



UNIVERSIDAD DE CÓRDOBA
ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE
INFORMÁTICA Y ANÁLISIS NUMÉRICO

PROGRAMACIÓN DECLARATIVA

INGENIERÍA INFORMÁTICA

CUARTO CURSO

PRIMER CUATRIMESTRE



Tema 8.- Introducción al Lenguaje Prolog



Primera
parte:
Scheme

Tema 1.- Introducción al Lenguaje Scheme

Tema 2.- Expresiones y Funciones

Tema 3.- Predicados y sentencias condicionales

Tema 4.- Iteración y Recursión

Tema 5.- Tipos de Datos Compuestos

Tema 6.- Abstracción de Datos

Tema 7.- Lectura y Escritura

Segunda
parte: Prolog

Tema 8.- **Introducción al Lenguaje Prolog**

Tema 9.- Elementos Básicos de Prolog

Tema 10.- Listas

Tema 11.- Reevaluación y el “corte”

Tema 12.- Entrada y Salida

Segunda parte: Prolog

Tema 8.- Introducción al Lenguaje Prolog

Tema 9.- Elementos Básicos de Prolog

Tema 10.- Listas

Tema 11.- Reevaluación y el “corte”

Tema 12.- Entrada y Salida

Índice

1. Características Fundamentales de la Programación Lógica
2. Hechos
3. Preguntas
4. Variables
5. Conjunción, disyunción y negación
6. Reglas

Índice

1. Características Fundamentales de la Programación Lógica
2. Hechos
3. Preguntas
4. Variables
5. Conjunción, disyunción y negación
6. Reglas

1. Características Fundamentales de la Programación Lógica

- Introducción
- Orígenes de la programación lógica
- Funcionamiento básico de Prolog

1. Características Fundamentales de la Programación Lógica

- **Introducción**
- Orígenes de la programación lógica
- Funcionamiento básico de Prolog

1. Características Fundamentales de la Programación Lógica

- **Introducción**

- Programación declarativa
- Programación lógica
- Conceptos comunes de la programación lógica y la programación funcional

1. Características Fundamentales de la Programación Lógica

- **Introducción**

- **Programación declarativa**
- Programación lógica
- Conceptos comunes de la programación lógica y la programación funcional

1. Características Fundamentales de la Programación Lógica

- **Introducción**

- **Programación declarativa**

- Su prioridad es responder a la siguiente pregunta

*¿**Qué** problema hay que resolver?*

- **No** se preocupa de

*¿**Cómo** hay que resolver el problema?*

1. Características Fundamentales de la Programación Lógica

- **Introducción**

- **Programación declarativa**

- **Tipos de programación declarativa**

- **Programación funcional**

- ✓ Basada en los conceptos de **función y expresión matemática.**

- **Programación Lógica**

- ✓ Basada en los fundamentos teóricos de la **lógica matemática.**

1. Características Fundamentales de la Programación Lógica

- **Introducción**

- Programación declarativa
- **Programación lógica**
- Conceptos comunes de la programación lógica y la programación funcional

1. Características Fundamentales de la Programación Lógica

- **Introducción**

- **Programación lógica**

- **Programa = Lógica + Control**

- **Lógica**

- ✓ Declaración de **qué** problema debe resolverse.

- ✓ Responsabilidad del **programador.**

- **Control**

- ✓ Declaración de **cómo** debe resolverse el problema.

- ✓ Responsabilidad del **intérprete.**

1. Características Fundamentales de la Programación Lógica

- **Introducción**

- **Programación lógica**

- **Hecho:** cosa que sucede.
 - **Regla:** relación entre hechos.

1. Características Fundamentales de la Programación Lógica

- **Introducción**

- Programación declarativa
- Programación lógica
- **Conceptos comunes de la programación lógica y la programación funcional**

1. Características Fundamentales de la Programación Lógica

- **Introducción**

- **Conceptos comunes de la programación lógica y la programación funcional**

- Recursión
- Listas
- Lenguajes interpretados
- Gestión automática de memoria:
 - ❑ *garbage collection* (recogida de basura).

1. Características Fundamentales de la Programación Lógica

- Introducción
- **Orígenes de la programación lógica**
- Funcionamiento básico de Prolog

1. Características Fundamentales de la Programación Lógica

- **Orígenes de la programación lógica**
 - Fundamentos teóricos
 - Precursores
 - Desarrollo histórico

1. Características Fundamentales de la Programación Lógica

- **Orígenes de la programación lógica**
 - **Fundamentos teóricos**
 - Precursores
 - Desarrollo histórico

1. Características Fundamentales de la Programación Lógica

- **Orígenes de la programación lógica**
 - **Fundamentos teóricos**
 - **Cláusulas de Horn**
 - **Principio de resolución de Robinson**

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica

- Fundamentos teóricos

- Cláusulas de Horn

- Alfred Horn (1951)

- Disyunción de literales negados, excepto uno:

$$\neg p_1 \vee \neg p_2 \vee \dots \neg p_n \vee q$$

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica

- Fundamentos teóricos

- Cláusulas de Horn

- Interpretación

$$\neg p_1 \vee \neg p_2 \vee \dots \vee \neg p_n \vee q$$

(aplicando la Ley de Morgan)

$$\equiv \neg (p_1 \wedge p_2 \wedge \dots \wedge p_n) \vee q$$

(aplicando la equivalencia de la conectiva “ \rightarrow ”)

$$\equiv p_1 \wedge p_2 \wedge \dots \wedge p_n \rightarrow q$$

Significa que si se verifican p_1, p_2, \dots y p_n entonces se verifica q .

