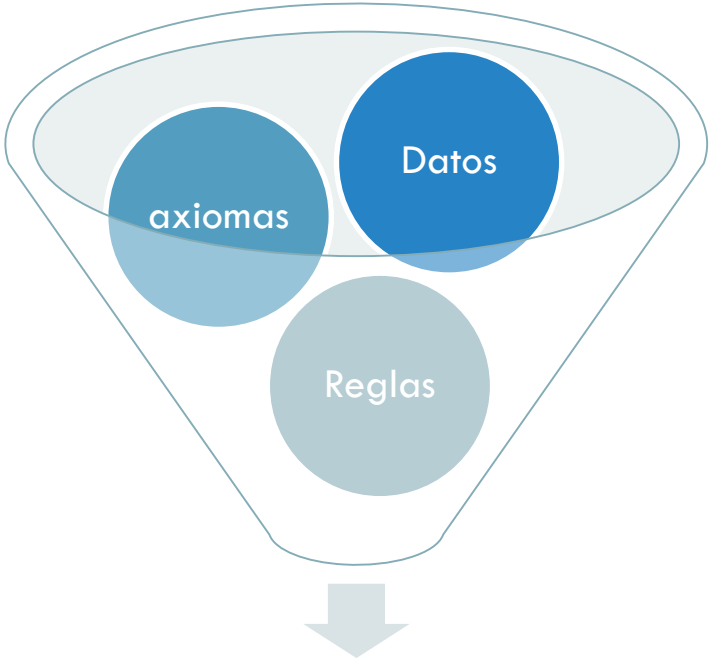
Programación Lógica

- David Barrera
- Steven Bustos
- Jhonatan Guzmán

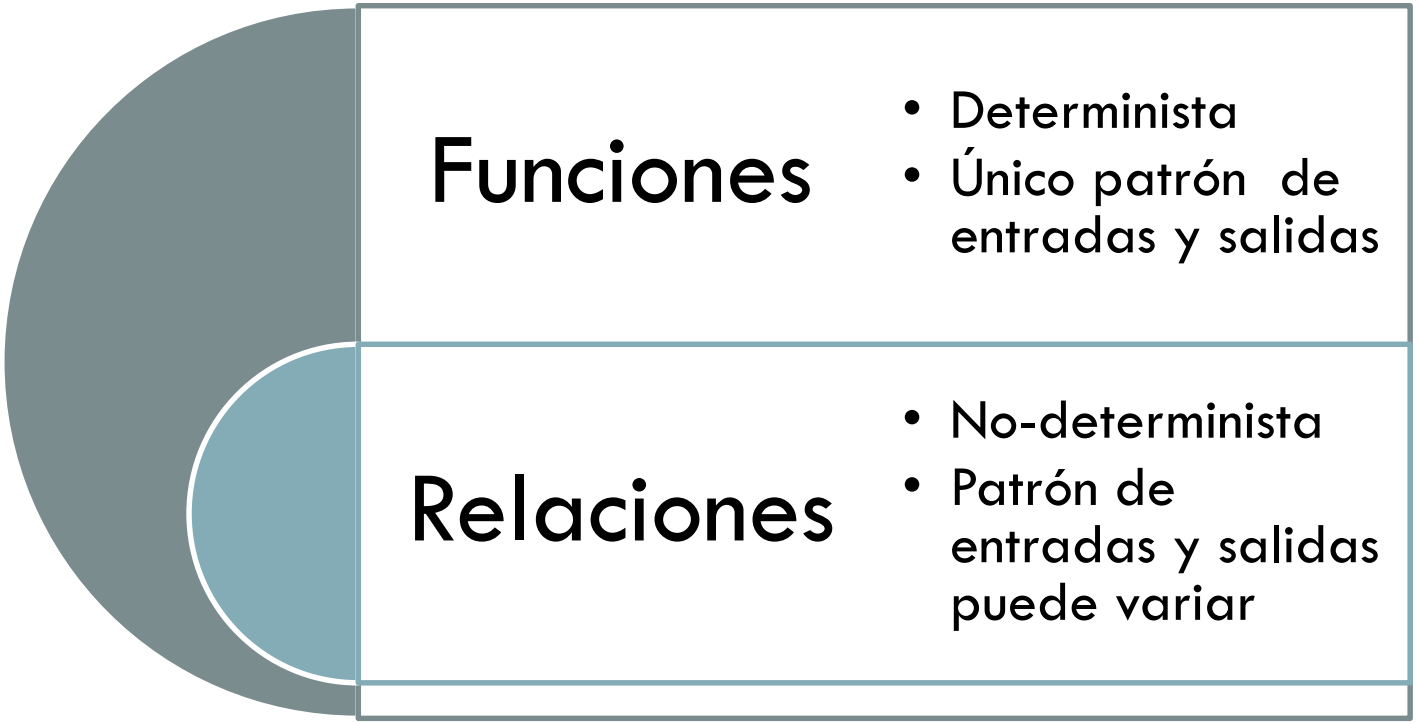
La filosofía de el paradigma se basa en relaciones y consiste en formulas lógicas que describen un problema

