

Tema 6: Aplicaciones

Dpto. Ciencias de la Computación Inteligencia Artificial
UNIVERSIDAD DE SEVILLA

Lógica Informática (Tecnologías Informáticas)

Software

Prover9 y Mace4
ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos
Problema de los rectángulos
Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku
Problema de las N Reinas
Problema del vecindario

Contenido

Software

Prover9 y Mace4

ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos

Problema de los rectángulos

Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku

Problema de las N Reinas

Problema del vecindario

Software

Prover9 y Mace4

ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos

Problema de los rectángulos

Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku

Problema de las N Reinas

Problema del vecindario

Prover9 y Mace4

- ▶ **Prover9** es un demostrador automático para las lógicas proposicionales y de primer orden.
- ▶ **Mace4** es un calculador de modelos.
- ▶ Disponen de un interfaz gráfico. Son libres y se pueden descargar en:

<http://www.cs.unm.edu/~mccune/mace4>

The screenshot displays the graphical user interface for Prover9 and Mace4. The interface is divided into two main panels, one for Prover9 and one for Mace4.

Prover9 Panel:

- Assumptions:**

```

base_a | altura_a. base_b | altura_b. base_c | altura_c.
base_d | altura_d. base_e | altura_e. base_f | altura_f.
base_a <-> base_c.
base_a & base_d -> base_f.
base_d & base_e -> base_b.
base_a & base_b -> base_e.
altura_d & altura_f -> altura_e.
altura_a & altura_c & altura_f -> altura.
altura_b & altura_d & altura_f -> altura.
altura_b & altura_e -> altura.

```
- Goals:**

```

base | altura.

```
- Buttons:** Highlight, Well Formed?, Clear.

Mace4 Panel:

- Model/Counterexample Search:**
 - Time Limit: 60 seconds.
 - Buttons: Start, Kill.
 - State: Model(s)
 - Buttons: Info, Show/Save.

Software

Prover9 y Mace4

ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos

Problema de los rectángulos

Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku

Problema de las N Reinas

Problema del vecindario

- ▶ El objetivo de ToulST es el de proporcionar una herramienta amigable para trabajar sobre el problema de la satisfactibilidad en lenguajes proposicionales.
- ▶ Está disponible en francés e inglés en:
<https://www.irit.fr/touist/>
- ▶ Esta basado en una herramientas anterior llamada SAToulouse, realizada en JAVA.
- ▶ Trabaja sobre ficheros de texto, bien con un interfaz propio (a veces da problemas), o con un editor cualquiera y después ejecutarlo por línea de comandos.



Software

Prover9 y Mace4

ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos

Problema de los rectángulos

Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku

Problema de las N Reinas

Problema del vecindario

- ▶ Trabaja sobre Lógica Proposicional, pero puede definir fórmulas con parámetros.
- ▶ Por ejemplo, una fórmula como:

$$\bigwedge_{i=1}^9 \bigvee_{j=1}^9 \bigwedge_{n=1}^9 \bigwedge_{m=1, m \neq n}^9 (p_{i,j,n} \rightarrow \neg p_{i,j,m})$$

Se escribiría en ToulST como:

```
bigand $i in [1..9]:  
  bigor $j in [1..9]:  
    bigand $n,$m in [1..9],[1..9] when $m != $n:  
      p($i,$j,$n) => not p($i,$j,$m)  
    end  
  end  
end
```

Software

Prover9 y Mace4

ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos

Problema de los rectángulos

Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku

Problema de las N Reinas

Problema del vecindario

Problema de los mentirosos

Enunciado: En una isla hay dos tribus, la de los veraces (que siempre dicen la verdad) y la de los mentirosos (que siempre mienten). Un viajero se encuentra con tres isleños A, B y C y cada uno le da la siguiente información:

- ▶ A dice "B y C son veraces si y sólo si C es veraz"
- ▶ B dice "Si A y C son veraces, entonces B y C son veraces y A es mentiroso"
- ▶ C dice "B es mentiroso si y sólo si A o B es veraz"

Determinar a qué tribu pertenece cada uno.

Software

Prover9 y Mace4
ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos
Problema de los rectángulos
Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku
Problema de las N Reinas
Problema del vecindario

Problema de los mentirosos

- ▶ **Formalización:** a , b y c representan que A, B y C son veraces (en consecuencia, $\neg a$, $\neg b$ y $\neg c$ representan que A, B y C son mentirosos).