1. Características Fundamentales de la Programación Lógica

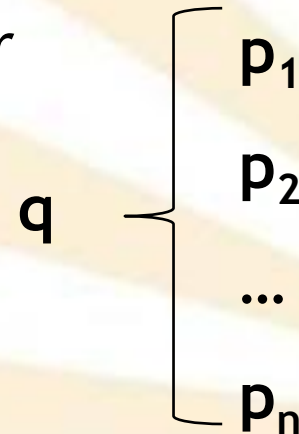
- Orígenes de la programación lógica

- Fundamentos teóricos

- Cláusulas de Horn

- q se verifica si se verifican p_1, p_2, \dots y p_n .

- Interpretación modular

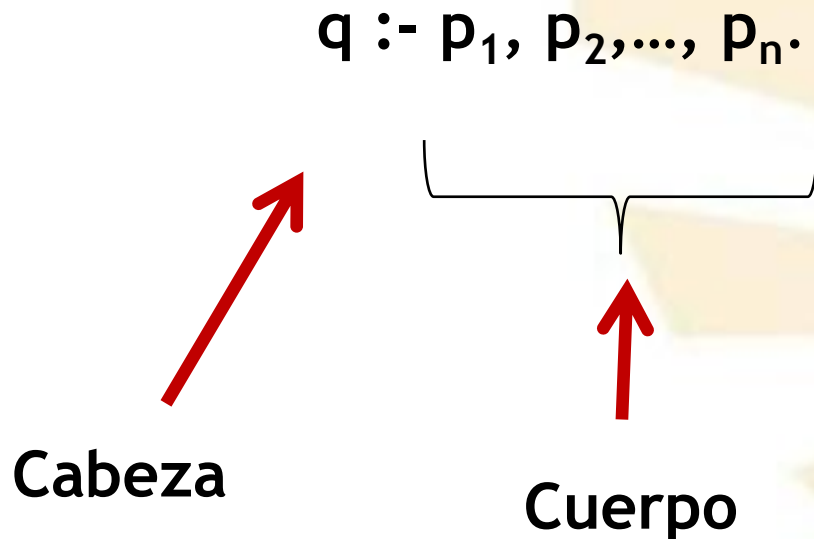


- Regla de Prolog

$$q \text{ :- } p_1, p_2, \dots, p_n.$$

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica
 - Fundamentos teóricos
 - Cláusulas de Horn
 - Regla de Prolog



1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica
 - Fundamentos teóricos
 - Cláusulas de Horn: ejemplo 1

*Si la figura posee cuatro lados iguales
y cuatro ángulos iguales
entonces la figura es un cuadrado*

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica

- Fundamentos teóricos

- Cláusulas de Horn: ejemplo 1

- Variables proposicionales

p_1 = la figura posee cuatro lados iguales

p_2 = la figura posee cuatro ángulos iguales

q = la figura es un cuadrado

- Significado

- ✓ Si se verifican p_1 y p_2 entonces se verifica q

- ✓ q se verifica si se verifican p_1 y p_2

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica

- Fundamentos teóricos

- Cláusulas de Horn: ejemplo 1

- Lógica proposicional

$$p_1 \wedge p_2 \rightarrow q$$

$$\equiv \neg(p_1 \wedge p_2) \vee q$$

$$\equiv \neg p_1 \vee \neg p_2 \vee q \quad \text{Cláusula de Horn}$$

- Cláusula de Horn en Prolog

$$q \text{ :- } p_1, p_2.$$

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica

- Fundamentos teóricos

- Cláusulas de Horn: ejemplo 2

Todos los hombres son mortales

- Lógica de predicados

- Predicados

- ✓ *hombre(x): x es un hombre*

- ✓ *mortal (x): x es mortal*

- Regla

- $\forall x (\text{hombre}(x) \rightarrow \text{mortal}(x))$

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica

- Fundamentos teóricos

- Cláusulas de Horn: ejemplo 2

- Significado

- Si se verifica *hombre* (x) entonces se verifica *mortal* (x)

- Significado equivalente

- *mortal* (x) si se verifica *hombre* (x)

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica

- Fundamentos teóricos

- Cláusulas de Horn: ejemplo 2

- Lógica de predicados

$\forall x (hombre(x) \rightarrow mortal(x))$

$hombre(x) \rightarrow mortal(x)$

- ✓ Cláusula de Horn

$\neg hombre(x) \vee mortal(x)$

- Regla de Prolog

$mortal(X) :- hombre(X).$

1. Características Fundamentales de la Programación Lógica

- **Orígenes de la programación lógica**
 - **Fundamentos teóricos**
 - Cláusulas de Horn
 - **Principio de resolución de Robinson**

