

# Tema 1: El sistema deductivo de Prolog

José A. Alonso Jiménez

Jose-Antonio.Alonso@cs.us.es

<http://www.cs.us.es/~jalonso>

Dpto. de Ciencias de la Computación e Inteligencia Artificial

UNIVERSIDAD DE SEVILLA

# Objetivos del curso

- **Lógica como:**
  - sistema de especificación y
  - lenguaje de programación
- **Principios:**
  - Programas = Teorías
  - Ejecución = Búsqueda de pruebas
  - Programación = Modelización
- **Prolog = Programming in Logic**
- **Relaciones con otros campos:**
  - Inteligencia artificial
  - Sistemas basados en el conocimiento
  - Procesamiento del lenguaje natural

# Declarativo vs. imperativo

- Paradigmas
  - Imperativo: Se describe *cómo* resolver el problema
  - Declarativo: Se describe *qué* es el problema
- Programas
  - Imperativo: Una sucesión de instrucciones
  - Declarativo: Un conjunto de sentencias
- Lenguajes
  - Imperativo: Pascal, C, Fortran
  - Declarativo: Prolog, Lisp puro, ML, Haskell
- Ventajas
  - Imperativo: Programas rápidos y especializados
  - Declarativo: Programas generales, cortos y legibles

# Historia de la programación lógica

- 1960: Demostración automática de teoremas
- 1965: Resolución y unificación (Robinson)
- 1969: QA3, obtención de respuesta (Green)
- 1972: Implementación de Prolog (Colmerauer)
- 1974: Programación lógica (Kowalski)
- 1977: Prolog de Edimburgo (Warren)
- 1981: Proyecto japonés de Quinta Generación
- 1986: Programación lógica con restricciones
- 1995: Estándar ISO de Prolog

# Deducción Prolog en lógica proposicional

- Base de conocimiento y objetivo:
  - Base de conocimiento:
    - Regla 1: Si un animal es ungulado y tiene rayas negras, entonces es una cebra.
    - Regla 2: Si un animal rumia y es mamífero, entonces es ungulado.
    - Regla 3: Si un animal es mamífero y tiene pezuñas, entonces es ungulado.
    - Hecho 1: El animal es mamífero.
    - Hecho 2: El animal tiene pezuñas.
    - Hecho 3: El animal tiene rayas negras.
  - Objetivo: Demostrar a partir de la base de conocimientos que el animal es una cebra.

# Deducción Prolog en lógica proposicional

- Programa

```
es_cebra      :- es_ungulado, tiene_rayas_negras. % R1
es_ungulado  :- rumia, es_mamífero.              % R2
es_ungulado  :- es_mamífero, tiene_pezuñas.      % R3
es_mamífero.                                     % H1
tiene_pezuñas.                                   % H2
tiene_rayas_negras.                              % H3
```

