

# Introducción a la Representación del Conocimiento

Pablo Barceló

# ¿Qué es la representación del conocimiento?

La **representación del conocimiento** es una de las áreas fundamentales de la Inteligencia Artificial (AI).

- ▶ Se encarga de estudiar cómo el conocimiento puede ser representado **simbólicamente** y manipulado **automáticamente** por programas de razonamiento.
- ▶ O en términos más sencillos, es el área de AI que se encarga de estudiar como el conocimiento contribuye al comportamiento inteligente.

Nosotros estudiaremos los simbolismos más utilizados para representar conocimiento, y algunos de los procedimientos computacionales que se han desarrollado para razonar con ellos.

# ¿Qué hace especial a la representación del conocimiento?

Hay muchas otras formas de estudiar el conocimiento y la inteligencia: Psicología, neurociencia, etc.

¿Qué es lo que hace diferente a la representación del conocimiento?

- ▶ En vez de estudiar al ser humano detenidamente (su mente, sus estados mentales, etc), **estudia lo que los seres humanos saben**.
- ▶ La premisa básica es que los seres humanos actúan inteligentemente porque saben muchas cosas acerca de muchas cosas, y porque pueden aplicar este conocimiento a su ambiente para cumplir sus objetivos.
- ▶ Se pregunta que es lo que un ser humano necesita saber para actuar inteligentemente.

# Conocimiento y comportamiento inteligente

Una característica fundamental es que el comportamiento inteligente parece estar condicionado por el conocimiento:

- ▶ Es decir, mientras más sabemos más inteligentemente actuamos (????).

Para una amplia gama de actividades, tomamos decisiones basadas en lo que sabemos **casi sin esfuerzo e inconcientemente**.

Cuando decimos que alguien ha actuado tontamente no es tanto porque no sabía lo que estaba haciendo, sino porque fue incapaz de utilizar lo que ya sabía en lo que estaba haciendo (actuar sin pensar).

# Aplicaciones de la representación del conocimiento

Las aplicaciones de la representación del conocimiento incluyen la modelación y simulación computacional de actividades inteligentes tales como:

- ▶ **Razonamiento deductivo** ¿Qué puede ser inferido válidamente acerca de lo que sabemos? ¿Cuáles son las lógicas que gobiernan nuestro sistema?
- ▶ **Razonamiento matemático** No sólo los mecanismos de demostración, sino idealmente también las intuiciones, heurísticas, etc.
- ▶ **Razonamiento de sentido común** Razonamiento a partir de conocimiento incompleto, inconsistente, etc.
- ▶ **Diagnóstico** ¿Cómo conocer la causa del mal funcionamiento de un sistema dada una cierta evidencia?

# Aplicaciones de la representación del conocimiento

- ▶ **Planificación** ¿Cuál es una secuencia de acciones en mi dominio que me lleven a un objetivo específico?
- ▶ **Razonamiento sobre conocimiento** ¿Qué sabe mi adversario? ¿Qué de lo que sabe puede revelarle alguna información secreta?
- ▶ **Revisión de conocimiento** ¿Cómo puedo actualizar la información sobre mi dominio, sin contradecir a la anterior?
- ▶ ...

Las aplicaciones de la representación del conocimiento van mucho más allá de AI:

- ▶ Bases de datos.
- ▶ Ingeniería de software.
- ▶ Ontologías.
- ▶ Web semántica.
- ▶ Muchos otros.

Trataremos de ver algunas de éstas aplicaciones con más detalle.

# ¿Por qué tanta lógica?

A lo largo del curso tendremos que estudiar bastante de **lógica**.  
¿Por qué?

# ¿Por qué tanta lógica?

A lo largo del curso tendremos que estudiar bastante de **lógica**.

¿Por qué?

Porque la lógica es el estudio de la relación de consecuencia entre proposiciones. Formaliza el estudio de qué puede ser deducido desde un base de conocimiento.

# ¿Por qué tanta lógica?

A lo largo del curso tendremos que estudiar bastante de **lógica**.  
¿Por qué?

Porque la lógica es el estudio de la relación de consecuencia entre proposiciones. Formaliza el estudio de qué puede ser deducido desde un base de conocimiento.

Aunque comúnmente ocuparemos los dos lenguajes de representación más populares (lógica proposicional y de primer orden), es importante detallar que la lógica es tan flexible como se quiera.