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica

- Fundamentos teóricos

- Principio de resolución de Robinson

- Propuesto por John Allan Robinson en 1965



Dr. Honoris Causa de la Universidad Politécnica de Madrid (1 de octubre de 2013)

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica
 - Fundamentos teóricos
 - Principio de resolución de Robinson
 - Caso básico (lógica proposicional)

Modus Ponens

$$p \rightarrow q$$

$$p$$

$$q$$

Principio de resolución

$$\neg p \vee q$$

$$p$$

$$q$$

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica

- Fundamentos teóricos

- Principio de resolución de Robinson

- Caso básico (lógica proposicional)

- ✓ Ejemplo

Si es un cuadrado entonces es un polígono

Es un cuadrado

Es un polígono

Proposiciones

➤ p: es un cuadrado

➤ q: es un polígono

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica
 - Fundamentos teóricos
 - Principio de resolución de Robinson
 - Caso básico (lógica proposicional)
 - ✓ Ejemplo

Modus Ponens

$$p \rightarrow q$$

$$p$$

$$q$$

Principio de resolución

$$\neg p \vee q$$

$$p$$

$$q$$

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica
 - Fundamentos teóricos
 - Principio de resolución de Robinson
 - Caso **general** (lógica proposicional)

$$p_1 \wedge p_2 \wedge \dots \wedge p_n \rightarrow q$$

$$p_1$$

$$p_2 \wedge \dots \wedge p_n \rightarrow q$$

Principio de resolución

$$\neg p_1 \vee \neg p_2 \vee \dots \vee \neg p_n \vee q$$

$$p_1$$

$$\neg p_2 \vee \dots \vee \neg p_n \vee q$$

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica
 - Fundamentos teóricos
 - Principio de resolución de Robinson
 - Caso básico (lógica de predicados)

Modus Ponens

$$p(x) \rightarrow q(x)$$

$$p(a)$$

$$q(a)$$

Principio de resolución

$$\neg p(x) \vee q(x)$$

$$p(a)$$

$$q(a)$$

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica
 - Fundamentos teóricos
 - Principio de resolución de Robinson
 - Caso básico (lógica de predicados)
 - ✓ Ejemplo

Todos los hombres son mortales

Sócrates es un hombre

Sócrates es mortal

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica
 - Fundamentos teóricos
 - Principio de resolución de Robinson
 - Caso básico (lógica de predicados)
 - ✓ Ejemplo

Modus Ponens

hombre(x) → mortal(x)

hombre(Sócrates)

mortal(Sócrates)

Principio de resolución

¬ hombre(x) ∨ mortal(x)

hombre(Sócrates)

mortal(Sócrates)

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica
 - Fundamentos teóricos
 - Principio de resolución de Robinson
 - Caso **general** (lógica de **predicados**)

$$p_1(x) \wedge p_2(x) \wedge \dots \wedge p_n(x) \rightarrow q(x)$$

$$p_1(a)$$

$$p_2(a) \wedge \dots \wedge p_n(a) \rightarrow q(a)$$

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica

- Fundamentos teóricos

- Principio de resolución de Robinson

- Caso general (lógica de predicados)

Principio de resolución

$$\neg p_1(x) \vee \neg p_2(x) \vee \dots \vee \neg p_n(x) \vee q(x)$$

$$p_1(a)$$

$$\neg p_2(a) \vee \dots \vee \neg p_n(a) \vee q(a)$$

1. Características Fundamentales de la Programación Lógica

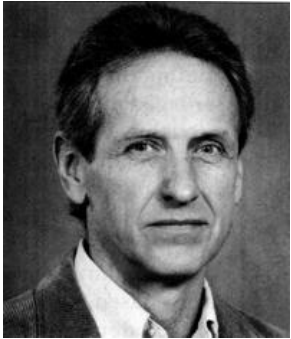
- **Orígenes de la programación lógica**
 - Fundamentos teóricos
 - **Precursores**
 - Difusión histórica

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica

- Precursores

- **Robert Kowalski** (Universidad de Edimburgo):
1972



- Fundamentos teóricos

- ✓ Cláusulas de Horn y principio de resolución

- **Maarten Van Emden** (Universidad de Edimburgo):



- Demostración experimental

1. Características Fundamentales de la Programación Lógica

- Orígenes de la programación lógica

- Precursores

- Alain Colmerauer y Phillipe Roussel

(Universidad de Marsella)



Colmerauer

- Implementación

- ✓ *Programmation en Logique: Prolog*

- Intérprete de Prolog codificado en Algol W de Niklaus Wirth.

- Utiliza la interpretación procedimental o modular de Kowalski.



Roussel

1. Características Fundamentales de la Programación Lógica

- **Orígenes de la programación lógica**
 - Fundamentos teóricos
 - Precursores
 - **Difusión histórica**

1. Características Fundamentales de la Programación Lógica

- **Orígenes de la programación lógica**

- **Difusión histórica**

- **Prolog-10 (Warren, años setenta)**

- Compilador escrito casi completamente en Prolog.

- **Desdén inicial**

- La comunidad científica **occidental** no tuvo en cuenta a la Programación Lógica hasta principios de los años ochenta.