- Sesión

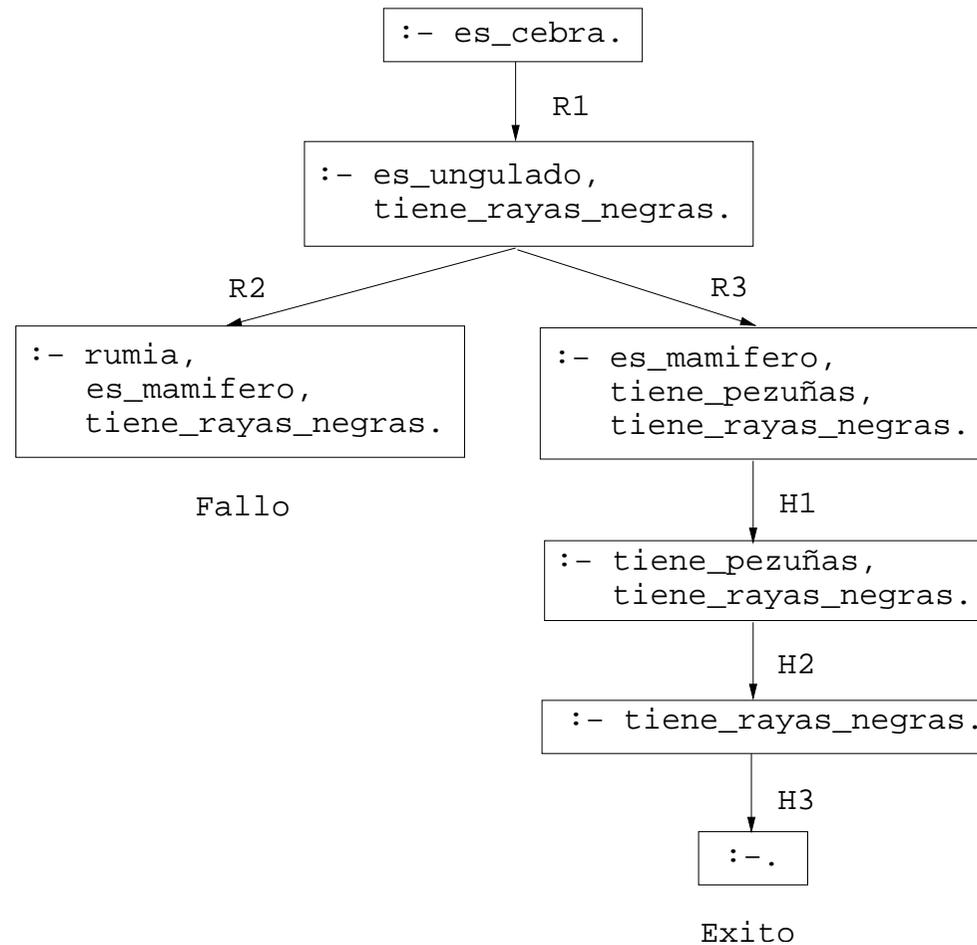
```
> pl
Welcome to SWI-Prolog (Version 5.0.3)
Copyright (c) 1990-2002 University of Amsterdam.
```

```
?- [animales].
Yes
```

```
?- es_cebra.
Yes
```

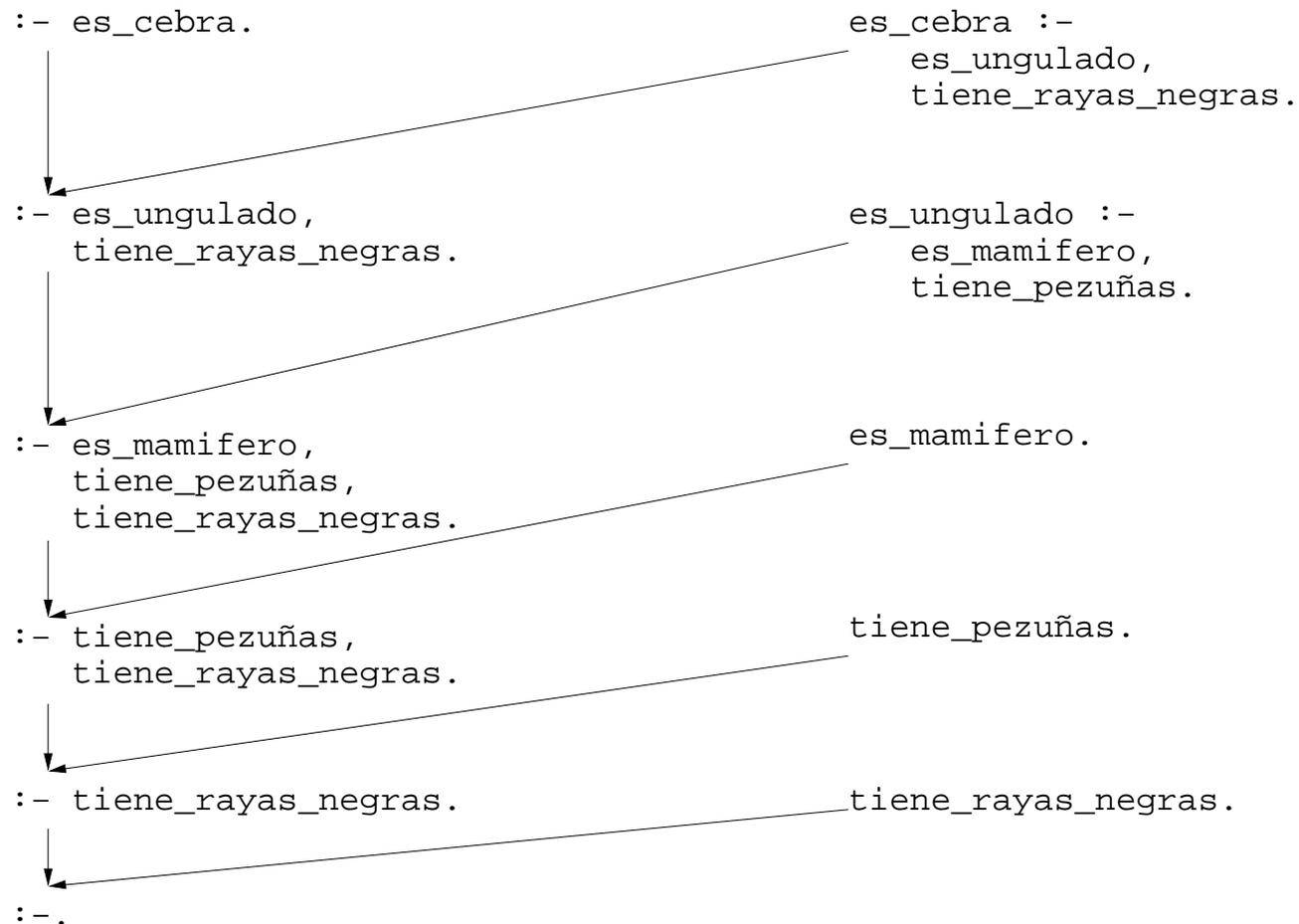
# Deducción Prolog en lógica proposicional