Programación Lógica



Programación Lógica

Relaciones vs Funciones



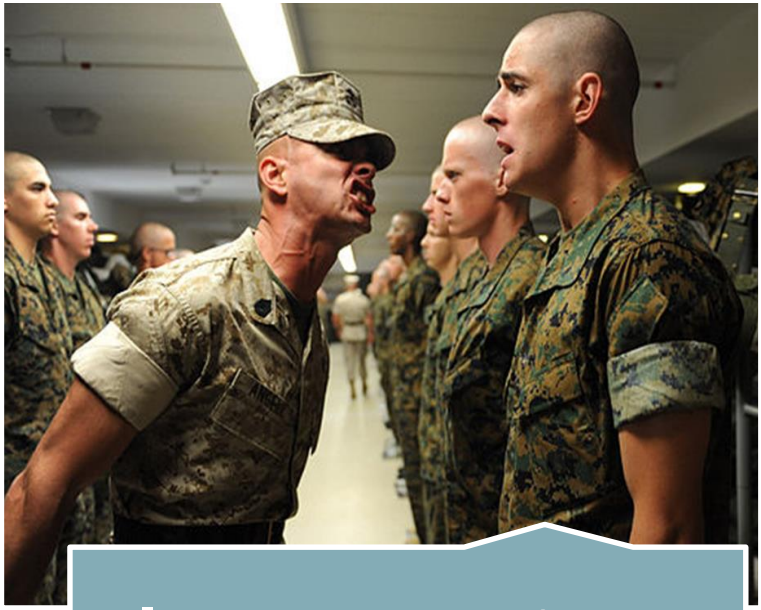
Relaciones vs Funciones

Ejemplo:

- X < Y, la relación ‘<’ puede usarse de 4 formas distintas
  - **Generador** (infinito) de parejas (X ,Y) que cumpla la relación
  - **Predicado** aplicable a parejas (X, Y)
  - **Generador** dado X retorne todos los Y mayores a X
  - **Generador** dado Y retorne todos los X tales que X < Y

Programación declarativa vs imperativa

Declarativa



Imperativa

Programación declarativa vs imperativa

Declarativa

- A donde llegar
- Lógica de computación

Imperativa

- Como llegar
- Sentencias de instrucciones

Algoritmo

(R. Kowalski)

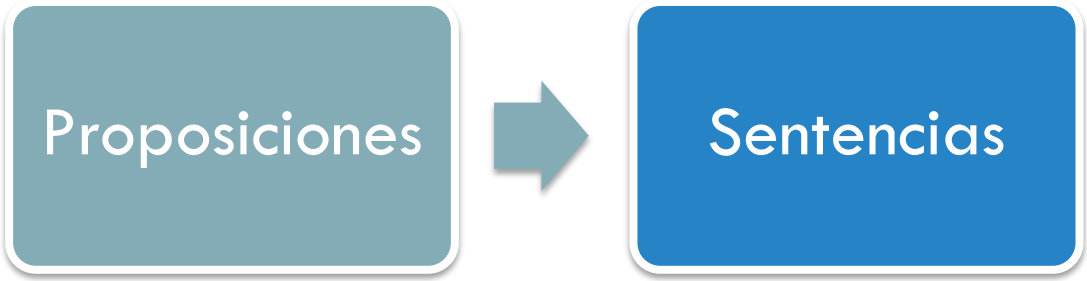
Lógica

Que se debe hacer

Control

Como encontrar la solución

Lógica proposicional



Lógica proposicional

$$
\begin{aligned}
\textit{Sentencia} &\rightarrow \textit{Sentencia Atómica} \mid \textit{Sentencia Compleja} \\
\textit{Sentencia Atómica} &\rightarrow \textbf{Verdadero} \mid \textbf{Falso} \mid \textit{Símbolo Proposicional} \\
\textit{Símbolo Proposicional} &\rightarrow \textbf{P} \mid \textbf{Q} \mid \textbf{R} \mid \ldots \\
\textit{Sentencia Compleja} &\rightarrow \neg\, \textit{Sentencia} \\
&\mid (\textit{Sentencia} \wedge \textit{Sentencia}) \\
&\mid (\textit{Sentencia} \vee \textit{Sentencia}) \\
&\mid (\textit{Sentencia} \Rightarrow \textit{Sentencia}) \\
&\mid (\textit{Sentencia} \Leftrightarrow \textit{Sentencia})
\end{aligned}
$$

Lógica proposicional

- Semántica. dada por una tabla de verdad

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
falso	*falso*	*verdadero*	*falso*	*falso*	*verdadero*	*verdadero*
falso	*verdadero*	*verdadero*	*falso*	*verdadero*	*verdadero*	*falso*
verdadero	*falso*	*falso*	*falso*	*verdadero*	*falso*	*falso*
verdadero	*verdadero*	*falso*	*verdadero*	*verdadero*	*verdadero*	*verdadero*

Lógica de primer orden

- Llamada también lógica de predicados.
- Más expresiva que la lógica proposicional.
- Se define como conjunto de términos, relaciones y funciones.
- Sentencia Atómica: Un predicado, Uno o mas términos.
- Hace uso de conectivas lógicas y cuantificadores.

