

## Semana 3: Equipos de trabajo para actividades

# Actividad grupal (ojo esta semana)

Es muy importante que activéis la cuenta de correo de @comunidadunir.net ya que se envía ahí la composición de los grupos.

- ▶ La coordinación académica crea los equipos de trabajo (normalmente en torno a semana 3). Consultad la cuenta de correo anteriormente citada.
  - ▶ La **actividad 1** es **obligatoriamente grupal**: quien no realice la actividad 1 en equipo tendrá 0 (si se hace de forma individual por propia decisión) (ved clase de presentación)
  - ▶ Se hace constar en la entrega (carátula) y en la hoja de control
  - ▶ Informad de incidencias sobre equipos al equipo de **mentoría** y **contactad** entre vosotros en un plazo de **una semana**.
- 
- ▶ **En RyPA:** Podéis **intercambiaros** personas entre diferentes equipos sin pedir autorización. La razón de intercambiaros es mantener el número de miembros de los equipos aproximadamente y no dejar a nadie atrás.
  - ▶ **En RyPA:** Además, podéis **opcionalmente** hacer en equipo las otras dos actividades, con el mismo equipo o volviendo a intercambiaros. No hace falta autorización previa, sí que conste en la entrega (hoja de control).

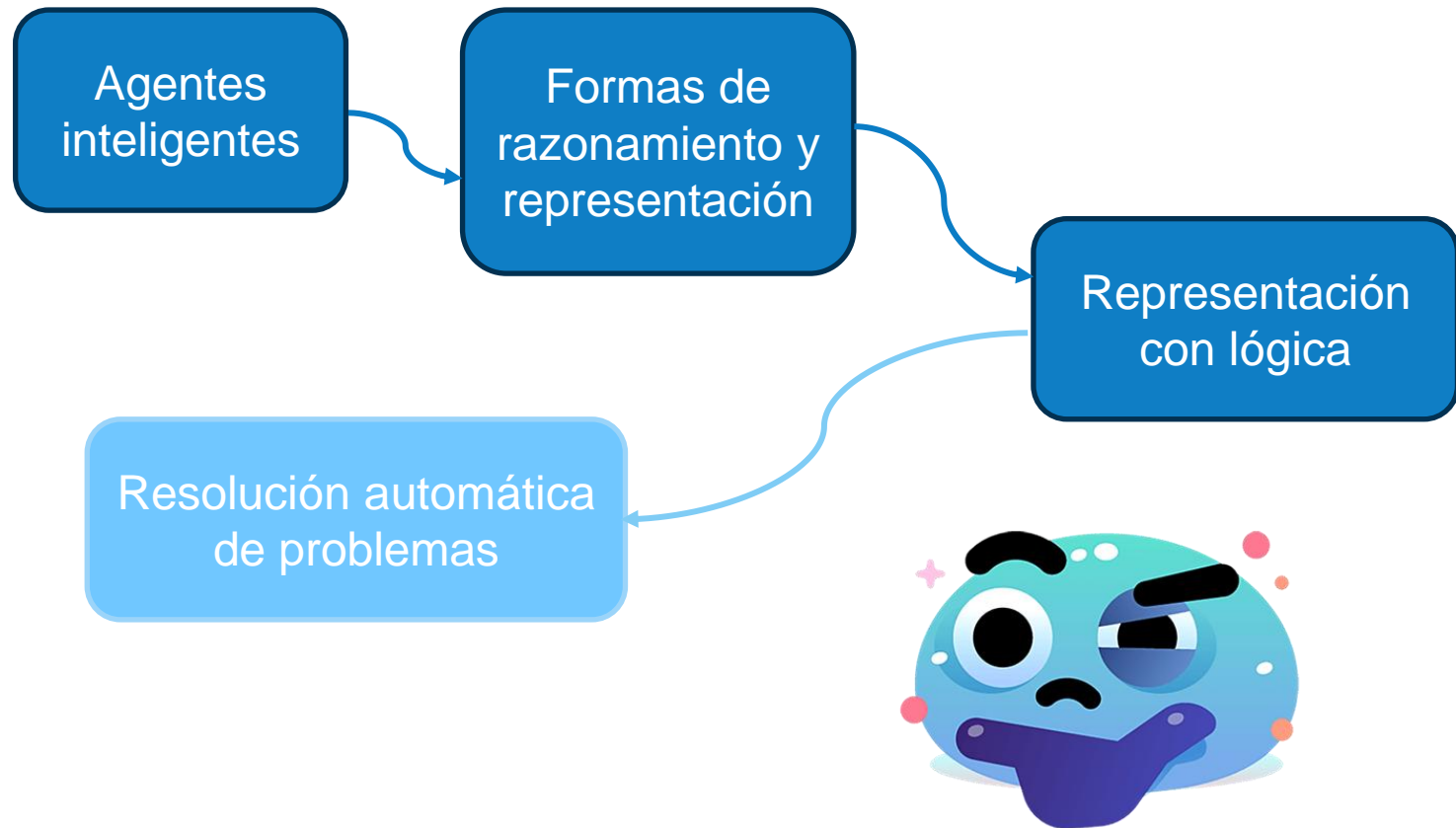


## Tema 3: Lógica y pensamiento humano

María Aurora Martínez Rey



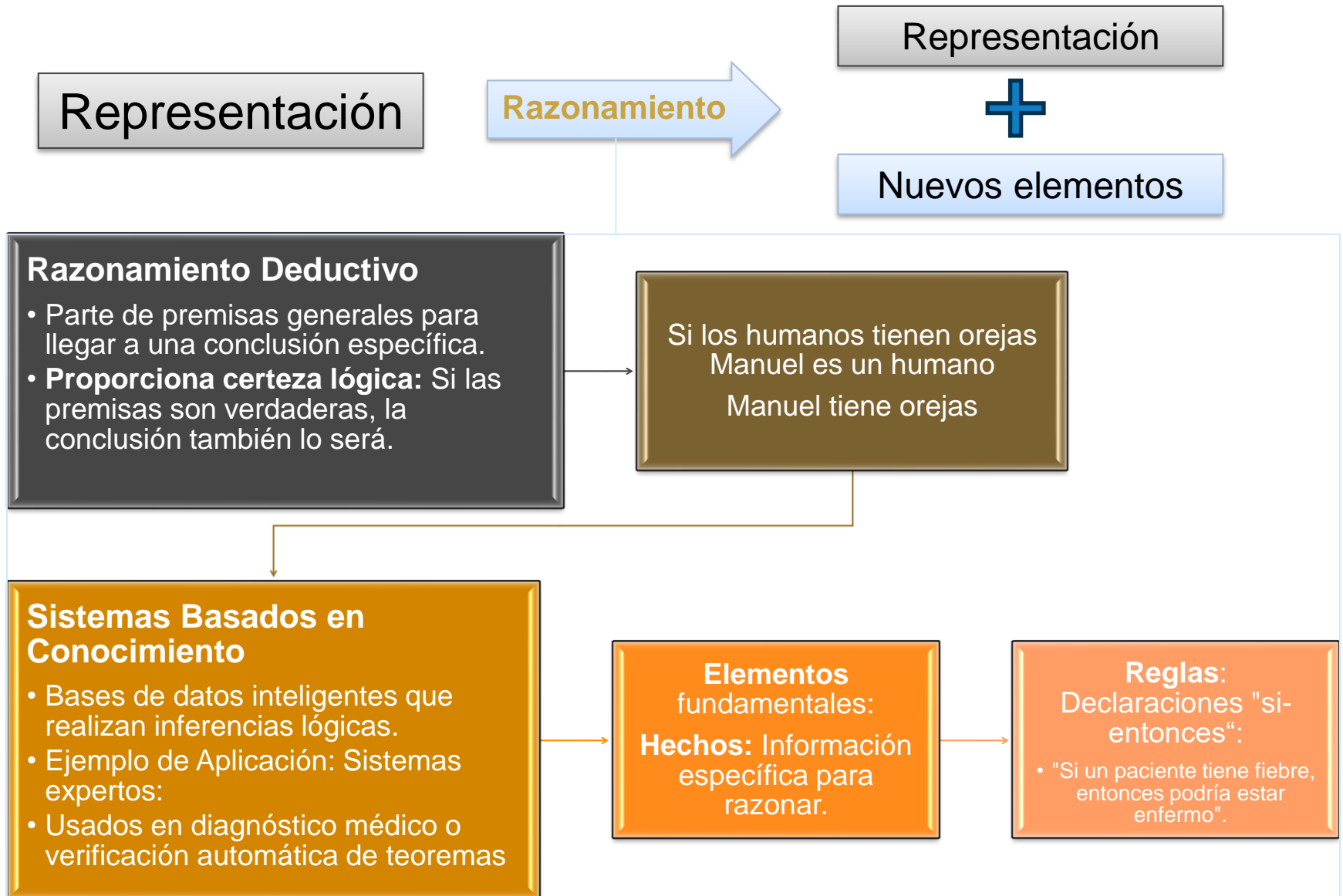
# Avanzando por la asignatura



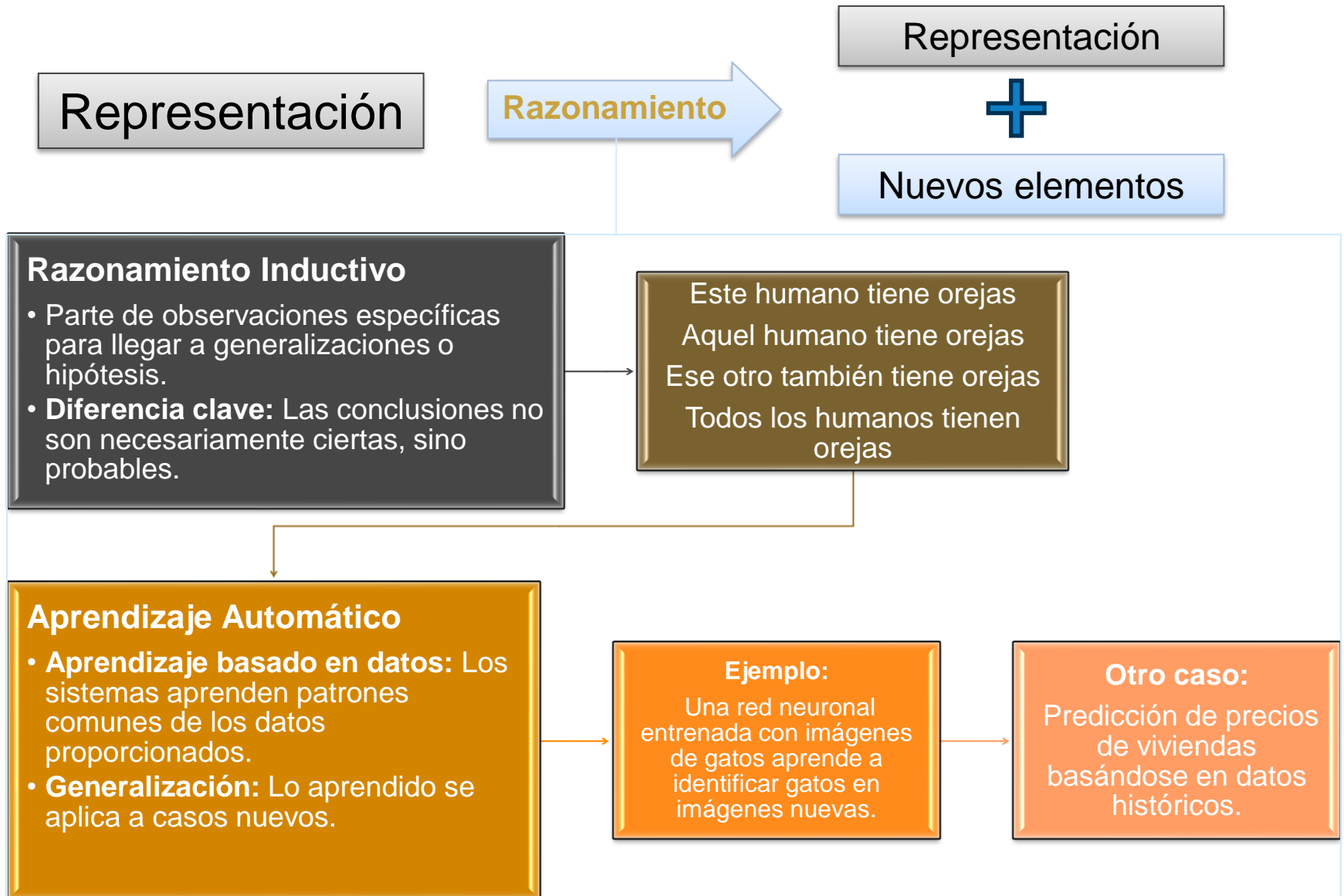