- Árbol de deducción



# Deducción Prolog en lógica proposicional

- Demostración por resolución SLD



# Deducción Prolog en lógica relacional

- Base de conocimiento:

- Hechos 1-4: 6 y 12 son divisibles por 2 y por 3.
- Hecho 5: 4 es divisible por 2.
- Regla 1: Los números divisibles por 2 y por 3 son divisibles por 6.

- Programa

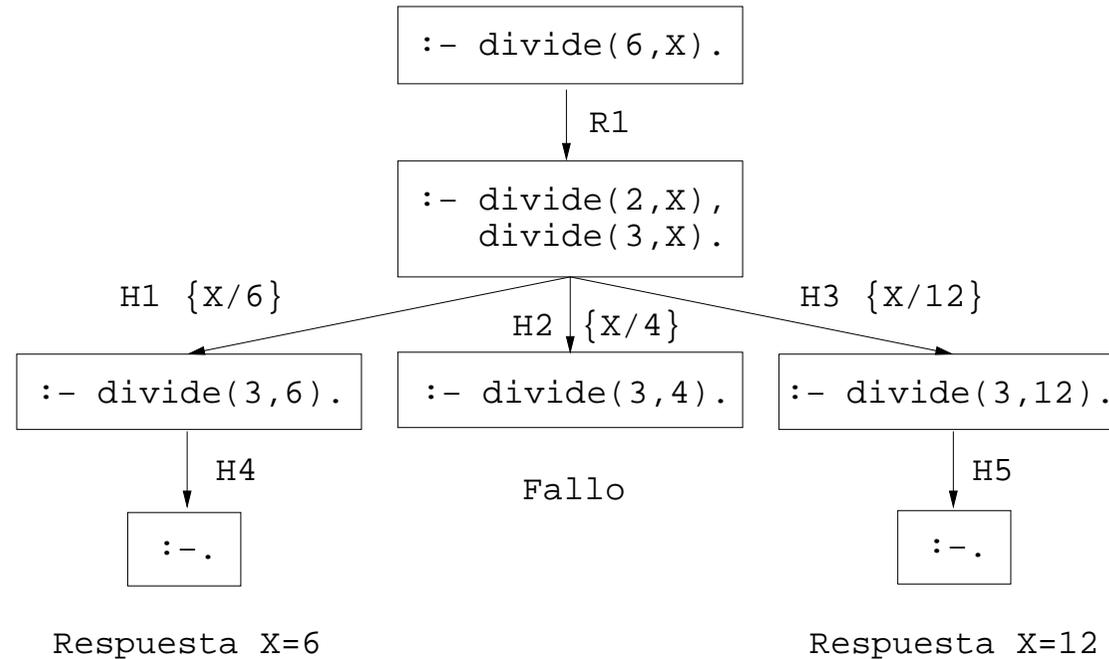
```
divide(2,6).           % Hecho 1
divide(2,4).           % Hecho 2
divide(2,12).          % Hecho 3
divide(3,6).           % Hecho 4
divide(3,12).          % Hecho 5
divide(6,X) :- divide(2,X), divide(3,X). % Regla 1
```