- ▶ **Representación:**

$$a \leftrightarrow (b \wedge c \leftrightarrow c)$$

$$b \leftrightarrow (a \wedge c \rightarrow b \wedge c \wedge \neg a)$$

$$c \leftrightarrow (\neg b \leftrightarrow a \vee b)$$

- ▶ Resultado del cálculo de modelos con Mace4:

$$v(a) = 1, v(b) = 1, v(c) = 0$$

- ▶ **Conclusión:** A y B son veraces y C es mentiroso.

Software

Prover9 y Mace4
ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos
Problema de los rectángulos
Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku
Problema de las N Reinas
Problema del vecindario

Problema de los mentirosos

Si añadimos $a \wedge b \wedge \neg c$ como objetivo, y usamos Prover9 para encontrar una demostración:

```

Save as... Close
===== prooftrans =====
Prover9 (32) version Dec-2007, Dec 2007.
Process 3764 was started by Fernando on Agente,
Sun Nov 3 14:03:47 2013
The command was "/cygdrive/c/Program Files (x86)/Prover9-Mace4/bin-win32/prover9".
===== end of head =====

===== end of input =====

===== PROOF =====

% ----- Comments from original proof -----
% Proof 1 at 0.00 (+ 0.03) seconds.
% Length of proof is 16.
% Level of proof is 6.
% Maximum clause weight is 3.
% Given clauses 8.

1 a <-> (b & c <-> c) # label(non_clause). [].
2 b <-> (a & c -> b & c & ~a) # label(non_clause). [].
3 c <-> (~b <-> a | b) # label(non_clause). [].
4 a & b & ~c # label(non_clause) # label(goal). [].
5 ~a | b | ~c. [1].
6 a | c. [1].
9 b | a. [2].
10 b | c. [2].
11 ~c | ~b. [3].
12 ~a | ~b | c. [4].
13 b | ~a. [10,5].
14 ~b | a. [11,6].
15 a. [14,9].
16 b. [13,15].
17 c. [12,15,16].
18 $F. [11,17,16].

===== end of proof =====

```

Software

Prover9 y Mace4
TouIST

Ejemplos Lógica Proposicional

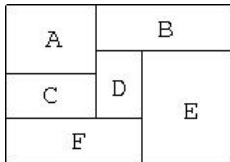
Problema de los mentirosos
Problema de los rectángulos
Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku
Problema de las N Reinas
Problema del vecindario

Problema de los rectángulos

Enunciado: Un rectángulo se divide en seis rectángulos menores como se indica en la figura. Demostrar que si cada una de los rectángulos menores tiene un lado cuya medida es un número entero, entonces la medida de alguno de los lados del rectángulo mayor es un número entero.



Formalización:

- ▶ *base*: la base del rectángulo mayor es un número entero
- ▶ *altura*: la altura del rectángulo mayor es un número entero
- ▶ $base_x$: la base del rectángulo x es un número entero
- ▶ $altura_x$: la altura del rectángulo x es un número entero

Software

Prover9 y Mace4
ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos
Problema de los rectángulos
Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku
Problema de las N Reinas
Problema del vecindario

Problema de los rectángulos

► Representación:

$base_A \vee altura_A, base_B \vee altura_B, base_C \vee altura_C$

$base_D \vee altura_D, base_E \vee altura_E, base_F \vee altura_F$

$base_A \leftrightarrow base_C, base_A \wedge base_D \rightarrow base_F$

$base_D \wedge base_E \rightarrow base_B, base_A \wedge base_B \rightarrow base$

$altura_D \wedge altura_F \rightarrow altura_E$

$altura_A \wedge altura_C \wedge altura_F \rightarrow altura$

$altura_B \wedge altura_D \wedge altura_F \rightarrow altura$

$altura_B \wedge altura_E \rightarrow altura$

► Objetivo: $base \vee altura$

► Prover9: THEOREM PROVED

Software

Prover9 y Mace4

ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos

Problema de los rectángulos

Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku

Problema de las N Reinas

Problema del vecindario

Problema de las 4 reinas

Enunciado: Calcular las formas de colocar 4 reinas en un tablero de 4x4 de forma que no se ataquen entre sí (no haya más de una reina en cada fila, columna o diagonal).