## Repaso tema 2

# Razonamiento y tipos de razonamiento



# Razonamiento y tipos de razonamiento



# Razonamiento y tipos de razonamiento



## Comparación

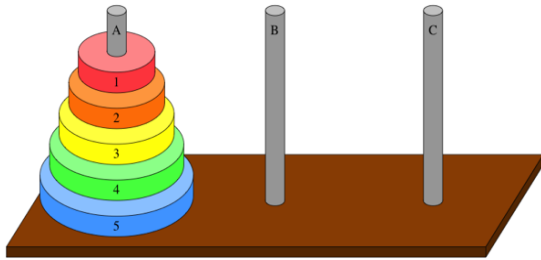
Característica	Razonamiento Deductivo	Razonamiento Inductivo
Punto de partida	Premisas generales	Observaciones específicas
Resultado	Conclusión específica	Generalización o hipótesis
Uso en IA	Sistemas basados en conocimiento	Aprendizaje automático
Ejemplo en IA	Reglas lógicas en un sistema experto	Redes neuronales, árboles de decisión



# Problema del conocimiento

- ▶ Ni humanos ni máquinas pueden acceder a la realidad directamente.
- ▶ En todos los casos se genera una representación (una imagen, quizá no exacta) interna de la realidad, con la que trabajar.
- ▶ Esta representación puede ser:  
completa/incompleta, precisa/ambigua, correcta/incorrecta,  
relevante/irrelevante.
- ▶ Aunque la realidad no cambie, la representación sí puede modificarse
- ▶ Podemos representar las situaciones (**estados**) pero también los **procesos** (por ejemplo usando reglas)

# Completa/Precisa/Correcta/Relevante



El juego consiste en pasar todos los discos al eje C, con **restricciones**

**Objetos:** discos, ejes, mesa

Discos: tamaño, posición

Posición: eje, altura

Ejes: capacidad

**Restricción 1:** sólo se puede mover el disco superior

**Restricción 2:** un disco siempre sobre **mesa u otro mayor**

¿Son relevantes el color, la mesa y la posición de los ejes?

¿Es precisa la restricción?

# Requisitos de la representación en IA

- ▶ **Formal**: sin ambigüedades
- ▶ **Expresiva**: suficientemente rica para capturar los aspectos relevantes