Podemos ajustar nuestro lenguaje a nuestros requerimientos (subconjuntos o superconjuntos de primer orden, lógicas paraconsistentes, etc).

# ¿Por qué tanta lógica?

Además, los lenguajes lógicos presentan varias ventajas:

- ▶ Estudiados por más de 100 años.
- ▶ Sintáxis y semántica muy precisa.
- ▶ Son lenguajes declarativos.
- ▶ Poder expresivo relativamente claro.
- ▶ Su complejidad es bien conocida.
- ▶ Muchos de ellos tienen procedimientos de manipulación simbólica asociados (incluso implementados).

# Algunas desventajas de la lógica

## Desventajas de la lógica:

- ▶ Incluso los lenguajes más básicos son indecidibles.
- ▶ Difíciles de implementar.
- ▶ A los programadores no les gusta utilizar variables.

# Características del conocimiento que queremos representar

Nuestro conocimiento debe ser almacenado en nuestro sistema computacional de tal forma que:

- ▶ Sea fácil de acceder;
- ▶ sea fácil de procesar computacionalmente; y
- ▶ sea fácil obtener el **conocimiento implícito** que hay en él (Si **todos los hombres son mortales** y **Sócrates es hombre**, entonces implícitamente **Sócrates es mortal**).

Además debería ser fácil razonar a meta-nivel acerca de ese conocimiento.

# Características del conocimiento que queremos representar

Quisiéramos además que nuestro conocimiento sea fácil de actualizar (tanto agregando como calificando conocimiento).

Por último, queremos que el conocimiento sea fácil de consultar.

Un ejemplo muy básico que cumple estos requerimientos es el de las bases de datos relacionales.

# Bases de datos relacionales

Las bases de datos son una forma simple de representar conocimiento.

¿Qué es una base de datos? ¿Cómo se ve?

# Bases de datos relacionales

Las bases de datos son una forma simple de representar conocimiento.

¿Qué es una base de datos? ¿Cómo se ve?

employee		
NAME	POSITION	SALARY
pepe	jefe	1000000
juan	junior	150000
mario	secretario	300000
.teresa	jefe	1500000

# Bases de datos relacionales: Modelo de datos

En primer lugar, proveen un **modelo de datos**: explicita como los datos serán almacenados y mostrados.

La base de datos es básicamente una **estructura** que contiene a ciertos elementos (dominio) que están en ciertas relaciones entre ellos.

Además hay un **esquema de datos**: los nombres de las tablas y sus atributos, además de los tipos de éstos (string, número, etc).

Este provee información sobre los datos (metadatos).

Los **lenguajes de consulta** proveen una manera sencilla de decir qué información queremos obtener desde nuestra base de datos.

En el caso de las bases de datos relacionales se utiliza SQL:

- ▶ Tiene una sintaxis y semántica muy precisa.
- ▶ Es claro su poder expresivo (lógica de primer orden más agregación).
- ▶ Tiene una contraparte algebraica (álgebra relacional).
- ▶ Fácil de procesar (polinomial si asumimos la consulta fija).
- ▶ Muy estudiado en términos de optimización de consultas.

# Bases de datos relacionales: Restricciones de integridad

Las bases de datos proveen además la posibilidad de especificar **restricciones de integridad**: Ayudan a decidir cuáles instancias del modelo de datos son legales.

Por ejemplo, la llave primaria

**Name** → **Position, Salary**

es una restricción de integridad en nuestro ejemplo anterior.

Las RI ayudan a preservar la correspondencia entre la realidad exterior que está siendo modelada y la base de datos que es una abstracción de esa realidad.

- ▶ **Ayudan a capturar la semántica de los datos.**

Además las RI restringen las posibles actualizaciones de la base de datos. ¿Cómo?

Dado que las RI restringen las posibles instancias del modelo de datos y son información semántica acerca de los datos, quizá podríamos utilizarlas para responder consultas.

- ▶ Pueden ser útiles para optimizar la consulta.