		R	
R			
			R
	R		

	R		
			R
R			
		R	

Formalización: r_{ij} ($1 \leq i, j \leq 4$) indica que hay una reina en la fila i columna j

Software

Prover9 y Mace4
ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos
Problema de los rectángulos
Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku
Problema de las N Reinas
Problema del vecindario

Problema de las 4 reinas

En cada fila hay una reina:

$$r_{11} \vee r_{12} \vee r_{13} \vee r_{14}, \quad r_{21} \vee r_{22} \vee r_{23} \vee r_{24}$$

$$r_{31} \vee r_{32} \vee r_{33} \vee r_{34}, \quad r_{41} \vee r_{42} \vee r_{43} \vee r_{44}$$

Si en una casilla hay reina, entonces no hay más reinas en su fila, su columna y sus diagonales:

$$r_{11} \rightarrow (\neg r_{12} \wedge \neg r_{13} \wedge \neg r_{14}) \wedge (\neg r_{21} \wedge \neg r_{31} \wedge \neg r_{41}) \wedge (\neg r_{22} \wedge \neg r_{33} \wedge \neg r_{44})$$

$$r_{12} \rightarrow (\neg r_{11} \wedge \neg r_{13} \wedge \neg r_{14}) \wedge (\neg r_{22} \wedge \neg r_{32} \wedge \neg r_{42}) \wedge (\neg r_{21} \wedge \neg r_{23} \wedge \neg r_{34})$$

$$r_{13} \rightarrow (\neg r_{11} \wedge \neg r_{12} \wedge \neg r_{14}) \wedge (\neg r_{23} \wedge \neg r_{33} \wedge \neg r_{43}) \wedge (\neg r_{31} \wedge \neg r_{22} \wedge \neg r_{24})$$

$$r_{14} \rightarrow (\neg r_{11} \wedge \neg r_{12} \wedge \neg r_{13}) \wedge (\neg r_{24} \wedge \neg r_{34} \wedge \neg r_{44}) \wedge (\neg r_{23} \wedge \neg r_{32} \wedge \neg r_{41})$$

$$r_{21} \rightarrow (\neg r_{22} \wedge \neg r_{23} \wedge \neg r_{24}) \wedge (\neg r_{11} \wedge \neg r_{31} \wedge \neg r_{41}) \wedge (\neg r_{32} \wedge \neg r_{43} \wedge \neg r_{12})$$

$$r_{22} \rightarrow (\neg r_{21} \wedge \neg r_{23} \wedge \neg r_{24}) \wedge (\neg r_{12} \wedge \neg r_{32} \wedge \neg r_{42}) \wedge (\neg r_{11} \wedge \neg r_{33} \wedge \neg r_{44} \wedge \neg r_{13} \wedge \neg r_{31})$$

$$r_{23} \rightarrow (\neg r_{21} \wedge \neg r_{22} \wedge \neg r_{24}) \wedge (\neg r_{13} \wedge \neg r_{33} \wedge \neg r_{43}) \wedge (\neg r_{12} \wedge \neg r_{34} \wedge \neg r_{14} \wedge \neg r_{32} \wedge \neg r_{41})$$

$$r_{24} \rightarrow (\neg r_{21} \wedge \neg r_{22} \wedge \neg r_{23}) \wedge (\neg r_{14} \wedge \neg r_{34} \wedge \neg r_{44}) \wedge (\neg r_{13} \wedge \neg r_{33} \wedge \neg r_{42})$$

Software

Prover9 y Mace4
ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos
Problema de los rectángulos
Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku
Problema de las N Reinas
Problema del vecindario

Problema de las 4 reinas

Si en una casilla hay reina, entonces no hay más reinas en su fila, su columna y sus diagonales:

$$r_{31} \rightarrow (\neg r_{32} \wedge \neg r_{33} \wedge \neg r_{34}) \wedge (\neg r_{11} \wedge \neg r_{21} \wedge \neg r_{41}) \wedge (\neg r_{42} \wedge \neg r_{13} \wedge \neg r_{22})$$

$$r_{32} \rightarrow (\neg r_{31} \wedge \neg r_{33} \wedge \neg r_{34}) \wedge (\neg r_{12} \wedge \neg r_{22} \wedge \neg r_{42}) \wedge (\neg r_{21} \wedge \neg r_{43} \wedge \neg r_{14} \wedge \neg r_{23} \wedge \neg r_{41})$$