- ▶ **Natural**: análoga a las formas naturales de expresión del conocimiento (esto no ocurre en Machine Learning)
- ▶ **Tratable**: se debe poder manipular para reflejar los procesos de razonamiento y generar respuestas

# Formas de representación usadas en IA

Marcos: que son representaciones estereotipadas de situaciones, conceptos, ideas u objetos. Los procesos de inferencia se realizan por medio de la jerarquía u ordenación de los mismos.

Lógica: representación e inferencia basada en modelos lógicos formales.

Reglas: por medio del encadenamiento hacia adelante se produce la inferencia de sentencias condicionales basadas en estructuras de la forma SI-ENTONCES.

Restricciones: interrelaciones entre variables por medio de un dominio de valores posibles. La inferencia se realiza a través de técnicas de propagación de restricciones o consistencia de arco.

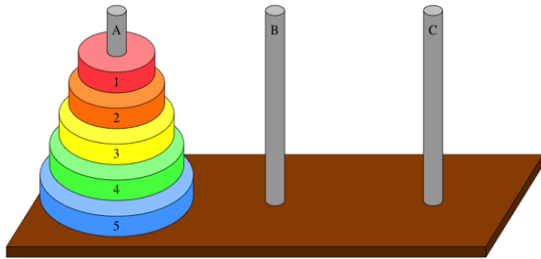
Red bayesiana: red causal basada en valoraciones cuantitativas probabilistas de la influencia o relaciones entre antecedentes y consecuentes.

Lógica difusa: valoraciones semánticas experienciales que nos identifican valores de pertenencia a conjuntos de características borrosas. El proceso de inferencia se realiza por medio de modelos matemáticos de lógica especializada para estos entornos.

Figura 1. Técnicas de representación.



# Formal/Expresiva/Natural/Tratable



¿Y si los discos pudieran estar en cualquier orden, y no siempre los pequeños sobre los grandes?