# Deducción Prolog en lógica relacional

- **Símbolos:**
  - Constantes: 2, 3, 4, 6, 12
  - Relación binaria: divide
  - Variable: X
- **Interpretaciones de la Regla 1:**
  - $divide(6,X) :- divide(2,X), divide(3,X).$
  - Interpretación declarativa:  
 $(\forall X)[divide(2, X) \wedge divide(3, X) \rightarrow divide(6, X)]$
  - Interpretación procedimental
- **Consulta:**
  - ¿Cuáles son los múltiplos de 6?  
 $?- divide(6,X).$   
X = 6 ;  
X = 12 ;  
No

# Deducción Prolog en lógica relacional

- **Árbol de deducción**



- **Comentarios:**

- Unificación
- Cálculo de respuestas
- Respuestas múltiples

# Deducción Prolog en lógica funcional

- Representación de los números naturales:

- $0, s(0), s(s(0)), \dots$

- Definición de la suma

$0 + Y = Y$

$s(X) + Y = s(X+Y)$

- Programa

```
suma(0,Y,Y).                % R1
```

```
suma(s(X),Y,s(Z)) :- suma(X,Y,Z). % R2
```

- Consulta:

- Pregunta: ¿Cuál es la suma de  $s(0)$  y  $s(s(0))$ ?

- Sesión:

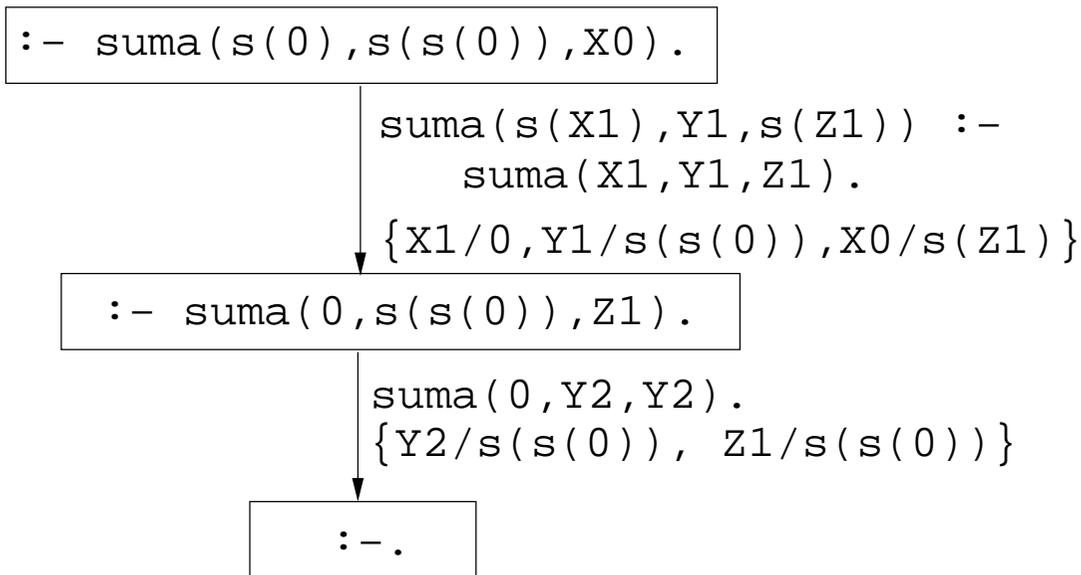
```
?- suma(s(0),s(s(0)),X).
```

```
X = s(s(s(0)))
```

```
Yes
```

# Deducción Prolog en lógica funcional

- Árbol de deducción:



Resp.:  $X = X0 = s(Z1) = s(s(s(0)))$

# Deducción Prolog en lógica funcional

- **Consulta:**

- ¿Cuál es la resta de  $s(s(s(0)))$  y  $s(0)$ ?