- **Factor decisivo**

- Proyecto de Quinta Generación de Japón (1981).

- **Versión estándar: Prolog de Edimburgo**

1. Características Fundamentales de la Programación Lógica

- Introducción
- Orígenes de la programación lógica
- **Funcionamiento básico de Prolog**

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**
 - Ejecución
 - Búsqueda de objetivos
 - Limitaciones
 - Aplicaciones

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**
 - **Ejecución**
 - Búsqueda de objetivos
 - Limitaciones
 - Aplicaciones

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Ejecución**

- Llamada al intérprete
- Carga de un fichero con hechos y reglas
- Preguntas
- Salida del intérprete

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Ejecución**

- Llamada al intérprete

pl (alternativas: *prolog*, *swipl*)

?-

- Salida del intérprete

?- halt.

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Ejecución**

- Carga de un fichero con **hechos y reglas**

- Ejecutar swipl y cargar el fichero.pl

\$ swipl -s fichero.pl

?-

- Cargar el fichero.pl

?- consult('fichero').

?- consult('fichero.pl').

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Ejecución**

- Carga de un fichero con **hechos y reglas**

- Cargar el fichero.pl: manera alternativa

- ?- [fichero].*

- Cargar *fichero_i.pl* ($\forall i \in \{1,2,\dots,n\}$)

- ?- [fichero₁, fichero₂, ..., fichero_n].*

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Ejecución**

- Carga de un fichero con hechos y reglas desde el teclado

?- [user].

...

<<fin de fichero>>

?-

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Ejecución**

- Listado de todos los hechos y reglas

?- listing.

- Listado de todos los hechos y reglas de un predicado

?- listing(predicado).

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Ejecución**

- Directorio actual

- ?- *pwd*.

- Contenido del directorio actual:

- ?- *ls*.

- Cambio al directorio “padre”

- ?- *cd('..')*.

- Cambio al directorio home/i99zazaz/ejemplo

- ?- *cd('/home/i99zazaz/ejemplo')*.

- Cambio al directorio de Windows C:\ejemplo

- ?- *cd('c:\ejemplo')*.

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Ejecución**

- Ayuda

- ?- *help(Topic)*

- ?- *apropos(Word).*

- Traza de ejecución del intérprete

- ?- *trace.*

Fin de la traza:

- ?- *nodebug.*

Call: (6) edinburgh:nodebug ? n (no debug)

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Ejecución**

- Traza mediante ventana gráfica

?- *guitracer*.

La próxima ejecución de trace será en una ventana gráfica.

- Fin de la traza mediante ventana gráfica

?- *noguitracer*.

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Ejecución**

- **Carga de un fichero con hechos y reglas**

- Los hechos son descritos en los apartados nº 2, 4, 5.

- Las reglas son descritas en el apartado nº 6.

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Ejecución**

- **Preguntas**

- Permite consultar si un hecho se puede deducir de la base de hechos y reglas del programa.
- Véase el apartado nº 3.

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**
 - Ejecución
 - **Búsqueda de objetivos**
 - Limitaciones
 - Aplicaciones

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Búsqueda de objetivos**

- **Programa**

- Conjunto de **hechos y reglas**

- **Ejecución de una consulta**

- Si la **consulta** coincide con un **hecho** entonces el **objetivo** se ha **conseguido**.

- En caso contrario, se comprueba si la consulta es la “**cabeza**” de una **regla**

- ✓ Si es así entonces se generan **nuevas consultas** utilizando el **cuerpo** de la regla.

- ✓ En caso contrario, el objetivo no se consigue.

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Búsqueda de objetivos**

- **Ejemplo1**

- **Hechos**

hombre(socrates).

hombre(aristoteles).

- **Regla**

mortal(X) :- hombre(X).

- **Consulta**

?- mortal(socrates).

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Búsqueda de objetivos**

- **Ejemplo1**

- Equivalencia con la lógica de predicados (1/2)

$\forall x (hombre(x) \rightarrow mortal(x))$

$hombre(socrates)$

$mortal(socrates)$

$hombre(x) \rightarrow mortal(x)$

$hombre(socrates)$

$mortal(socrates)$

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Búsqueda de objetivos**

- **Ejemplo1**

- Equivalencia con la lógica de predicados (2/2)

Principio de resolución

```
¬ hombre(x) ∨ mortal(x)
hombre(socrates)
──────────────────
mortal(socrates)
```

Prolog

```
mortal(X) :- hombre(X).
hombre(socrates).
?- mortal(socrates).
yes
```

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Búsqueda de objetivos**

- **Ejemplo1**

- **Consulta**

?- *mortal(socrates)*.

1. **No** existe el hecho *mortal (socrates)*

2. La consulta es la **cabeza** de la regla

mortal(X) :- hombre(X)

3. Se genera una **nueva consulta**:

hombre(socrates)

4. *hombre(socrates)* es un hecho del programa.

5. Respuesta: **sí** (yes)

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Búsqueda de objetivos**

- **Observación**

- Prolog considera como **falso** todo lo que **no conoce** o no puede deducir, aunque sea verdadero en la vida real.