Representación	Formal	Expresiva	Natural	Tratable																								
Hay unos discos de distintos diámetros insertados en tres ejes.	No	Sí	Sí	No																								
Disco1En1, Disco2En2, ...	Sí	Poco	Sí	Sí																								
Ejes=[ [1,2,3,4,5], [], [] ]	Sí	Sí	Sí	Sí																								
	No	Sí	Sí	No																								
<table><tr><th></th><th>Disco1</th><th>Disco2</th><th>Disco3</th><th>Disco4</th><th>Disco5</th></tr><tr><td>Eje1</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>Eje2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Eje3</td><td></td><td></td><td></td><td></td><td></td></tr></table>		Disco1	Disco2	Disco3	Disco4	Disco5	Eje1	X	X	X	X	X	Eje2						Eje3						Sí	No	No	Sí
	Disco1	Disco2	Disco3	Disco4	Disco5																							
Eje1	X	X	X	X	X																							
Eje2																												
Eje3																												

# Índice de contenidos teóricos

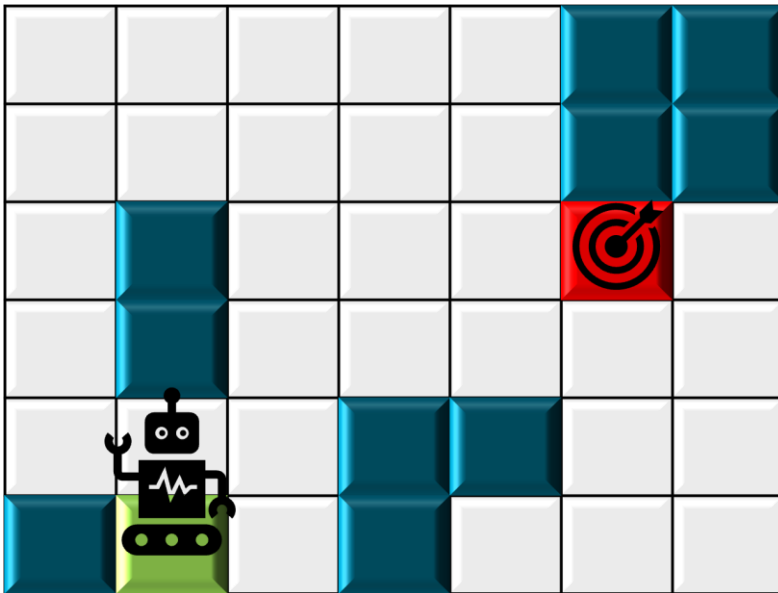
## Tema 3

- ▶ Lógica de proposiciones
- ▶ Proceso de decisión: tablas de verdad
- ▶ Lógica de predicados
- ▶ Otras lógicas



## Problema tema 3

# Problema: Encuentra la Salida del Laberinto



Imagina que eres el ingeniero encargado de programar un robot que debe navegar a través de un laberinto. Este robot:

- Puede moverse en cuatro direcciones: arriba, abajo, izquierda y derecha.
- Debe evitar chocar con paredes y permanecer dentro de los caminos permitidos.

El laberinto está representado como una cuadrícula donde cada celda puede estar libre o bloqueada (Azul oscuro). Tu tarea será modelar este problema usando lógica proposicional y diseñar las reglas necesarias para que el robot (celda verde) tome decisiones inteligentes y encuentre la salida (celda roja).



# Preguntas para el problema

1. ¿Cómo representarías el estado del laberinto para que el robot lo "comprenda"?
2. ¿Qué reglas lógicas puede seguir el robot para decidir cada movimiento?
3. ¿Cómo podríamos evaluar si el camino elegido cumple con el objetivo?



# Lógica de proposiciones

# ¿Qué es?

Mediante el uso de un lenguaje formal (alfabeto + reglas), estudia los valores de verdad de las proposiciones, independientemente de su contenido.

## Sintaxis

- Conjunto de símbolos (letras y conectores) que permiten escribir o formalizar correctamente las fórmulas... Alfabeto.

$$\{q, r, s, p, p_1, p_2, \dots, \perp, \top\}$$
$$\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$$

## Semántica

- Distintas interpretaciones de una fórmula permiten que esta sea aceptada como verdadera o como falsa.... Construcción del significado... Reglas.

1.  $p_k$ .
2.  $(\neg \Box)$ .
3.  $(\Box * \Box)$ .

# Algunos términos...

- Proposición: Enunciado lingüístico que puede ser verdadero o falso. Puede ser atómica(simple) o molecular(compleja):

$$(p \wedge q).$$

- Antecedente y Consecuente.

$$(p \wedge q) \rightarrow r.$$

- Interpretación: Es lo que necesitamos para decidir el valor de verdad de una fórmula compleja. La interpretación  $I$  satisface a la fórmula  $\varphi$ , formalmente:  $I \models \varphi$ .



# Conectores, Precedencia y Paréntesis

Negación

$(\neg p)$

Conjunción

$(p \wedge q)$

Disyunción

$(p \vee q)$

Implicación

$(p \Rightarrow q)$

Bicondicional

$(p \Leftrightarrow q)$

1º Nivel	$\neg$
2º Nivel	$\wedge$
3º Nivel	$\vee$
4º Nivel	$\Rightarrow \Leftrightarrow$

$p \vee q \Rightarrow r$

# Características de la lógica de proposiciones

ES UNA FORMA DE REPRESENTACIÓN FORMAL QUE PERMITE EXPRESAR CONOCIMIENTO DE FORMA NO AMBIGUA

CONOCEMOS UN ALGORITMO PARA SABER SI UNA INFERENCIA ES CORRECTA

TENEMOS CONOCIMIENTO DE CÓMO HACER INFERENCIA

# ¿Para qué sirve la Lógica de Proposiciones?

Razonamiento matemático

Desambiguación del lenguaje

Creación de especificaciones de hardware o software

Circuitaría digital

Escribir código

Analizar código

Solución de puzzles lógicos

# Lógica proposicional

Lógica matemática: Lenguaje **formal** con una gramática y sintaxis

- ▶ **Proposición simple:**  $p, q$ , llueve ... (afirmaciones, enunciados declarativos)
- ▶ **Proposición compuesta (fórmula):**  $A, B$ , (mayúsculas). Combinación de simples mediante conectivas

- **Condicional (Implicación):**
  - Si  $A$  entonces  $B = A \rightarrow B$
  - Solo  $A$  si  $B = A \rightarrow B$
- **Equivalencia:**  $A$  si y solo si  $B \quad A \leftrightarrow B$

Importante: prioridad de  $\rightarrow$  es menor que la de  $\wedge$  y  $\vee$ :

