de primer orden es uno de los formalismos mas utilizados para representar conocimiento en Inteligencia Artificial.
- **Elementos:**
 - Constantes individuales
 - Símbolos de predicado
 - Enunciados atómicos
 - Enunciados atómicos combinados
 - Predicados con consecuente

Lógica Proposicional: Elementos

- **Constantes individuales:** Son símbolos (nombres) que se usan para referir a algún objeto individual fijo.

Ejemplo: *Juan* es una constante individual que representa o denota una persona particular.

- **Símbolos de predicado:** Son utilizados para denotar alguna propiedad o relación entre objetos.

Ejemplo: *Ana es docente.*

Juan es alumno.

Ana enseña a Juan

Lógica Proposicional: Elementos

- **Enunciados atómicos:** Son enunciados formados por un predicado seguido por el número correcto de nombres, es decir, poseen un predicado simple y el número apropiado de constantes individuales.

Ejemplo:

docente(Ana)

alumno (Juan)

enseña(Ana, Juan)

El **orden** de los nombres en un enunciado atómico es importante.

enseña (Ana, Juan) no es lo mismo que

enseña (Juan, Ana)

Lógica Proposicional: Elementos

- **Enunciados atómicos combinados:** Son enunciados que combinan dos o mas enunciados simples utilizando un conectivo lógico (o, y, no).

Ejemplo:

docente (Ana) y alumno(Juan)

docente (Ana) y enseña (Ana, Juan)

- **Predicados con consecuente:** Constituido por una serie de argumentos uno llamado conclusión que son apoyados por otros llamados antecedentes.

Ejemplo:

docente (Ana) si alumno(Juan) y enseña (Ana, Juan)

conclusión

antecedentes

Prolog : Introducción

- Es un lenguaje de programación declarativo basado en la lógica de primer orden.
- Es un lenguaje de programación hecho para *representar y utilizar el conocimiento* que se tiene sobre un determinado dominio.
- Cada “línea de programa” es una declaración, no una orden. Se tiene un conjunto de aseveraciones expresadas simbólicamente, que expresan conocimientos de una situación real o ficticia.
- La lógica que encuentra el conjunto de soluciones posibles está implícita en el motor de inferencia, no hay que programarla.

Prolog: Cálculo de relaciones

- En prolog se utiliza un tipo determinado de reglas para definir relaciones, llamadas **cláusulas de Horn**.
- Las cláusulas de Horn son fórmulas (predicados) bien formadas, que se encuentran en forma clausal.
- Estas reglas están compuestas por dos partes: el **consecuente** y el **antecedente**.
- Estructura general:

conclusión *si* **condición**

B <- A1,A2,...,An

conclusión o condición o

consecuente antecedente

Prolog: Cálculo de relaciones

Estas cláusulas pueden adoptar las siguientes formas:

- conclusión

$B \leftarrow$

Afirmación ó Hecho

- conclusión si condición

$B \leftarrow A_1, A_2, \dots, A_n$

Implicación ó Regla

- si condición

$\leftarrow A_1, A_2, \dots, A_n$

Negación ú Objetivo

Prolog: Cálculo de relaciones

Afirmación

B<- Afirmación incondicional

- Estamos afirmando algo que no necesita ser probado. Esto significa que estamos diciendo una verdad que no debe ser comprobada, que no tiene condición de valides.
- Son llamados hechos, porque son verdades por sí mismas.
- Por ejemplo : docente (ana)

Decimos que ana posee la propiedad de ser docente, sin que se deba satisfacer ninguna condición.

Prolog: Cálculo de relaciones

Implicación

$B \leftarrow A_1, A_2, \dots, A_n$ Afirmación condicional

donde el predicado B es verdadero si A_1, A_2, \dots, A_n son verdaderos conjuntamente.

- Es un hecho condicional, llamado predicado con consecuente en la lógica de primer orden. Con este tipo de estructuras manifestamos que un objeto puede poseer cierta propiedad o que puede existir cierta relación entre objetos si se cumple la condición.
- Ejemplo: `docente(ana) si enseña(ana, juan)`
Expresa la idea de que “ana es docente si le enseña a juan”

Prolog: Cálculo de relaciones

Negación

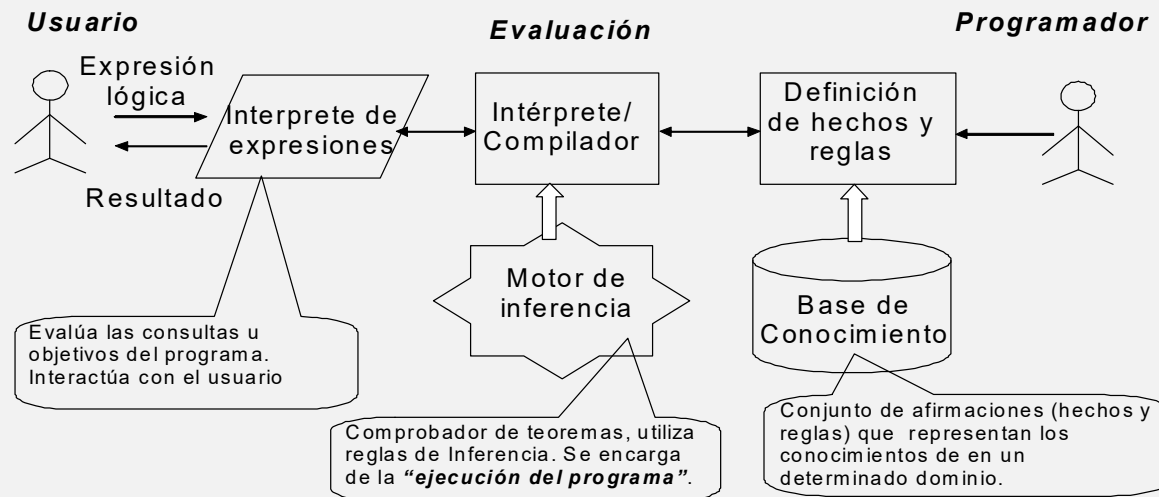
$\leftarrow A_1, A_2, \dots, A_n$ Cláusula objetivo o goal