**Ejemplo:** Asuma que en el ejemplo anterior tenemos una RI que dice:

- ▶ Ningún jefe puede ganar menos que 600000,

y que tenemos además la consulta:

- ▶ 

```
SELECT Name
FROM employee
WHERE Salary < 400000
```

Entonces, usando la RI sabemos que no necesitamos construir toda esta vista, sino tan solo:

- ▶ 

```
SELECT Name
FROM employee
WHERE Position ≠ jefe AND Salary < 400000
```

Hay otras suposiciones que están implícitas en una base de datos:

- ▶ **Mundo cerrado** Si una tupla de elementos no pertenece a una tabla  $T$ , quiere decir que tal hecho es falso.
- ▶ **Nombres únicos** Los elementos  $a$  y  $b$  de la base de datos son diferentes en el mundo real .

Aunque para nosotros este tipo de supuestos parecen obvios, deben ser explicitados al motor que responde automáticamente nuestras consultas a la base de datos.

Estas son suposiciones a meta-nivel (no pertenecen al esquema de la base de datos).

En definitiva:

- ▶ Las bases de datos relacionales son un pequeño pero elegante modelo de un sistema de representación del conocimiento.
- ▶ Como todo tal sistema hay funcionalidades que le faltan, y que podrían ser agregadas en la medida que los usuarios lo requieran.
- ▶ Además, es un sistema que es escalable y que ha tenido impacto comercial.
- ▶ Es, sin embargo, muy limitado, para otro tipo de aplicaciones (mover un robot, por ejemplo).

La **web semántica** intenta enriquecer con contenido semántico los documentos en la web.

Tal contenido semántico podría ser utilizado después, a la hora de intercambiar, consultar, o buscar algo en esos documentos.

Necesitamos, pues, de un lenguaje lo suficientemente expresivo como para especificar el contenido semántico e intercambiarlo en la web.

# Web semántica: Objetivos

En la web semántica hay dos objetivos fundamentales:

- ▶ Encontrar un lenguaje que pueda ser utilizado para transferir e intercambiar contenido semántico en la web;
- ▶ enriquecer los lenguajes actuales más utilizados para intercambiar información en la web (HTML o XML), de tal forma de que puedan también referirse a la información semántica.

Es decir, queremos un lenguaje que sea capaz de transferir tanto datos como meta-datos.

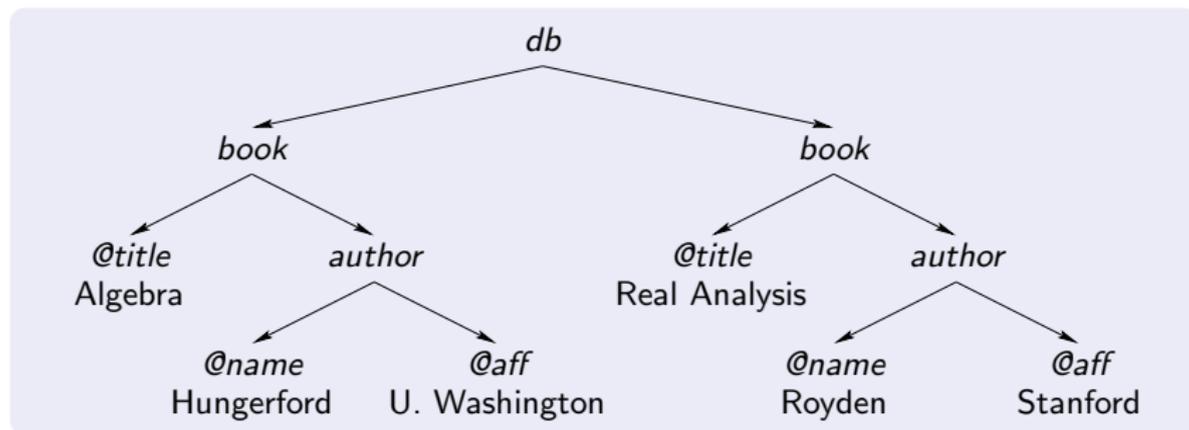
Vista parcial de la BD de una biblioteca como documento XML:

```
<db>
  <book title="Algebra">
    <author name="Hungerford" aff="U. Washington">
    </author>
  </book>
  <book title="Real Analysis">
    <author name="Royden" aff="Stanford">
    </author>
  </book>
</db>
```

Un documento XML contiene información *semiestructurada*, y por tanto, puede ser mostrado como un árbol:

# Web semántica: XML

Un documento XML contiene información *semiestructurada*, y por tanto, puede ser mostrado como un árbol:

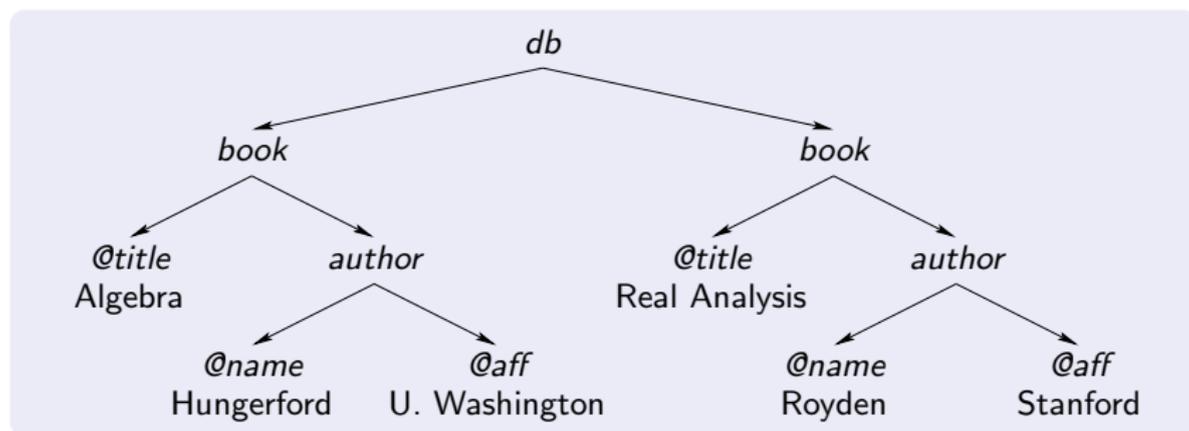


XML permite también especificar contenido semántico muy básico en la forma de **DTDs**.

- ▶ Definen el esquema del documento.
- ▶ Facilitan el intercambio de datos.
- ▶ Son fáciles de especificar (se usan gramáticas libres de contexto).

# Web semántica: XML y DTDs

Por ejemplo,



satisface el siguiente DTD:  $db \rightarrow book^+$  (siempre asumimos que existe un orden entre *hermanos*).

# Web semántica: Ventajas y desventajas de XML

Ventajas de XML:

- ▶ Fácil de entender, fácil de parsear.

Desventajas de XML:



RDF (**Resource Description Framework**) permite construir modelos de datos en términos de objetos (recursos web) y relaciones entre esos objetos.

Las bases de datos RDF son **bases de datos de grafos**, compuestas por la unión de triples de la forma:

*(Picasso, paints, Guernica).*

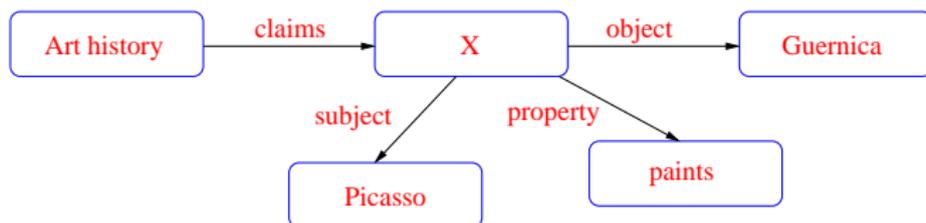
Aquí 'Picasso' es el **sujeto**, 'paints' es la **propiedad**, y 'Guernica' es el **objeto**.

# Web semántica: RDF y reificación

RDF presenta interesantes propiedades de **reificación**, i.e. un triple puede funcionar también como un objeto.

- ▶ Art history claims that Picasso paints Guernica.

Para representar esto debemos introducir un nuevo objeto X, que funciona como un valor nulo o indefinido.



Esto muestra que RDF provee un mayor poder semántico que XML.

# Web semántics: RDFS

RDFS es un vocabulario para describir propiedades y clases de recursos RDF.

Tiene nombres reservados para especificar clases y propiedades de alto nivel de estas clases como 'es subclase de', 'es una instancia de', adem'as de algunas restricciones de integridad.

Los triples pueden ser ahora puestos juntos en una base de datos de grafos que tiene estos links semánticos.

Cuando consultamos esta base de datos, consultamos también su información semántica.