- Ejemplo: si se consulta

?- mortal(platon).

la respuesta será **no**

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**
 - Ejecución
 - Búsqueda de objetivos
 - **Limitaciones**
 - Aplicaciones

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Limitaciones**

- Prolog **no** es un lenguaje de programación lógica completamente puro:
 - ❑ Prolog establece un **orden** de precedencia entre reglas, que no existe en la lógica.
 - ❑ Prolog sólo permite la **negación** en el **cuerpo** de las cláusulas.

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**
 - Ejecución
 - Búsqueda de objetivos
 - Limitaciones
 - **Aplicaciones**

1. Características Fundamentales de la Programación Lógica

- **Funcionamiento básico de Prolog**

- **Aplicaciones**

- Demostración de teoremas
- Representación del conocimiento
- Problemas de búsqueda
- Planificación
- Procesamiento del lenguaje natural (PLN)
- Sistemas expertos
- Especificación de algoritmos
- Etc.

Índice

1. Características Fundamentales de la Programación Lógica
2. **Hechos**
3. Preguntas
4. Variables
5. Conjunción, disyunción y negación
6. Reglas

2. Hechos

- Definición
- Ejemplos
- Sintaxis
- Semántica

2. Hechos

- **Definición**
- Ejemplos
- Sintaxis
- Semántica

2. Hechos

- **Definición**

- Un hecho representa una afirmación que se realiza sobre un asunto.

2. Hechos

- Definición
- **Ejemplos**
- Sintaxis
- Semántica

2. Hechos

- **Ejemplos**

- **Hechos en Prolog**

- hombre(socrates).*

- sexo(socrates, masculino).*

- padre_de(juan, luis).*

- tiene(antonio, bicicleta).*

- da(laura, libro, maria).*

- **Significados de los hechos**

- Sócrates es un hombre

- El padre de Juan es Luis

- Antonio tiene una bicicleta

- Laura da un libro a María

2. Hechos

- **Ejemplos**

- **Hechos**

- Juan está enamorado de María
- María está enamorada de Juan
- Laura está enamorada de Luis

- **Hechos en Prolog**

enamorado_de(juan, maria).

enamorado_de(maria,juan).

enamorado_de(laura, luis).

2. Hechos

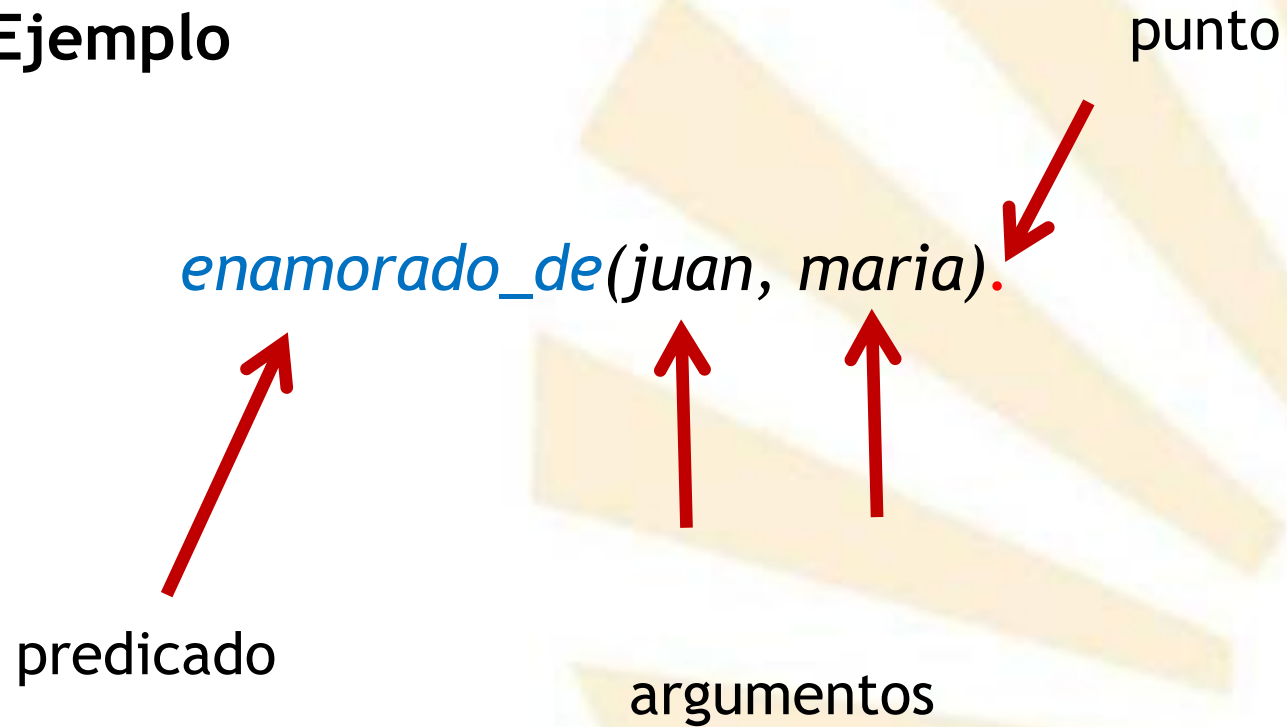
- Sintaxis

predicado (*argumento*₁, *argumento*₂, ... *argumento*_n).