Lógica de primer orden

$$
\begin{aligned}
\textit{Sentencia} &\rightarrow \textit{SentenciaAtómica} \\
&\mid (\textit{Sentencia Conectiva Sentencia}) \\
&\mid \textit{Cuantificador Variable}\ldots \textit{Sentencia} \\
&\mid \neg \textit{Sentencia} \\
\\
\textit{SentenciaAtómica} &\rightarrow \textit{Predicado}(\textit{Término}\ldots) \mid \textit{Término} = \textit{Término} \\
\\
\textit{Término} &\rightarrow \textit{Función}(\textit{Término}\ldots) \\
&\mid \textit{Constante} \\
&\mid \textit{Variable}
\end{aligned}
$$

Lógica de primer orden

$$
\begin{aligned}
\textit{Conectiva} &\rightarrow \Rightarrow \mid \wedge \mid \vee \mid \Leftrightarrow \\
\textit{Cuantificador} &\rightarrow \forall \mid \exists \\
\textit{Constante} &\rightarrow A \mid X_1 \mid \textit{Juan} \mid \cdots \\
\textit{Variable} &\rightarrow a \mid x \mid s \mid \cdots \\
\textit{Predicado} &\rightarrow \textit{AntesDe} \mid \textit{TieneColor} \mid \textit{EstáLLoviendo} \mid \cdots \\
\textit{Función} &\rightarrow \textit{Madre} \mid \textit{PiernaIzquierda} \mid \cdots
\end{aligned}
$$

Ejemplos

Ricardo y Juan son hermanos

- *Hermano(Ricardo, Juan)*

El padre de Ricardo está casado con la madre de Juan

- *CasadoCon(Padre(Ricardo), Madre(Juan))*

Todo estudiante de la Universidad Nacional es inteligente

- $\forall$ *estudiante Estudia(estudiante,Unal)* $\rightarrow$ *Inteligente(estudiante)*

Clausula de Horn

Clausula (disyunción de literales)

- Poderoso uso en programación lógica y teoría de modelos.
- Alfred Horn (1918 - 2001).

Formula lógica con a lo máximo 1 literal positivo.

$$\neg p \vee \neg q \vee \cdots \vee \neg t \vee u$$

Clausula de Horn

Programación Lógica.

- Forma de implicación.

$$(p \land q \land \cdots \land t) \rightarrow u$$

Clasificación.

- Objetivo o consulta: Ningún literal positivo.
- Definida: Exactamente un literal positivo.

Clausula de Horn

- **Programación lógica.**
  - Clausula definida : procedimiento.