Cada uno de los predicados A_1, A_2, \dots, A_n de la cláusula son denominados subobjetivos. Esta es en realidad la cláusula que queremos probar del conjunto de sentencias del programa.

- Se deduce que en la prueba del objetivo se esta empleando el principio de resolución de Robinson. Se niega la cláusula que se desea probar y se la agrega a las cláusulas del programa. Luego se aplican las reglas de resolución y, al derivarse una cláusula nula, queda demostrado que la cláusula original es verdadera.

Prolog: Elementos de un programa

- **Base de Conocimiento:** Está compuesto por un conjunto de afirmaciones (hechos y reglas) que representan los conocimientos que poseemos en un determinado dominio.
- **Motor de Inferencia:** Es el que se encarga de la “*ejecución del programa*”. Es un comprobador de teoremas, el cual utiliza la Reglas de Inferencia.
- **Intérprete de expresiones:** Permite evaluar las consultas u objetivos del programa. Interactúa con el usuario, mostrando los resultados.



Prolog: Programa

- Un programa de Prolog consiste en una base de datos de hechos y reglas.
- No hay imposición de estructura en un programa Prolog, no hay procedimiento principal, y no hay anidamiento de definiciones.
- Todos los hechos y reglas son globales en alcance, y el alcance de una variable es el hecho o la regla donde aparece.
- Un programa en Prolog se ejecuta realizando una pregunta. La pregunta se llama consulta. Los hechos, reglas y consultas se llaman cláusulas.

Prolog: Estructura de un programa

Los pasos a seguir para escribir un programa en Prolog:

- Declarar y definir HECHOS y REGLAS sobre los objetos y relaciones en un archivo .pl.
- Hacer CONSULTAS sobre los objetos y relaciones, en la ventana o interprete de expresiones.

Prolog: Estructura de un programa

Hechos:

docente (persona).
alumno (persona).
enseña (persona, persona).

Características:

- Los nombres de objetos y relaciones deben comenzar con una letra minúscula.
- Primero se escribe la relación, y luego los objetos separados por comas y encerrados entre paréntesis.
- Al final de un hecho debe ir el carácter "." (punto).
- El orden de los objetos dentro de la relación es arbitrario.

Prolog: Estructura de un programa

Reglas:

alumno (persona) :- cursaMateria(persona).

Características:

- Una regla consiste en una cabeza y un cuerpo, unidos por el signo " :- ".
- La cabeza está formada por un único hecho.
- El cuerpo puede ser uno o más hechos (conjunción de hechos), separados por un conector lógico.
- Las reglas finalizan con un punto (".").

Prolog: Estructura de un programa

Preguntas o Consultas:

- Las consultas permiten obtener información sobre los objetos y relaciones existentes.
- Cuando se hace una consulta a Prolog, éste efectuará una búsqueda por toda la Base de Conocimiento intentando encontrar hechos que coincidan con la pregunta.
- Prolog considera que todo lo que hay en la Base de conocimiento es verdad, y lo que no, es falso. De manera que si Prolog responde “yes” es que ha podido demostrarlo, y si no, es que no lo ha podido demostrar.

Prolog: Estructura

Hechos

(afirmación incondicional)

B<-

alumno(ana)

Reglas

(afirmación condicional)

B<-A1,A2,...,An

alumno(ana) :- cursaMateria(ana)

Consultas

(objetivo)

<-A1,A2,...,An

exactas: ? alumno(ana).

true o false

con variables: ? alumno(X).

X=ana

archivo.pl

interprete de
comandos

Ejemplificación en Prolog

Definición de hechos y reglas en SWI-Prolog:

- Para editar un archivo: `?- edit(file('Ejercicio1.pl'))`.

`alumno (jose).`

`alumno (ana).`

`alumno (juan).`

`aprobo_primer_parcial(jose).`

`aprobo_primer_parcial(juan).`

`aprobo_primer_parcial(ana).`

`aprobo_segundo_parcial(jose).`

`aprobo_segundo_parcial(ana).`

`alumno_regular (X) :- alumno (X),`

`aprobo_primer_parcial(X),`

`aprobo_segundo_parcial(X).`

hechos

regla

Ejemplificación en Prolog

Consultas:

- Abrimos el archivo para consultar:
? consult('Ejercicio1.pl').
- ?- alumno_regular(jose).
true
- ?- alumno_regular(juan).
false
- ?- alumno_regular(X).
X = jose
X = ana
2 soluciones.