2. Hechos

- Sintaxis

- Ejemplo



2. Hechos

- **Sintaxis**

- **Restricciones**

- Los nombres de los **predicados** y los **átomos** comienzan por una letra **minúscula**.
- Se tiene que poner un **punto “.”** al final del hecho.

2. Hechos

- **Sintaxis**

- **Restricciones**

- **Número de argumentos de un predicado**

- Constante.

- **Observación**

- Puede haber predicados distintos con el mismo nombre y diferente número de argumentos.

- tiene**(juan, bicicleta).

- tiene**(ana, bicicleta)

- tiene**(juan, casa, grande).

2. Hechos

- **Semántica**

- Cada **hecho** posee un **significado arbitrario**.
- El **significado** de un hecho debe permanecer **invariable** una vez que ha sido establecido.
- El **predicado** establece una **relación** entre los argumentos.
- El significado de un predicado depende del **orden** de los argumentos.
- Se deben utilizar **identificadores descriptivos** de los predicados y los argumentos.

2. Hechos

- **Semántica**
 - **Base de datos** de un programa en Prolog:
 - Conjunto de **hechos** y **reglas** que posee.

Índice

1. Características Fundamentales de la Programación Lógica
2. Hechos
- 3. Preguntas**
4. Variables
5. Conjunción, disyunción y negación
6. Reglas

3. Preguntas

- Se pueden realizar preguntas a la base de datos de un programa de Prolog

- **Ejemplo:**

?- tiene(maria,libro).

punto



- Se está preguntando si María tiene el libro.
- Prolog busca en su base de datos, compuesta por hechos (y reglas), si contiene el hecho

tiene(maria,libro).

- Es necesario que **coincidan**
 - el nombre del **predicado** o relación
 - los **argumentos** en número y orden.

3. Preguntas

- **Ejemplo**

- **Hechos**

tiene(maria, libro).

tiene(maria, cuaderno).

tiene(juan, cuaderno).

tiene(juan, bicicleta).

- **Preguntas**

?- tiene(maria, libro).

yes

?- tiene(juan, libro).

no

3. Preguntas

- **Ejemplo**

- **Pregunta**

?- *tiene(maria, bicicleta).*

no

- **Falso por defecto**

- Prolog responde “no” porque

- no hay ningún hecho que coincida con la pregunta

- ni regla que permita deducir ese hecho.

- Se recuerda que **Prolog** considera **falso** todo lo que **no conoce**.

Índice

1. Características Fundamentales de la Programación Lógica
2. Hechos
3. Preguntas
4. **Variables**
5. Conjunción, disyunción y negación
6. Reglas

4. Variables

- Las variables permiten hacer preguntas más versátiles
- Primer carácter de una variable:
 - Letra mayúscula
 - Subrayado “_”
- Ejemplos
 - ¿Qué cosas tiene Juan?

?- tiene(juan, Cosa).

Cosa = cuaderno

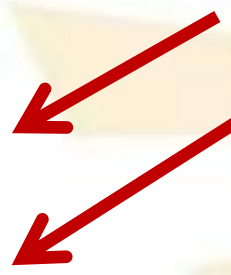
:
;

Cosa = bicicleta

:
;

no

Punto y coma:
nueva solución



4. Variables

- Ejemplos

- ¿Quién tiene un cuaderno?

?- tiene(_persona,cuaderno).

Punto y coma:
nueva solución

_persona = maria

;

_persona = juan

Enter

Fin de la búsqueda

4. Variables

- Preguntas con dos variables
 - ¿Qué cosas tiene cada persona?

?- tiene(_persona, Cosa).

_persona = maria, Cosa = libro

;

_persona = maria, Cosa = cuaderno

;

_persona = juan, Cosa = cuaderno

;

_persona = juan, Cosa = bicicleta

Enter

4. Variables

- Preguntas con una variable repetida.

- Hechos

camino(baena, luque).

camino(luque, zuheros).

camino(baena, baena).

- ¿Qué pueblo tiene un camino circular?

?- *camino(Pueblo, Pueblo).*

Pueblo= baena

4. Variables

- **Variable anónima: símbolo de subrayado “_”**
 - Comprueba **si existe un valor** que se pueda asociar a la variable que haga verdadero el predicado.
 - **No importa** conocer cuál es el valor concreto asociado a la variable anónima.
 - Cada aparición de una variable anónima es **independiente**.

4. Variables

- Variable anónima: símbolo de subrayado “ ”

- Preguntas con variable anónima

- ¿Quién tiene algo?

?- *tiene(Persona,)*.

Persona = maria

;

Persona = juan

Enter

- ¿Alguien tiene un cuaderno?