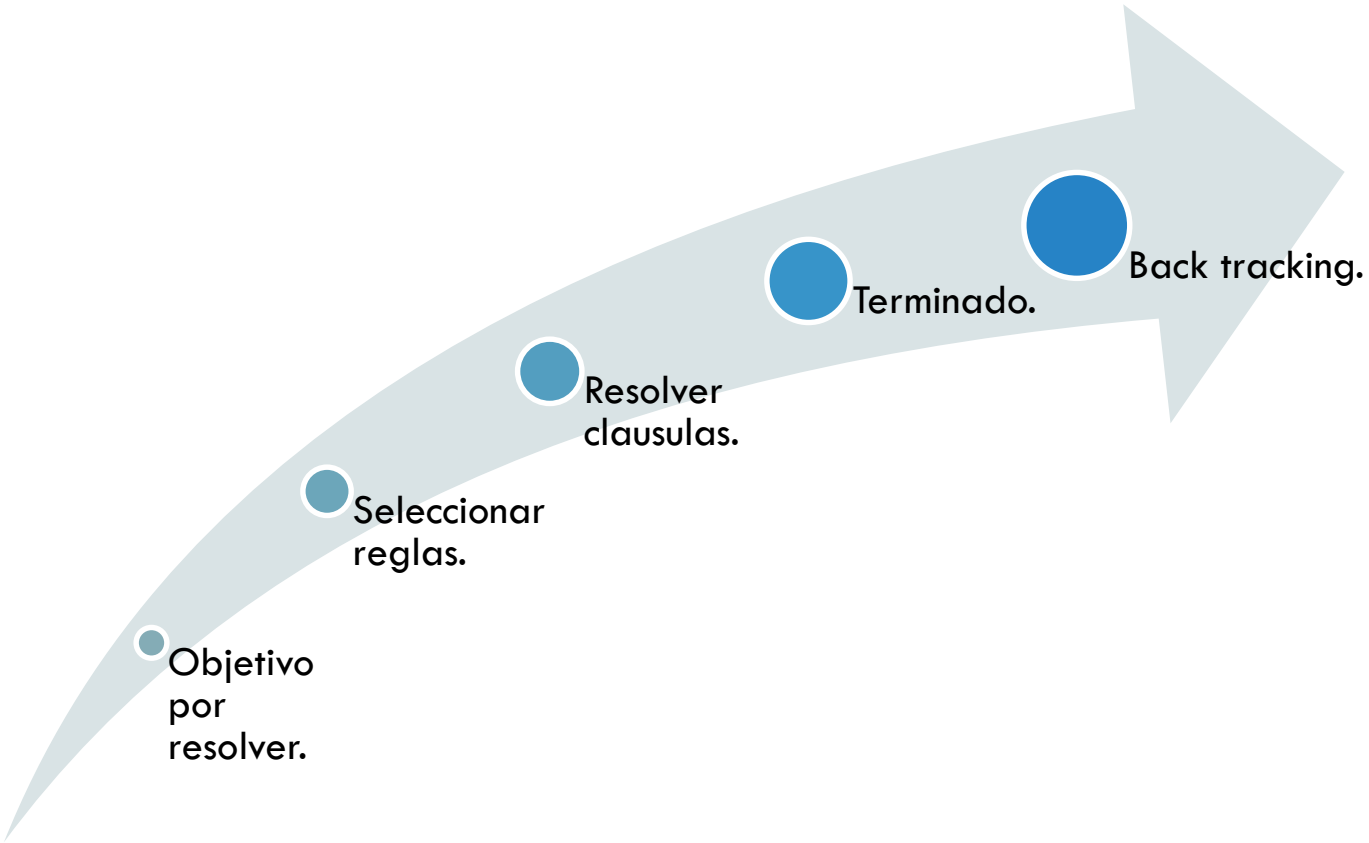
$$(p \land q \land \cdots \land t) \rightarrow u$$

  - Resolución SLD
  - Consecuente y antecedente.
  - Ejemplo:

    mascota(A,B) :- domestico(A), amo(B,A).

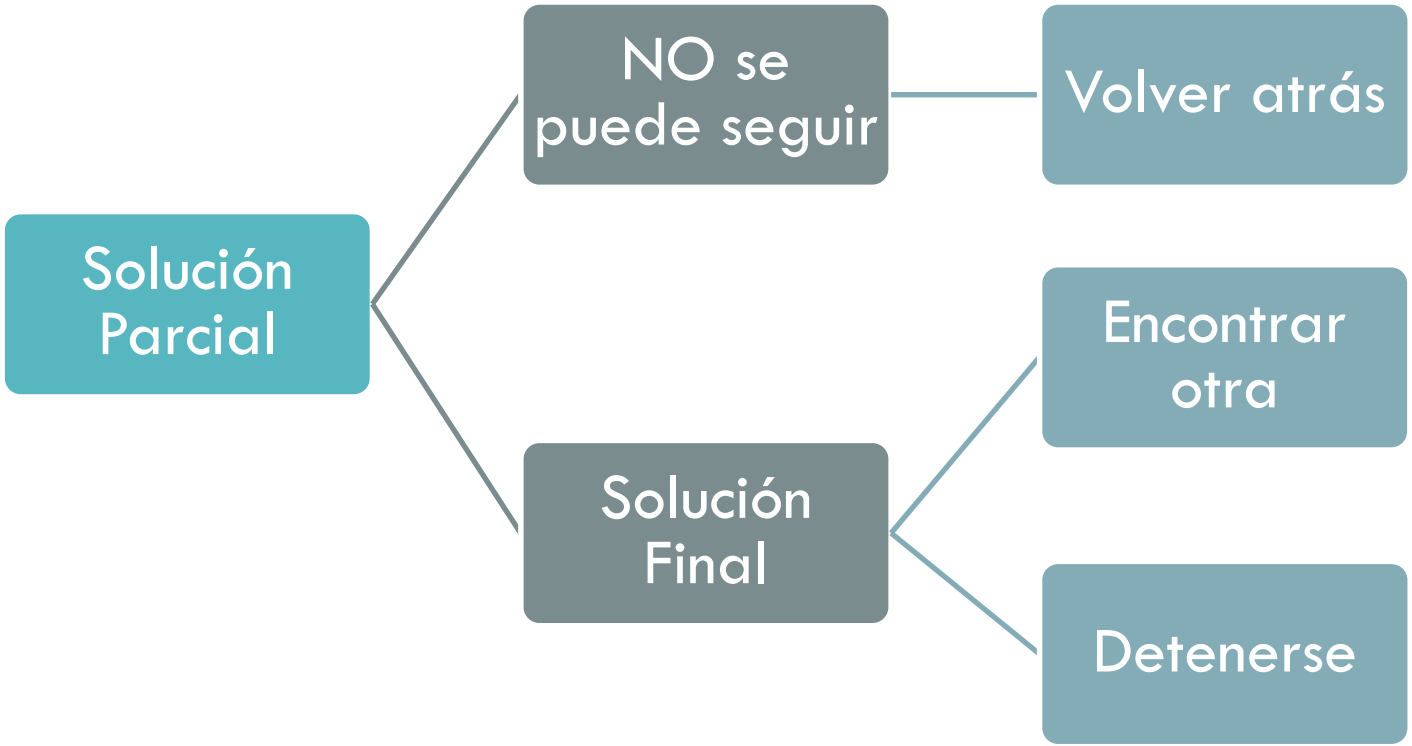
    $\big(\text{domestico}(\text{A}) \land amo(B, A)\big) \rightarrow mascota(A, B)$