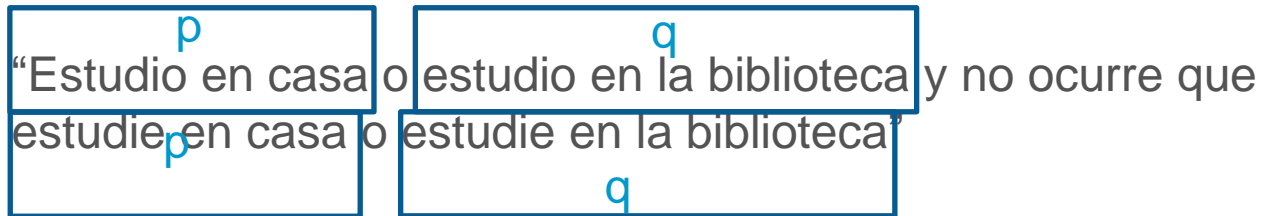
$p \rightarrow q \wedge \neg r$  se agrupa  $p \rightarrow (q \wedge (\neg r))$

Nota: El condicional está en la base del concepto de REGLA

	Prio- ridad	Nombre	Ejemplo
$\neg$ $\sim$	1	NO, NOT	No tengo ganas de ir: $\neg p$
$\wedge$	2	Y, AND	Llueve y hace frío: $l \wedge f$
$\vee$	2	O, OR	Como carne o como pescado: $c \vee p$
$\rightarrow$	3	IMPLICA	Si voy, tendré problemas: $v \rightarrow p$
$\leftrightarrow$	4	EQUIVALE A	Iré si y solo si tú vienes: $y \leftrightarrow t$



# Lógica proposicional: representación



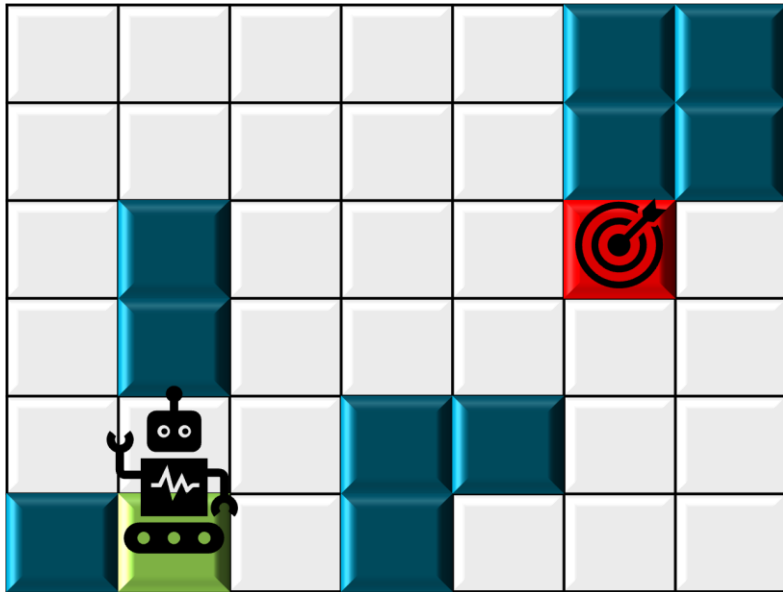
Se formaliza:  $(p \vee q) \wedge \neg(p \wedge q)^1$

¡cuidado paréntesis!

<sup>1</sup>Ver diferencia: "estudio en casa o en la biblioteca y no estudio en casa y en la biblioteca (ambas a la vez)", que es una frase con "sentido":

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

# Representando un problema



Para aplicar un procedimiento automático hay más elementos relevantes:

1. La posición del agente (**estado**)
2. La posición **objetivo** (o conjunto de objetivos)
3. Los movimientos posibles (**operadores** o **acciones**): serán aquellos que muevan en horizontal y vertical una casilla
4. La posición de las paredes (**restricciones** al movimiento)
5. Tal vez: el **coste** de realizar cada acción, en cada situación

Cada **estado**: posición del robot (+ mapa). Ej: (X,Y,Paredes)

La **solución**: lista (conjunto ordenado) de acciones (corresponde a un camino): un ejemplo sería el siguiente: D, U,U,U,D,D,D.

# Problema: Encuentra la Salida del Laberinto

42	41	40	39	38	37	36
29	30	31	32	33	34	35
28	27	26	25	24	23	22
15	16	17	18	19	20	21
14	13	12	11	10	9	8
1	2	3	4	5	6	7

R1:  $\text{Pos02} \wedge \neg \text{Pared03} \rightarrow \text{Pos03}$

R2:  $\text{Pos03} \wedge \neg \text{Pared12} \rightarrow \text{Pos12}$

R3:  $\text{Pos12} \wedge \neg \text{Pared11} \rightarrow \text{Pos11}$

... (así sucesivamente las combinaciones para los adyacentes)...

RF1: Si  $\text{Pos23} \wedge \text{Salida23} \rightarrow \text{Salvado}$

... (así sucesivamente para cada posición X)

...

Un problema concreto:  $\text{Pos12} = \text{True}$ , resto  $\text{Pos} = \text{False}$ ,  $\text{Pared11} = \text{True}$ , etc.

La pregunta: ¿Se puede salir del laberinto? se transforma en:

1. Las reglas R valen para todos los laberintos de ese tamaño
2. Codifica el caso concreto poniendo a V y F las proposiciones correspondientes
3. ¿Se deduce "Salvado" de las premisas y las reglas de movimiento para ese caso concreto?

# Lógica no monótona

En un agente, la verdad de las proposiciones irá **variando**

42	41	40	39	38	37	36
29	30	31	32	33	34	35
28	27	26	25	24	23	22
15	16	17	18	19	20	21
14	13	12	11	10	9	8
1	2	3	4	5	6	7

Una solución será una **lista** (conjunto ordenado) de **acciones** (corresponde a un camino) que lleva a la posición objetivo.