- **Sesión:**

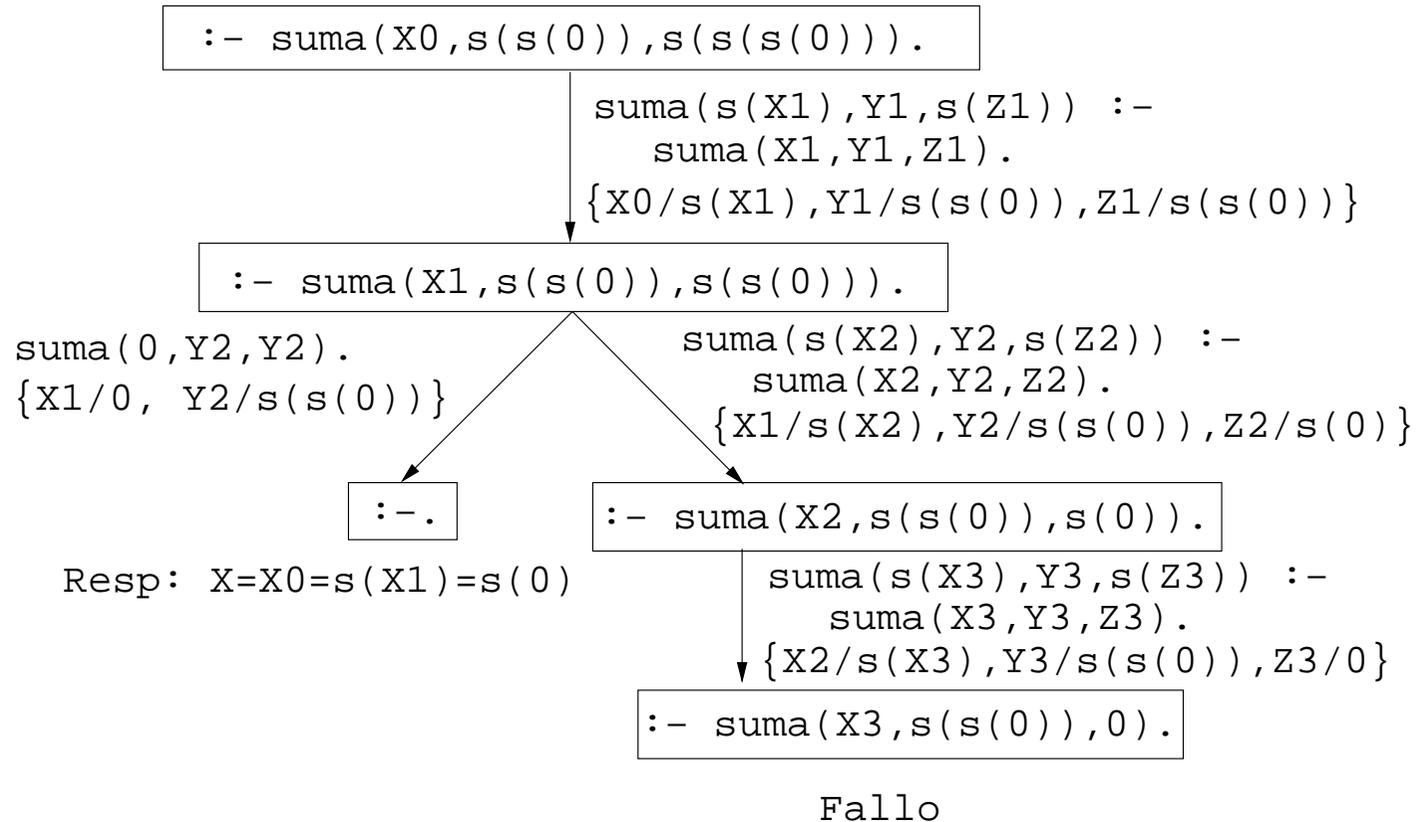
```
?- suma(X,s(s(0)),s(s(s(0)))).
```

```
X = s(0) ;
```

```
No
```