Resolución SLD

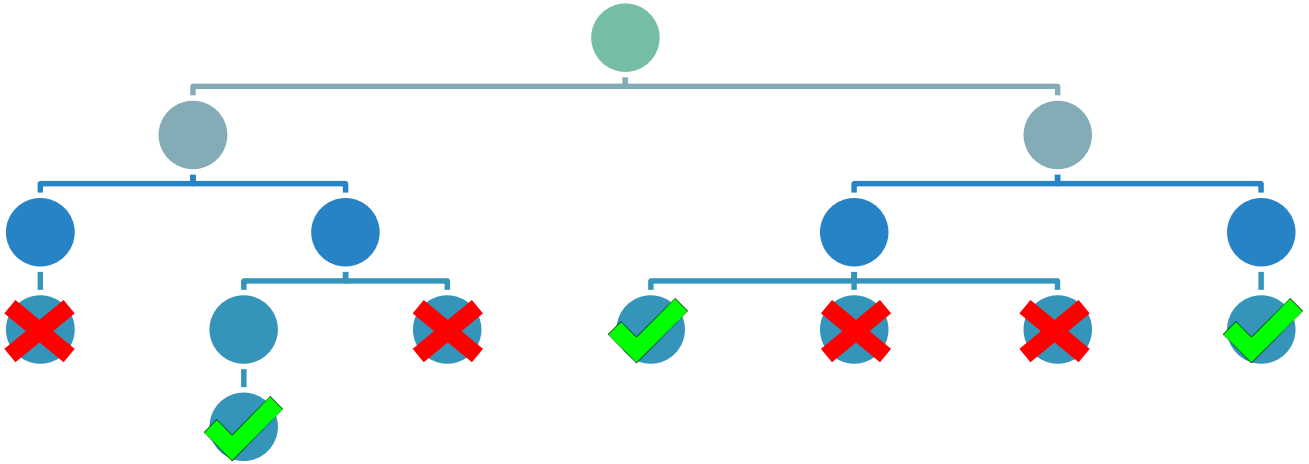


Backtracking

Backtracking



Backtracking



Ventajas

- Relaciones multipropósito
- No depende de la implementación
- Programas sencillos, potentes y elegantes
- Generación rápida de prototipos e ideas complejas

Desventajas

- Altamente ineficiente
- Pocas áreas de aplicación
- Poca investigación en tecnologías complementarias

Lenguajes

Prolog

- proveniente del francés PROgrammation en LOGique
- Producción interpretada
- Usado para inteligencia artificial
- Se basa en lógica de primer orden
- Es declarativo

pyDatalog

- Convierte Python en lenguaje lógico
- Seguro para hilos
- Acceso a otras librerías de Python

Lenguajes

ALF

- Algebraico – lógico - funcional
- Multi-paradigma
- Cláusulas de Horn.
- Evaluación de expresiones y funciones matemáticas