$$r_{33} \rightarrow (\neg r_{31} \wedge \neg r_{32} \wedge \neg r_{34}) \wedge (\neg r_{13} \wedge \neg r_{23} \wedge \neg r_{43}) \wedge (\neg r_{11} \wedge \neg r_{22} \wedge \neg r_{44} \wedge \neg r_{24} \wedge \neg r_{42})$$

$$r_{34} \rightarrow (\neg r_{31} \wedge \neg r_{32} \wedge \neg r_{33}) \wedge (\neg r_{14} \wedge \neg r_{24} \wedge \neg r_{44}) \wedge (\neg r_{12} \wedge \neg r_{23} \wedge \neg r_{43})$$

$$r_{41} \rightarrow (\neg r_{42} \wedge \neg r_{43} \wedge \neg r_{44}) \wedge (\neg r_{11} \wedge \neg r_{21} \wedge \neg r_{31}) \wedge (\neg r_{14} \wedge \neg r_{23} \wedge \neg r_{32})$$

$$r_{42} \rightarrow (\neg r_{41} \wedge \neg r_{43} \wedge \neg r_{44}) \wedge (\neg r_{12} \wedge \neg r_{22} \wedge \neg r_{32}) \wedge (\neg r_{31} \wedge \neg r_{24} \wedge \neg r_{33})$$

$$r_{43} \rightarrow (\neg r_{41} \wedge \neg r_{42} \wedge \neg r_{44}) \wedge (\neg r_{13} \wedge \neg r_{23} \wedge \neg r_{33}) \wedge (\neg r_{21} \wedge \neg r_{32} \wedge \neg r_{34})$$

$$r_{44} \rightarrow (\neg r_{41} \wedge \neg r_{42} \wedge \neg r_{43}) \wedge (\neg r_{14} \wedge \neg r_{24} \wedge \neg r_{34}) \wedge (\neg r_{11} \wedge \neg r_{22} \wedge \neg r_{33})$$

Software

Prover9 y Mace4

ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos

de los rectángulos

Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku

Problema de las N Reinas

Problema del vecindario

Problema de las 4 reinas

Búsqueda de modelos con Mace4:

$$r_{11} : 0, r_{12} : 0, r_{13} : 1, r_{14} : 0$$

$$r_{21} : 1, r_{22} : 0, r_{23} : 0, r_{24} : 0$$

$$r_{31} : 0, r_{32} : 0, r_{33} : 0, r_{34} : 1$$

$$r_{41} : 0, r_{42} : 1, r_{43} : 0, r_{44} : 0$$

$$r_{11} : 0, r_{12} : 1, r_{13} : 0, r_{14} : 0$$

$$r_{21} : 0, r_{22} : 0, r_{23} : 0, r_{24} : 1$$

$$r_{31} : 1, r_{32} : 0, r_{33} : 0, r_{34} : 0$$

$$r_{41} : 0, r_{42} : 0, r_{43} : 1, r_{44} : 0$$

Software

Prover9 y Mace4

ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos

Problema de los rectángulos

Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku

Problema de las N Reinas

Problema del vecindario

Problema: Resolver sudokus.

- ▶ Un sudoku está dado por un tablero de 9×9 casillas, dividido a su vez en 3×3 bloques de tamaño 3×3 .

	1	8				7		
			3			2		
	7							
				7	1			
6							4	
3								
4			5					3
	2			8				
							6	

- ▶ Cada casilla puede contener un número del 1 al 9.
- ▶ Inicialmente se proporcionan los números que ocupan algunas de las casillas. El juego consiste en rellenar el resto de casillas de modo que cada fila, cada columna y cada bloque contenga todos los números del 1 al 9.

Software

Prover9 y Mace4
ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos
Problema de los rectángulos
Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku
Problema de las N Reinas
Problema del vecindario

Solución con un LPO usando Mace4

Una solución será una función $f : [1, 9] \times [1, 9] \rightarrow [1, 9]$ de forma que $f(x, y)$ indica el valor que hay que escribir en la casilla que ocupa la posición (x, y) .