# Deducción Prolog en lógica funcional

- Árbol de deducción:



# Deducción Prolog en lógica funcional

- **Consulta:**

- **Pregunta:** ¿Cuáles son las soluciones de la ecuación

$$X + Y = s(s(0))?$$

- **Sesión:**

```
?- suma(X,Y,s(s(0))).
```

```
X = 0           Y = s(s(0)) ;
```

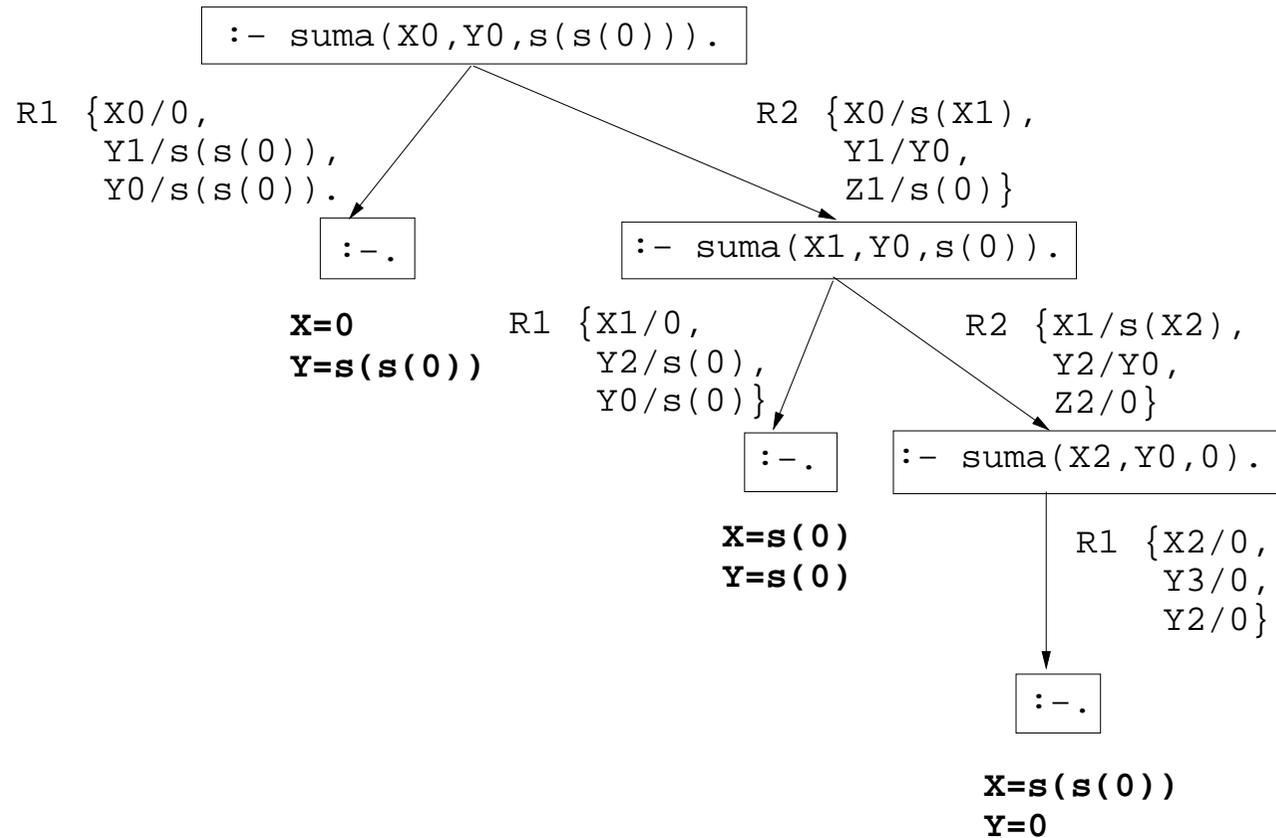
```
X = s(0)       Y = s(0) ;
```

```
X = s(s(0))   Y = 0 ;
```

```
No
```

# Deducción Prolog en lógica funcional

- Árbol de deducción:



# Deducción Prolog en lógica funcional

- Consulta:

- Pregunta: resolver el sistema de ecuaciones

$$1 + X = Y$$

$$X + Y = 1$$

- Sesión:

?- suma(s(0), X, Y), suma(X, Y, s(0)).