Mercury

- Basado en Prolog
- Lenguaje de alto nivel
- Puramente lógico
- Declarativo

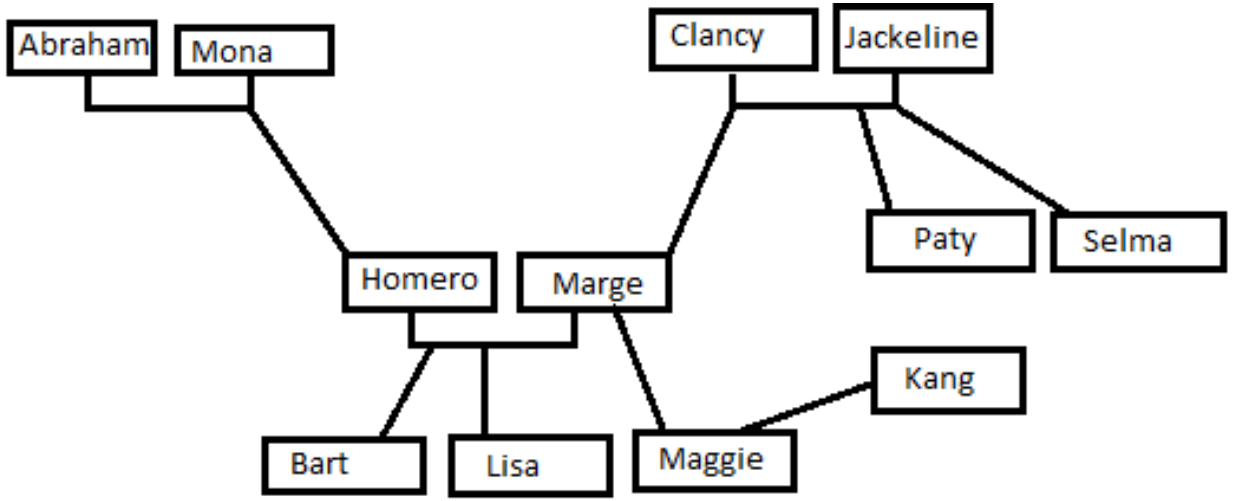
Lenguajes

Otros lenguajes derivados de Prolog

- Ace
- PALS
- Actor Prolog
- CLP(FD)
- CSP (Constraint Satisfaction Problem)
- Lambda Prolog
- Logtalk

Ejemplos

Consideremos la siguiente imagen, representa una relación básica de familia entre los personajes de los Simpsons.



Ejemplos

Representemos las características de los personajes, y su relación, además de hacer las reglas para tíos, abuelos, y hermanos.

- Características :
  - Las personas son hombres o mujeres.

```
1   hombre(homero).
2   hombre(bart).
3   hombre(abraham).
4   hombre(clancy).
5   hombre(kang).
6   /*.......................*/
7   mujer(jacqueline).
8   mujer(mona).
9   mujer(marge).
10  mujer(paty).
11  mujer(selma).
12  mujer(maggie).
13  mujer(lisa).
```

Ejemplos

- Relación :
  - Las personas son hijos de otras personas.

```
14  /*.........................*/
15  progenitor(homero,bart).
16  progenitor(marge,bart).
17  /*.........................*/
18  progenitor(homero,lisa).
19  progenitor(marge,lisa).
20  /*.........................*/
21  progenitor(kang,maggie).
22  progenitor(marge,maggie).
23  /*.........................*/
24  progenitor(abraham,homero).
25  progenitor(mona,homero).
26  /*.........................*/
27  progenitor(clancy,marge).
28  progenitor(jacqueline,marge).
29  /*.........................*/
30  progenitor(clancy,selma).
31  progenitor(jacqueline,selma).
32  /*.........................*/
33  progenitor(clancy,paty).
34  progenitor(jacqueline,paty).
```

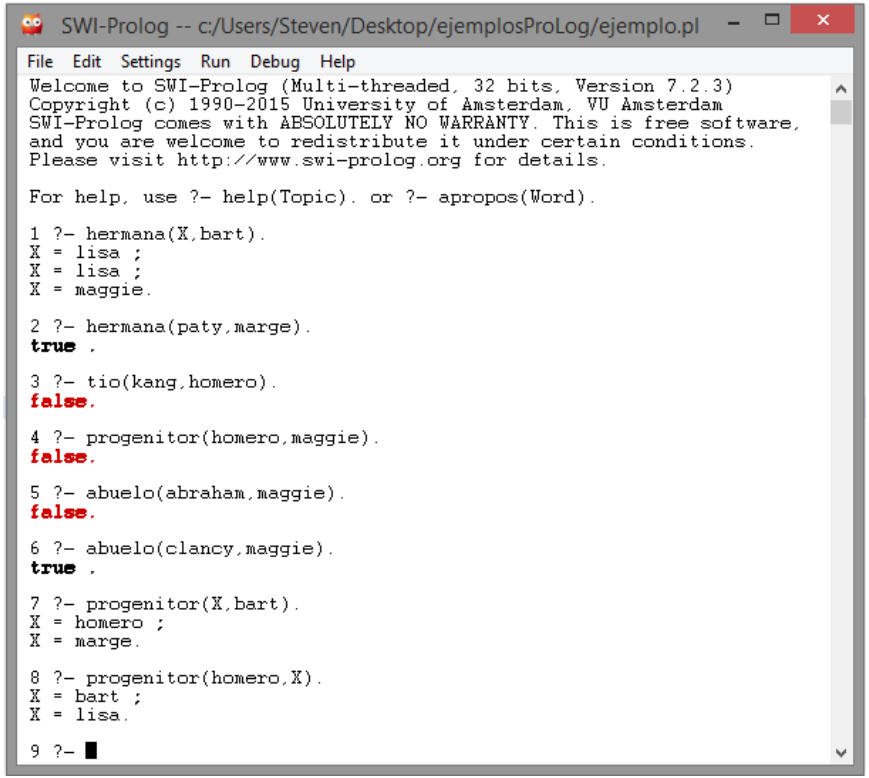
Ejemplos

- Reglas :
  - Ahora las reglas de abuelos, tíos y hermanos.

```
35  /*.........................*/
36  abuelo(X,Y):- progenitor(M,Y),progenitor(X,M),hombre(X).
37  abuela(X,Y):- progenitor(M,Y),progenitor(X,M),mujer(X).
38  /*.........................*/
39  hermano(X,Y):- progenitor(M,Y),progenitor(M,X),hombre(X).
40  hermana(X,Y):- progenitor(M,Y),progenitor(M,X),mujer(X).
41  /*.........................*/
42  tio(X,Y):- progenitor(M,Y),hermano(M,X).
43  tia(X,Y):- progenitor(M,Y),hermana(M,X).
```

Ejemplos

- Comprobemos algunas consultas.