Restricciones del problema:

- ▶ El dominio de trabajo es $[1, 9]$:
$$\text{assign}(\text{domain_size}, 9).$$
(en realidad, Mace4 trabajará con $[0, 8]$)
- ▶ En cada fila todos los elementos son distintos:
$$\text{all } x \text{ all } y1 \text{ all } y2 (f(x, y1)=f(x, y2) \rightarrow y1 = y2).$$
- ▶ En cada columna todos los elementos son distintos:
$$\text{all } y \text{ all } x1 \text{ all } x2 (f(x1, y)=f(x2, y) \rightarrow x1 = x2).$$
- ▶ En cada fila hay un elemento de cada tipo:
$$\text{all } x \text{ all } z \text{ exists } y (f(x, y) = z).$$
- ▶ En cada columna hay un elemento de cada tipo:
$$\text{all } y \text{ all } z \text{ exists } x (f(x, y) = z).$$

Software

Prover9 y Mace4
ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos
Problema de los rectángulos
Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku
Problema de las N Reinas
Problema del vecindario

Solución con un LPO usando Mace4

Una solución será una función $f : [1, 9] \times [1, 9] \rightarrow [1, 9]$ de forma que $f(x, y)$ indica el valor que hay que escribir en la casilla que ocupa la posición (x, y) .

Restricciones del problema:

- ▶ En cada bloque todos los elementos son distintos:

all x1 all y1 all x2 all y2

(mismo_intervalo(x1,x2) &

mismo_intervalo(y1,y2) &

f(x1, y1)=f(x2, y2) -> x1 = x2 & y1 = y2).

donde mismo_intervalo es una **relación de equivalencia**, y además verifica (recuerda que Mace4 trabajará de 0 a 8):

mismo_intervalo(0,1) & mismo_intervalo(1,2),

mismo_intervalo(3,4) & mismo_intervalo(4,5),

mismo_intervalo(6,7) & mismo_intervalo(7,8),

-mismo_intervalo(0,3) & -mismo_intervalo(3,6) &

-mismo_intervalo(0,6)

Software

Prover9 y Mace4

ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos

Problema de los rectángulos

Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku

Problema de las N Reinas

Problema del vecindario

Solución con un LPO usando Mace4

Una solución será una función $f : [1, 9] \times [1, 9] \rightarrow [1, 9]$ de forma que $f(x, y)$ indica el valor que hay que escribir en la casilla que ocupa la posición (x, y) .

Ya solo queda añadir la información de las casillas que sí están rellenas: $f(0,0) = 1$. $f(1,1)=2$

1			2			3		
	2			3			4	
		3			4			5
6			4			5		
	7			5			6	
		8			6			7
8			0			7		
	0			1			8	
		1			2			4

Software

Prover9 y Mace4

ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos

Problema de los rectángulos

Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku

Problema de las N Reinas

Problema del vecindario

Solución con un LPO usando Mace4

Una solución será una función $f : [1, 9] \times [1, 9] \rightarrow [1, 9]$ de forma que $f(x, y)$ indica el valor que hay que escribir en la casilla que ocupa la posición (x, y) .

Y al ejecutar Mace4 devolverá una tabla con los valores de f , que son una solución al Sudoku planteado:

1	4	5	2	8	0	3	7	6
7	2	6	5	3	1	8	4	0
0	8	3	7	6	4	1	2	5
6	1	0	4	2	7	5	3	8
3	7	4	1	5	8	0	6	2
2	5	8	3	0	6	4	1	7
8	6	2	0	4	3	7	5	1
4	0	7	6	1	5	2	8	3
5	3	1	8	7	2	6	0	4

Software

Prover9 y Mace4

ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos

Problema de los rectángulos

Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku

Problema de las N Reinas

Problema del vecindario

Problema de las N Reinas

En este caso, en un LPO definimos un predicado $R(x, y)$ que indica si hay una reina en la posición (x, y) . Restricciones:

- ▶ En cada fila al menos hay una reina:

$$\text{all } x \text{ exists } y \ R(x, y).$$

- ▶ En cada fila a lo sumo hay una reina:

$$R(x, y_1) \ \& \ R(x, y_2) \ \rightarrow \ y_1 = y_2.$$

- ▶ En cada columna a lo sumo hay una reina:

$$R(x_1, y) \ \& \ R(x_2, y) \ \rightarrow \ x_1 = x_2.$$

- ▶ En cada diagonal principal a lo sumo hay una reina:

$$R(x_1, y_1) \ \& \ R(x_2, y_2) \ \& \ (x_2 + -x_1 = y_2 + -y_1) \ \rightarrow \ x_1 = x_2 \ \& \ y_1 = y_2.$$

- ▶ En cada diagonal secundaria a lo sumo hay una reina:

$$R(x_1, y_1) \ \& \ R(x_2, y_2) \ \& \ (x_1 + -x_2 = y_2 + -y_1) \ \rightarrow \ x_1 = x_2 \ \& \ y_1 = y_2.$$

Para Mace4 hemos de añadir la opción `set(arithmetric)`.
para considerar las propiedades habituales en las operaciones aritméticas, y en las opciones hemos de indicar el tamaño del mundo (el tamaño del tablero).