?- *tiene(,cuaderno)*.

yes

4. Variables

- Variable anónima: símbolo de subrayado “ ”
 - Preguntas con variable anónima
 - ¿Alguien tiene algo?
?- tiene(,).
yes

Índice

1. Características Fundamentales de la Programación Lógica
2. Hechos
3. Preguntas
4. Variables
- 5. Conjunción, disyunción y negación**
6. Reglas

5. Conjunciones, disyunciones y negaciones

- Se pueden hacer preguntas más completas si se utilizan los **operadores lógicos**
 - Disyunción lógica:
 - Símbolo de punto y coma “;”
 - Conjunción lógica:
 - Símbolo de la coma “,”
 - Negación lógica:
 - **not**

5. Conjunciones, disyunciones y negaciones

- **Conjunción lógica**

- La respuesta será verdadera si ambas respuestas son verdaderas

- **Ejemplos**

?- *tiene(maria,libro) , tiene(juan,cuaderno).*

yes

?- *tiene(maria,bicicleta) , tiene(juan,cuaderno).*

no

5. Conjunciones, disyunciones y negaciones

- **Conjunción lógica**

- ¿Qué cosa tienen en común María y Juan?

?- *tiene(maria,X) , tiene(juan,X)*.

X = cuaderno

- **Reevaluación o *backtracking* (lógica + control)**

- Fase 1: X toma el valor de **libro**

- Fase 2: **no** se verifica *tiene(juan,libro)*

- Fase 3: X toma el valor de *cuaderno*

- Fase 4: se **verifica** el hecho

tiene(juan,cuaderno)

5. Conjunciones, disyunciones y negaciones

- **Conjunción lógica**

- ¿Tiene algo María y tiene algo Juan?

- ?- *tiene(maria,_) , tiene(juan,_)*.

- yes*

5. Conjunciones, disyunciones y negaciones

- **Conjunción lógica**

- Hechos

enamorado_de(juan, maria).

enamorado_de(maria,juan).

enamorado_de(laura, luis).

- ¿Están Juan y María enamorados?

?- *enamorado_de (juan,maria),*

enamorado_de(maria,juan).

- ¿Es correspondido el amor de Laura?

?- *enamorado_de(laura,X), enamorado_de(X,laura).*

- Etc.

5. Conjunciones, disyunciones y negaciones

- **Disyunción lógica**
 - La respuesta será verdadera si alguna de las respuestas es verdadera.

- **Ejemplos**

?- *tiene(maria,libro) ; tiene(juan,cuaderno).*

yes

?- *tiene(maria,bicicleta) ; tiene(juan,cuaderno).*

yes

5. Conjunciones, disyunciones y negaciones

- Ejemplos

- ¿Qué cosas tienen Juan o María?

?- tiene(juan,X) ; tiene(maria,X).

X = cuaderno

X = bicicleta

X = libro

X = cuaderno

no

5. Conjunciones, disyunciones y negaciones

- **Negación lógica**

- La respuesta será verdadera si la respuesta original es falsa.

- **Ejemplos**

?- *not*(tiene(juan, cuaderno))

no

?- *not*(tiene(juan, tijeras)).

yes

5. Conjunciones, disyunciones y negaciones

- **Precedencia de los operadores lógicos**
 - Máxima precedencia:
 - negación lógica (not)
 - Precedencia intermedia:
 - conjunción lógica (,)
 - Mínima precedencia:
 - disyunción lógica (;)
 - Los paréntesis controlan la precedencia.

5. Conjunciones, disyunciones y negaciones

- Precedencia de los operadores lógicos
 - Ejemplos

?- *tiene(juan, cuaderno); tiene(maria, rotulador),
not(tiene(juan, bicicleta)).*

yes

?- *(tiene(juan, cuaderno); tiene(maria, rotulador)),
not(tiene(juan, bicicleta)).*

no

Índice

1. Características Fundamentales de la Programación Lógica
2. Hechos
3. Preguntas
4. Variables
5. Conjunción, disyunción y negación
- 6. Reglas**

6. Reglas

mortal(X) :- hombre(X). ← punto

↑
Cabeza

↑
Cuerpo

6. Reglas

- Las **reglas** evitan indicar hechos que se pueden deducir a partir de otros.
- Las **reglas** permiten representar **dependencias** entre hechos.

mortal(X) :- hombre(X).

- Las **reglas** permiten representar **definiciones**

buen_estudiante(Persona):-

asiste_a_clase(Persona),

estudia(Persona).

6. Reglas

- Reglas recursivas
- Ejemplo de las conexiones por carretera

carretera(cordoba,granada).

carretera(cordoba,jaen).

carretera(cordoba,malaga).

carretera(cordoba,sevilla).

carretera(sevilla, cadiz).

carretera(sevilla, huelva).

carretera(sevilla, malaga).

conectado (Origen, Destino):-

carretera(Origen, Destino).

conectado (Origen, Destino):-

carretera (Origen,Intermedio),

conectado (Intermedio, Destino).

6. Reglas

- Reglas recursivas
- Ejemplo de “la familia” (1/6)

mujer(ana).
mujer(marta).
mujer(alicia).
mujer(carmen).
mujer(laura).
mujer(isabel).
mujer(silvia).

hombre(juan).
hombre(luis).
hombre(miguel).
hombre(alberto).
hombre(rodrigo).
hombre(pedro).