*MoverDerecha\_1020,*  
*MoverDerecha\_2030\_... etc., muchas*  
*versiones de cada regla "MoverDerecha"*

## CON PROPOSICIONES

Cualquier situación concreta (estados): una proposición por casilla, sólo está a True la que indica la posición:

Pos00=False

Pos10=True

Pos20=False , .. etc. ( $X * Y$  proposiciones)

Mapa (restricciones):

Pared00=False

Pared10=False ... ( $X*Y$  proposiciones)

Operadores (Conocimiento de inferencia, procedimientos, acciones):

**Ojo, generales para todo laberinto**

*MoverDerecha* (¡muchas reglas!):

SI Pos00=True Y Pared10 =False HACER:

Pos00=False Y Pos10=True

(... muchas, una por cada movimiento...)

*MoverIzquierda* (¡muchas reglas!): ...

*MoverAbajo*: ...

*MoverArriba*: ...

# Preguntas para el problema

Ahora, usando lógica...

Puedes empezar a practicar, escribiendo reglas simples en lógica proposicional para representar el problema y sus reglas. Finalmente, te animo a experimentar con herramientas que pueden ayudarte a evaluar la validez de tu razonamiento, como las tablas de verdad

<https://web.stanford.edu/class/cs103/tools/truth-table-tool/>

Observación: ¡en el examen no se puede usar la calculadora!

# Ejemplo de uso de lógica para razonar

Putting It All Together: The Logic Behind the Forensic Scientific Method and the Inferential Test

<https://www.heartlandforensic.com/writing/putting-it-all-together-the-logic-behind-the-forensic-scientific-method-and-the-inferential-test/>



# Construcción de tablas de verdad



# Tablas de verdad en lógica de proposiciones

Construcción de la **tabla de verdad** de una fórmula F:

1. Las **proposiciones simples** tienen valor de verdad 0 o 1
2. Las **conectivas** operan con los valores de las proposiciones que conectan de cierta forma (tablas de las **conectivas**)
3. Las **fórmulas** por lo tanto tienen un valor de verdad 0 o 1 para cada combinación de valores de sus proposiciones simples (**interpretación** de la fórmula)
4. La **tabla de verdad** de una fórmula contiene **una fila por posible interpretación** (combinación de valores para las proposiciones **simples** que intervienen)

Valores de TablaVerdad (F)	F es una ...
Todos 1	Tautología
Todos 0	Contradicción
Algunos 0 y otros 1	Contingencia,falacia,inconsistencia

# Reglas

p	$\neg p$
1	0
0	1

p	q	$p \wedge q$
1	1	1
1	0	0
0	1	0
0	0	0

p	q	$p \vee q$
1	1	1
1	0	1
0	1	1
0	0	0

p	q	$p \Rightarrow q$
1	1	1
1	0	0
0	1	1
0	0	1

p	q	$p \Leftrightarrow q$
1	1	1
1	0	0
0	1	0
0	0	1

# Ejemplo

Evaluación

$$(p \vee q) \wedge (\neg q \vee r)$$

$$(p \vee q) \wedge (\neg q \vee r)$$

$$(1 \vee 0) \wedge (\neg 0 \vee 1)$$

$$1 \wedge (\neg 0 \vee 1)$$

$$1 \wedge (1 \vee 1)$$

$$1 \wedge 1$$

$$1$$

$$p^i = 1$$

$$q^i = 0$$

$$r^i = 1$$

Asignación de valores  $i$

Tabla de Verdad

p	q	r	$(p \vee q)$	$\neg q$	$(\neg q \vee r)$	$(p \vee q) \wedge (\neg q \vee r)$
1	1	1	1	0	1	1
1	1	0	1	0	0	0
1	0	1	1	1	1	1
1	0	0	1	1	1	1
0	1	1	1	0	1	1
0	1	0	1	0	0	0
0	0	1	0	1	1	0
0	0	0	0	1	1	0

# Método automático

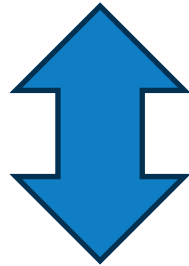
La construcción de la **tabla de verdad** de una fórmula  $F$  de lógica de proposiciones se puede resolver con un programa muy sencillo.

<https://web.stanford.edu/class/cs103/tools/truth-table-tool/>

<https://calculator-online.net/truth-table-calculator/>

# Importancia de las Tautologías

SI  $\neg(p \vee q) \rightarrow (p \rightarrow r)$  ES UNA TAUTOLOGÍA



TEOREMA DE LA DEDUCCIÓN

ENTONCES DE LA PREMISA

$\neg(p \vee q)$

ES UNA DEDUCCIÓN VÁLIDA  $(p \rightarrow r)$

Se transforma cualquier implicación ( $\rightarrow$ ) en una deducción (y viceversa)

# Teorema de la deducción

Dadas las **fórmulas**  $A, B, C, Q, \dots$  (pueden ser compuestas o simples)

- Si y solo si

**F:**  $A \wedge B \wedge C \dots \rightarrow Q$   
es tautología

- La deducción  $A, B, C, \dots \Rightarrow Q$  es **correcta**,  
es un **razonamiento válido** (de las **premisas**  
 $A, B, C$  se **deduce**  $Q$ )

Tabla de:  $A \wedge B \wedge C \dots \rightarrow Q$   
es 1 para **toda**  
interpretación