Ejemplos

- Fibonacci en Mercury.

```
1   :- module fib.
2   :- interface.
3   :- import_module io.
4   :- pred main(io::di, io::uo) is det.
5
6   :- implementation.
7   :- import_module int.
8
9   :- func fib(int) = int.
10  fib(N) = (if N =< 2 then 1 else fib(N - 1) + fib(N - 2)).
11
12  main(!IO) :-
13          io.write_string("fib(10) = ", !IO),
14          io.write_int(fib(10), !IO),
15          io.nl(!IO).
```

Ejemplos

- Fibonacci en Prolog.

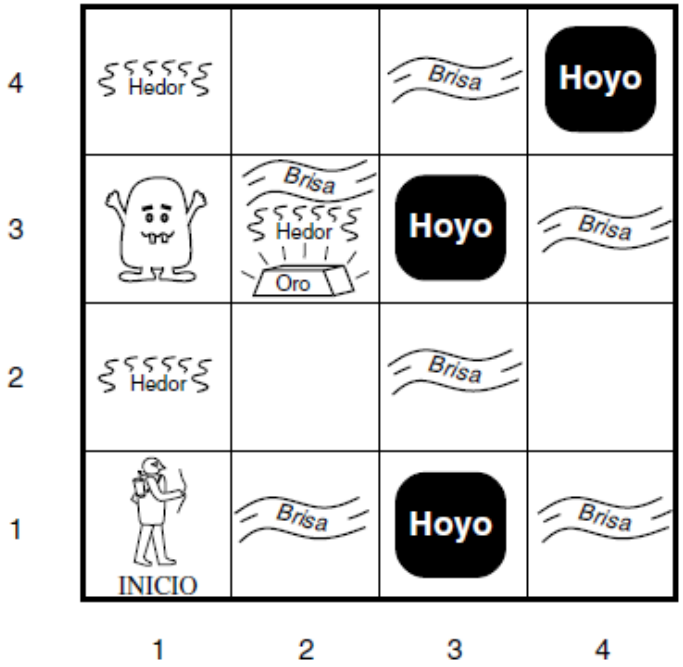
```
1  fibo(0,0).
2  fibo(1,1).
3  fibo(N,F):- N1 is N-1, N2 is N1-1,
4              fibo(N1,A),
5              fibo(N2,B),
6              F is A+B.
```

```
1 ?- fibo(0,F).
F = 0 .

2 ?- fibo(8,F).
F = 21 .

3 ?- fibo(10,F).
F = 55
```

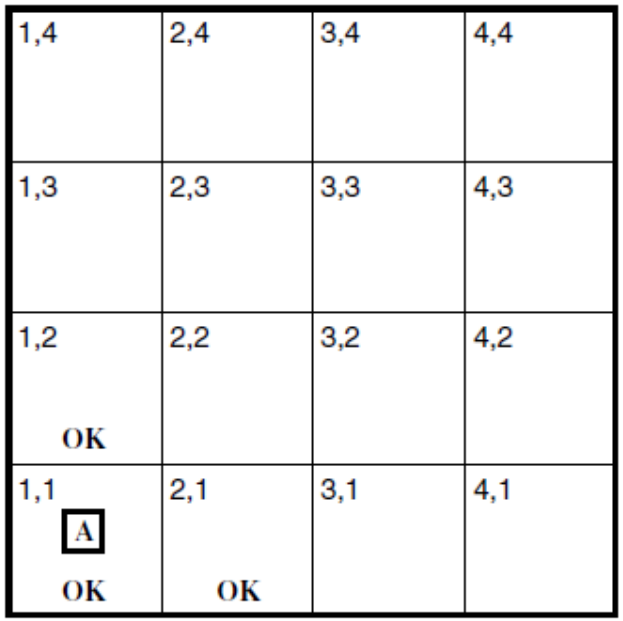
Áreas de aplicación

- Inferencia de tipos
- Procesamiento lenguaje natural
- Comprobación de teoremas
- Computación simbólica
- Parsing
- Inteligencia Artificial
- Búsqueda de patrones