6. Reglas

- Ejemplo de “la familia” (2/6)

*/ * padres(Persona, Padre, Madre). */*

padres(juan,luis,ana).

padres(alberto,luis,ana).

padres(marta,luis,ana).

padres(alicia,luis,ana).

padres(rodrigo,juan,laura).

padres(carmen,juan,laura).

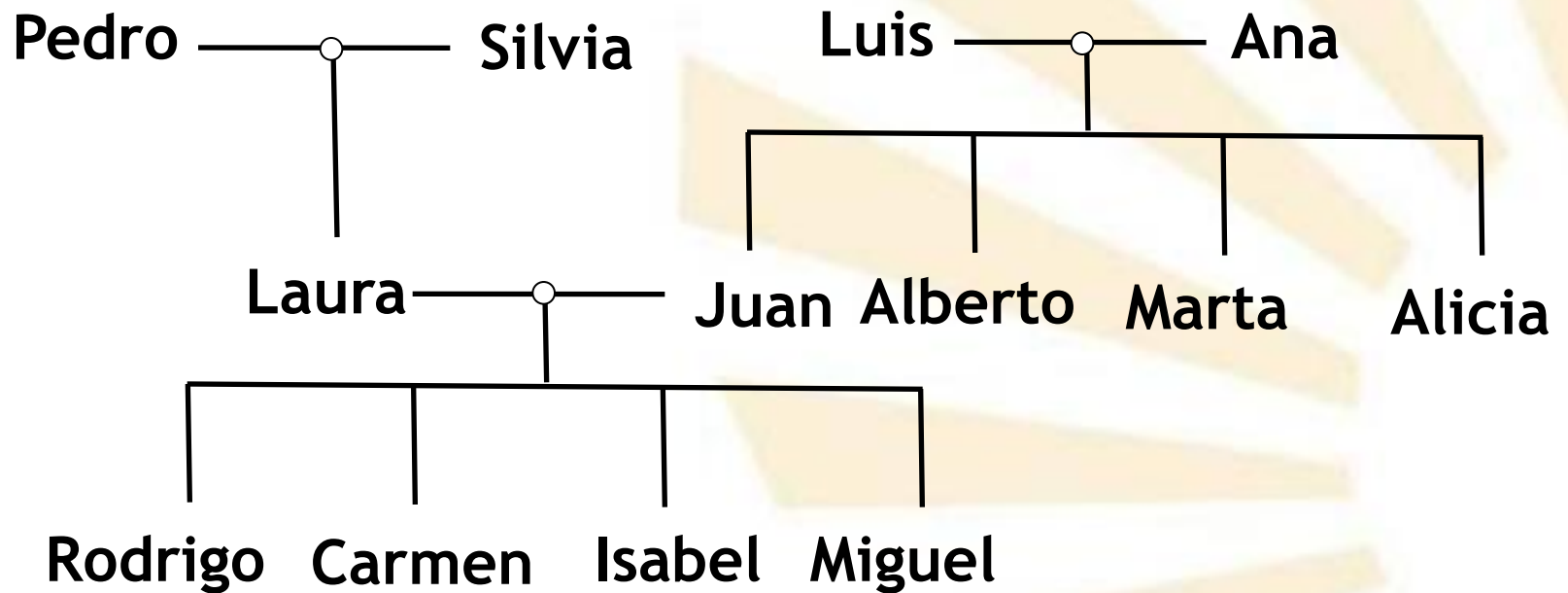
padres(isabel,juan,laura).

padres(miguel,juan,laura).

padres(laura,pedro,silvia).

6. Reglas

- Ejemplo de “la familia” (3/6)
 - Árbol genealógico



6. Reglas

- Ejemplo de “la familia” (4/6)

/ X es hermana de Y: primera versión*/*

hermana(X,Y):-

mujer(X),

padres(X,Padre,Madre),

padres(Y,Padre,Madre).

- **Problema**

?- *hermana(alicia,Y).*

Y = marta

;

Y = alicia

Error

6. Reglas

- Ejemplo de “la familia” (5/6)

/ X es hermana de Y: segunda versión */*

*hermana_verdadera(X,Y):-
mujer(X),
padres(X,Padre,Madre),
padres(Y,Padre,Madre),
X \= Y.*

6. Reglas

- Ejemplo de “la familia” (6/6)

/ X es un ancestro de Y */*

```
ancestro(X,Y):-  
    padres(Y,X,_).
```

```
ancestro(X,Y):-  
    padres(Y,_,X).
```

```
ancestro(X,Y):-  
    padres(Y,Z,_),  
    ancestro(X,Z).
```

```
ancestro(X,Y):-  
    padres(Y,_,Z),  
    ancestro(X,Z).
```

6. Reglas

- **Ejercicios sobre la familia**
 - Abuelo/a
 - Nieto/a
 - Primos/as
 - Etc.

- **Ejercicio de los donantes de sangre**



UNIVERSIDAD DE CÓRDOBA
ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE
INFORMÁTICA Y ANÁLISIS NUMÉRICO

PROGRAMACIÓN DECLARATIVA

INGENIERÍA INFORMÁTICA

CUARTO CURSO

PRIMER CUATRIMESTRE



Tema 8.- Introducción al Lenguaje Prolog