Software

Prover9 y Mace4
ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos
Problema de los rectángulos
Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku
Problema de las N Reinas
Problema del vecindario

Problema del vecindario

En un barrio hay 5 casas adosadas, de diferentes colores, habitadas por cinco hombres de diferentes nacionalidades.

Cada uno posee un animal diferente, bebe una bebida diferente y fuman una marca de tabaco distinta.

Sabiendo lo siguiente, ¿podrías determinar el sistema completo?

- ▶ El inglés vive en la casa roja.
- ▶ El español tiene un perro.
- ▶ El noruego vive en la primera casa.
- ▶ El de la casa amarilla fuma Ducados.
- ▶ El vecino del que fuma Chester tiene un canario.
- ▶ La casa de al lado del noruego es azul.
- ▶ El que fuma Fortuna tiene un gato.
- ▶ El que fuma Celtas bebe vino.
- ▶ El ucraniano bebe té.
- ▶ El japonés fuma Habanos.
- ▶ El vecino del que fuma Ducados tiene un caballo.
- ▶ El de la casa verde bebe café.
- ▶ La casa blanca está después de la verde.
- ▶ El de la tercera casa bebe leche.

Software

Prover9 y Mace4
ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos
Problema de los rectángulos
Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku
Problema de las N Reinas
Problema del vecindario

Problema del vecindario

Nuestra primera aproximación va a ser numerando las casas del 0 al 4, y definiendo para cada propiedad del enunciado un predicado sobre $[0, 4]$ de forma que se verifique sobre i si la propiedad se da en la casa i -ésima. De esta forma, las afirmaciones anteriores se convierten en:

Ingles(x) \leftrightarrow Roja(x).

Espanol(x) \leftrightarrow Perro(x).

Noruego(0).

Ducados(x) \leftrightarrow Amarilla(x).

Chester(x) & Canario(y) \rightarrow vecino(x,y).

Noruego(x) & Azul(y) \rightarrow vecino(x,y).

Fortuna(x) \leftrightarrow Gato(x).

Celtas(x) \leftrightarrow Vino(x).

Ucraniano(x) \leftrightarrow Te(x).

Japones(x) \leftrightarrow Habanos(x).

Ducados(x) & Caballo(y) \rightarrow vecino(x,y).

Cafe(x) \leftrightarrow Verde(x).

Verde(x) & Blanca(y) \rightarrow sucesor(y,x).

Leche(2).

Software

Prover9 y Mace4

ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos

Problema de los rectángulos

Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku

Problema de las N Reinas

Problema del vecindario

Problema del vecindario

Donde, además, damos la definición de vecino:

```

sucesor(x,y) <-> x+1 = y & x < y.
vecino(x,y) <-> sucesor(x,y) | sucesor(y,x).

```

Las condiciones que indican que cada casa tiene al menos una nacionalidad, animal, bebida, color y tabaco asociados:

```

Ingles(x) | Espannol(x) | Ucrainiano(x) | Japones(x) | Noruego(x).
Perro(x) | Gato(x) | Caballo(x) | Cebra(x) | Canario(x).
Agua(x) | Leche(x) | Vino(x) | Te(x) | Cafe(x).
Roja(x) | Azul(x) | Amarilla(x) | Blanca(x) | Verde(x).
Celtas(x) | Fortuna(x) | Ducados(x) | Chester(x) | Habanos(x).

```

Y que cada propiedad se aplica a lo sumo a una casa:

```

Ingles(x) & Ingles(y) -> x = y.
Espannol(x) & Espannol(y) -> x = y.
Ucrainiano(x) & Ucrainiano(y) -> x = y.
Japones(x) & Japones(y) -> x = y.
...

```

Software

Prover9 y Mace4
ToulST

Ejemplos Lógica Proposicional

Problema de los mentirosos
Problema de los rectángulos
Problema de las 4 reinas

Ejemplos Lógica Primer Orden

Sudoku
Problema de las N Reinas
Problema del vecindario