El mundo del Wumpus

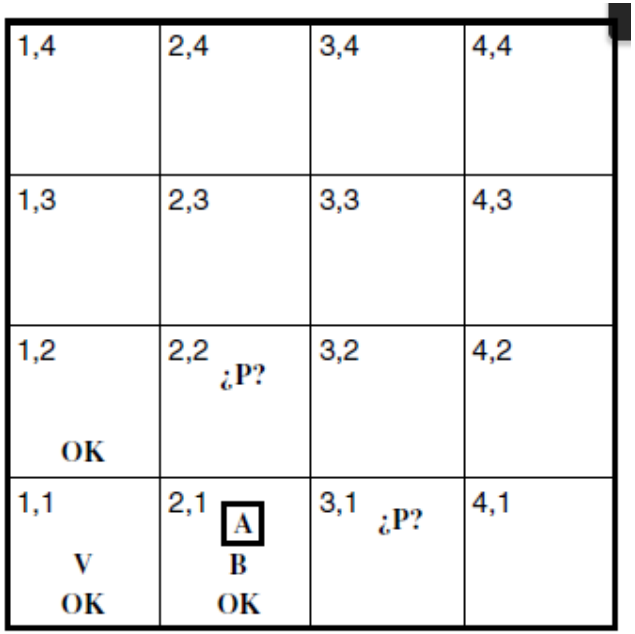


El mundo del Wumpus



(a)

A = *Agente*
B = *Brisa*
G = *Resplandor, Oro*
OK = *Casilla segura*
P = *Hoyo*
S = *Mal hedor*
V = *Visitada*
W = *Wumpus*



(b)

El mundo del Wumpus

¿ Cual es la base del conocimiento ?

Para los Hoyos:

- *Hi,j* es verdadero si hay un hoyo en la casilla [i, j].
- *Bi,j* es verdadero si hay una corriente de aire (una brisa) en la casilla [i, j].

El mundo del Wumpus

Y se sabe:

- $R1$: $\neg H1,1$
- $R2$: $B1,1 \Leftrightarrow (H1,2 \vee H2,1)$
- $R3$: $B2,1 \Leftrightarrow (H1,1 \vee H2,2 \vee H3,1)$
- $R4$: $\neg B1,1$
- $R5$: $B2,1$

El mundo del Wumpus

$$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha) \quad \text{Conmutatividad de } \wedge$$

$$(\alpha \vee \beta) \equiv (\beta \vee \alpha) \quad \text{Conmutatividad de } \vee$$

$$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma)) \quad \text{Asociatividad de } \wedge$$

$$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma)) \quad \text{Asociatividad de } \vee$$

$$\neg(\neg\alpha) \equiv \alpha \quad \text{Eliminación de la doble negación}$$

$$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{Contraposición}$$

$$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta) \quad \text{Eliminación de la implicación}$$

$$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) \quad \text{Eliminación de la bicondicional}$$

$$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta) \quad \text{Ley de Morgan}$$

$$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta) \quad \text{Ley de Morgan}$$

$$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) \quad \text{Distribución de } \wedge \text{ respecto a } \vee$$

$$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) \quad \text{Distribución de } \vee \text{ respecto a } \wedge$$

El mundo del Wumpus

$$\frac{\alpha \Rightarrow \beta, \quad \alpha}{\beta}$$ Modus Ponens

$$\frac{\alpha \wedge \beta}{\alpha}$$ Eliminación ^

$$\frac{\alpha \Leftrightarrow \beta}{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)} \quad \text{y} \quad \frac{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}{\alpha \Leftrightarrow \beta}$$ Eliminación Bi-condicional

El mundo del Wumpus

Inferencia:

- R6: ($B1,1 \Rightarrow (H1,2 \vee H2,1)) \wedge ((H1,2 \vee H2,1) \Rightarrow B1,1$) (Bi-condicional)
- R7: ($(H1,2 \vee H2,1) \Rightarrow B1,1$) (Eliminación ^)
- R8: ($\neg B1,1 \Rightarrow \neg(H1,2 \vee H2,1)$) (Contraposición)
- R9: $\neg(H1,2 \vee H2,1)$ (Modus Ponens)
- R10: $\neg H1,2 \wedge \neg H2,1$

Bibliografía

- Horn:
  - https://en.wikipedia.org/wiki/Horn_clause
- SLD:
  - https://en.wikipedia.org/wiki/SLD_resolution
- Programación Lógica:
  - http://blog.koalite.com/2013/08/que-es-la-programacion-logica/
  - http://www.paradylenguajes.com.ar/rmonzon/Intro_PyL.pdf
  - http://www.ecured.cu/Programaci%C3%B3n_l%C3%B3gica
  - http://spivey.oriel.ox.ac.uk/wiki/files/lp/logic.pdf
- Prolog:
  - http://www.learnprolognow.org/lpnpage.php?pagetype=html&pageid=lpn-htmlse1
- Mercury:
  - http://www.mercurylang.org/information/doc-release/reference_manual.pdf
- Inteligencia artificial:
  - Artificial Intelligence: A Modern Approach, (Third edition) by Stuart Russell and Peter Norvig

Gracias.