1. A  
2. B  
3. C  
...  
Q

Número finito de  
filas:  $2^N$

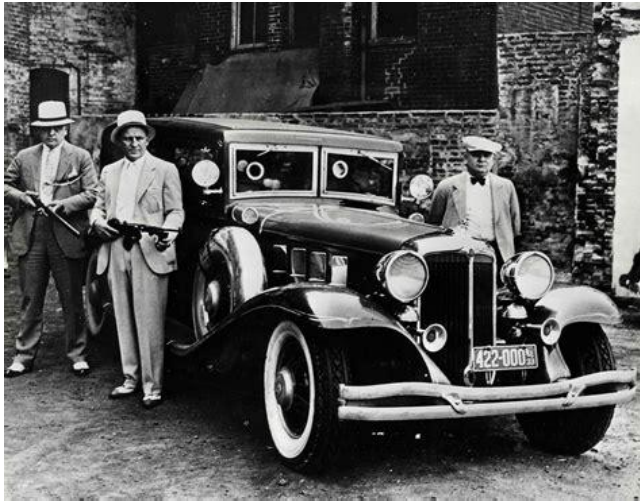
N: número de letras  
de proposición

p	q	r	...	A	B	C	...	Q	$A \wedge B \wedge C \wedge \dots$	$F: A \wedge B \wedge C \wedge \dots \rightarrow Q$
0	0	0		?	?	?		?	?	1
1	0	0		?	?	?		?	?	1
... (cada fila es una interpretación de <b>F</b> que parte de una combinación de valores para p,q,r ... ) ...										
1	1	1		?	?	?		?	?	1

Nota: En lógica de predicados existen también tablas de verdad, pero no siempre se pueden enumerar todos los casos

# Problema adicional (proposiciones)

*¿Cómo crear un sistema que razona automáticamente?*



Se ha producido un atraco y se sabe que las personas que participaron se fueron conduciendo un automóvil. Se interroga a tres conocidos delincuentes Par, Qun y Rag. La policía obtiene la siguiente información:

- (1) Par, Qun y Rag son los únicos posibles culpables
- (2) Rag nunca hace un trabajo sin que esté Par (no excluye otros) como cómplice.
- (3) Qun no sabe conducir.

Fuente: <https://www.vintag.es/2017/03/17-amazing-vintage-photographs-show.html>

¿Cómo podemos identificar las personas culpables?

Adaptado de: <https://brilliant.org/problems/you-be-the-detective-1/>



# Representación del problema adicional (I)

Proposiciones:

P: Par es culpable

Q: Qun es culpable

R: Rag es culpable

(1) *Par, Qun y Rag son los únicos posibles culpables*

$P \vee Q \vee R$  (Premisa 1)

(2) *Rag nunca hace un trabajo sin que esté Par (no excluye otros) como cómplice. Es decir, si estuvo Rag, estuvo Par.*

$R \rightarrow P$  (Premisa 2)

(3) *Qun no sabe conducir. Es decir, si Qun estuvo, tuvo que estar acompañado de otra u otras personas.*

$Q \rightarrow P \vee R$  (Premisa 3)

Identificar culpables: ¿se deduce P de lo anterior? ¿Q? ¿R?

# Problema de clase, razonamientos válidos

¿Es Par culpable?

$P \vee Q \vee R, R \rightarrow P, Q \rightarrow P \vee R \Rightarrow P$

Fórmula Culpable(Pat):

$$(P \vee Q \vee R) \wedge (R \rightarrow P) \wedge (Q \rightarrow P \vee R) \rightarrow P$$

Prueba en: <https://web.stanford.edu/class/cs103/tools/truth-table-tool/>

P	Q	R	Culpable(Pat)
F	F	F	T
F	F	T	T
F	T	F	T
F	T	T	T
T	F	F	T
T	F	T	T
T	T	F	T
T	T	T	T

Tautología

Por tanto, podemos concluir  
que en efecto Par es culpable

# Problema de clase, razonamientos válidos

¿Es Q culpable? ¿Es R culpable?

Fórmula Culpable(Q):  $(P \vee Q \vee R) \wedge (R \rightarrow P) \wedge (Q \rightarrow P \vee R) \rightarrow Q$

Fórmula Culpable(R):  $(P \vee Q \vee R) \wedge (R \rightarrow P) \wedge (Q \rightarrow P \vee R) \rightarrow R$

Prueba en: <https://web.stanford.edu/class/cs103/tools/truth-table-tool/>

P	Q	R	Culpable(Qun)
F	F	F	T
F	F	T	T
F	T	F	T
F	T	T	T
T	F	F	F
T	F	T	F
T	T	F	T
T	T	T	T

Contingencia

P	Q	R	Culpable(Rag)
F	F	F	T
F	F	T	T
F	T	F	T
F	T	T	T
T	F	F	F
T	F	T	T
T	T	F	F
T	T	T	T

Contingencia

Por tanto, no podemos asegurar la culpabilidad de Q ni de R

# Problemas para casa (proposiciones)

¿Es válido el razonamiento siguiente?

Si X es un cuervo, entonces X vuela

Si X es un pájaro, entonces X vuela

Por tanto, Si X es un cuervo, entonces X es un pájaro

*Prueba en:* <https://web.stanford.edu/class/cs103/tools/truth-table-tool/>

¿Es válido el razonamiento siguiente?

Si X es un cuervo, entonces X es un ave

Si X es un ave, entonces X vuela

Por tanto, Si X es un cuervo, entonces X es un ave

*Prueba en:* <https://web.stanford.edu/class/cs103/tools/truth-table-tool/>



# Lógica de predicados

# Lógica de predicados

**Extiende** la lógica de proposiciones. No es decidible en general.

- ▶ **Términos**: constantes (**Pepe**, **a**, **2**), variables (**X**, **T**) (mayúsculas)
- ▶ **Dominio**: conjunto al que pertenecen los términos (personas, cosas..)
- ▶ **Predicado**: nombre, seguido de 1 a N términos
- ▶ **Funciones**: nombre, seguido de 1 a N términos, aplica sobre otro término

nombrePredicado(términos)

verde(semáforo1)

padre(Juan,María)

nombreFunción(términos) = término

suma(1,2) = 3

	Prioridad	Nombre	Ejemplos
$\forall$	1	Cuantificador universal, "para todo"	Todos los estudiantes sacan buena nota: $\forall X (\text{estudiantes}(X) \rightarrow \text{buenaNota}(X))$
$\exists$	1	Cuantificador existencial "existe"	Algunos estudiantes destacan: $\exists T (\text{estudiante}(T) \wedge \text{destaca}(T))$ Toda persona tiene algún progenitor: $\forall X (\text{persona}(X) \rightarrow \exists Y \text{ progenitor}(X,Y))$