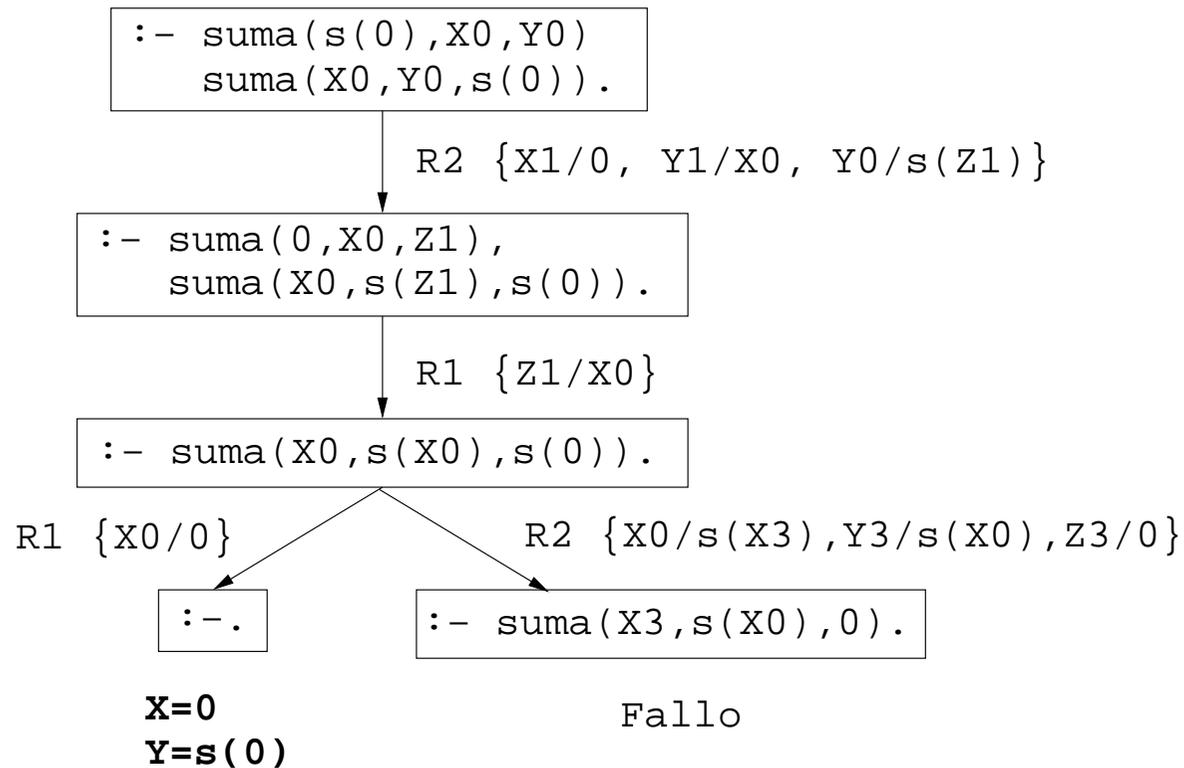
X = 0

Y = s(0) ;

No

# Deducción Prolog en lógica funcional

- Árbol de deducción



# Bibliografía

- Alonso, J.A. y Borrego, J. *Deducción automática (Vol. 1: Construcción lógica de sistemas lógicos)* (Ed. Kronos, 2002)  
[www.cs.us.es/~jalonso/libros/da1-02.pdf](http://www.cs.us.es/~jalonso/libros/da1-02.pdf)
  - Cap. 2: Introducción a la programación lógica con Prolog.
- Bratko, I. *Prolog Programming for Artificial Intelligence (2nd ed.)* (Addison–Wesley, 1990)
  - Cap. 1: “An overview of Prolog”
  - Cap. 2: “Syntax and meaning of Prolog programs”
- Clocksin, W.F. y Mellish, C.S. *Programming in Prolog (Fourth Edition)* (Springer Verlag, 1994)
  - Cap. 1: “Tutorial introduction”
  - Cap. 2: “A closer look”