# Laberinto con lógica de predicados

42	41	40	39	38	37	36
29	30	31	32	33	34	35
28	27	26	25	24	23	22
15	16	17	18	19	20	21
14	13	12	11	10	9	8
1	2	3	4	5	6	7

**Solución:** lista (conjunto ordenado) de operadores (corresponde a un camino)

*MoverDerecha, MoverDerecha, MoverAbajo, MoverAbajo, MoverDerecha, MoverDerecha, MoverAbajo, MoverAbajo, MoverIzquierda*

## CON PREDICADOS

Estado (Conocimiento del problema, situación):

**pos**(X,Y,Z) con  $Z=\{\text{pared, libre, agente}\}$  es True o False según el valor de X e Y

- (X\*Y predicados) (incluye mapa)
- límites del mapa (maxX, minY, etc.)

Operadores (Conocimiento de inferencia, procedimientos, acciones):

Predicado 2

*MoverDerecha* (¡una regla!):

SI **pos**(X,Y,agente)=True Y (X < maxX)

**pos**(X+1,Y,libre)=True

HACER:

**pos**(X+1,Y,agente)=True Y

**pos**(X,Y,libre)=True Y **pos**(X,Y,agente)=False

Y **pos**(X+1,Y,libre)=False

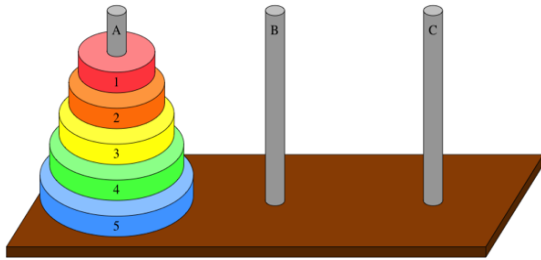
*MoverIzquierda* (una regla): ...

*MoverAbajo* (una regla): ...

*MoverArriba* (una regla): ...



# Otro ejemplo en lógica de predicados



## Objetos:

términos constantes

d1,d2,d3,d4,d5 (discos)

ejeA, ejeB, ejeC (ejes)

mesa

## Predicados:

propiedades y relaciones

**en** (Eje,Disco),

**tam** (Disco,Número)

**sobre** (X,Disco)

(X: Disco o mesa)

Si consideramos la representación de un estado (no de las reglas de movimiento):

¿Completa?

¿Precisa?

¿Correcta?

¿Relevante?

¿Falta algo?

¿Hay ambigüedad?

¿Hay errores o permite inconsistencias?

¿Sobra algo?

## Situación de la figura:

en(ejeA,d1), en(ejeA,d2),

en(ejeA,d3),en(ejeA,d4),

en(ejeA,d5),

tam(d1,1),tam(d2,2),

tam(d3,3),tam(d4,4),

tam(d5,5),

sobre(d2,d1),sobre(d3,d2),

sobre(d4,d3),sobre(d5,d4),

sobre(mesa,d5)



# Programación lógica

# Prolog

Para mostrar un ejemplo que contiene elementos de un lenguaje de primer orden vamos a trabajar sobre la siguiente base de conocimientos:

## Hechos:

1. Atlanta se encuentra en Georgia.
2. Houston y Austin se encuentran en Texas.
3. Toronto se encuentra en Ontario.

Que, usando el predicado `located_in`, podemos representar con las siguientes clausulas:

```
located_in(atlanta,georgia). % Clause 1
located_in(houston,texas).  % Clause 2
located_in(austin,texas).   % Clause 3
located_in(toronto,ontario). % Clause 4
```

## Reglas:

1. Lo que está en Georgia o Texas, también está en USA.
2. Lo que está en Ontario, también está en Canadá.
3. Lo que está en USA o Canadá, también está en Norte América.

Que podemos representar con las siguientes clausulas (geo.pl):

```
located_in(X,usa) :- located_in(X,georgia). % Clause 5
located_in(X,usa) :- located_in(X,texas).   % Clause 6
located_in(X,canada) :- located_in(X,ontario). % Clause 7
located_in(X,north_america) :- located_in(X,usa). % Clause 8
located_in(X,north_america) :- located_in(X,canada). % Clause 9
```

Observa que al estar trabajando con predicados que sí reciben argumentos ya no es necesario usar la directiva vista en el ejemplo anterior.

Vamos a ver cuál es el árbol de deducción que sigue Prolog para resolver la siguiente clausula:

```
located_in(toronto,north_america).
```

Base de hechos

Base de reglas  
(implicaciones)

PROLOG online:

<https://swish.swi-prolog.org/>

Consulta

Fuente: apuntes de clase

# Resumen y recapitulación

- ☐ ¿Conocemos la sintaxis de la lógica de proposiciones y la lógica de predicados?
- ☐ ¿Conocemos la definición de interpretación, tautología, contradicción y contingencia/falacia/inconsistencia?
- ☐ ¿Sabemos cómo usar una tabla de verdad para conocer si una deducción es correcta?
- ☐ ¿Sabemos calcular la tabla de verdad de cualquier fórmula de lógica de proposiciones?

# Desafío



Has rescatado a la hija del Rey de unos piratas y, como recompensa, te permiten buscar el tesoro del reino.

**Frente a ti hay tres cofres.**

El Rey te dice: '**Uno de estos cofres contiene el tesoro. Elige sabiamente.**'

Pero la Reina añade una condición: '**Exactamente dos de las inscripciones en los cofres son falsas, y una es verdadera.**'

Tu tarea: Usando lo que hemos aprendido, debes determinar **en qué cofre está el tesoro.**

**Analiza la información, formula el problema y encuentra la solución correcta.**



[www.unir.net](http://www.unir.net)