Ontologías

Sistemas Inteligentes Distribuidos

Sergio Alvarez
Javier Vázquez

Bibliografía

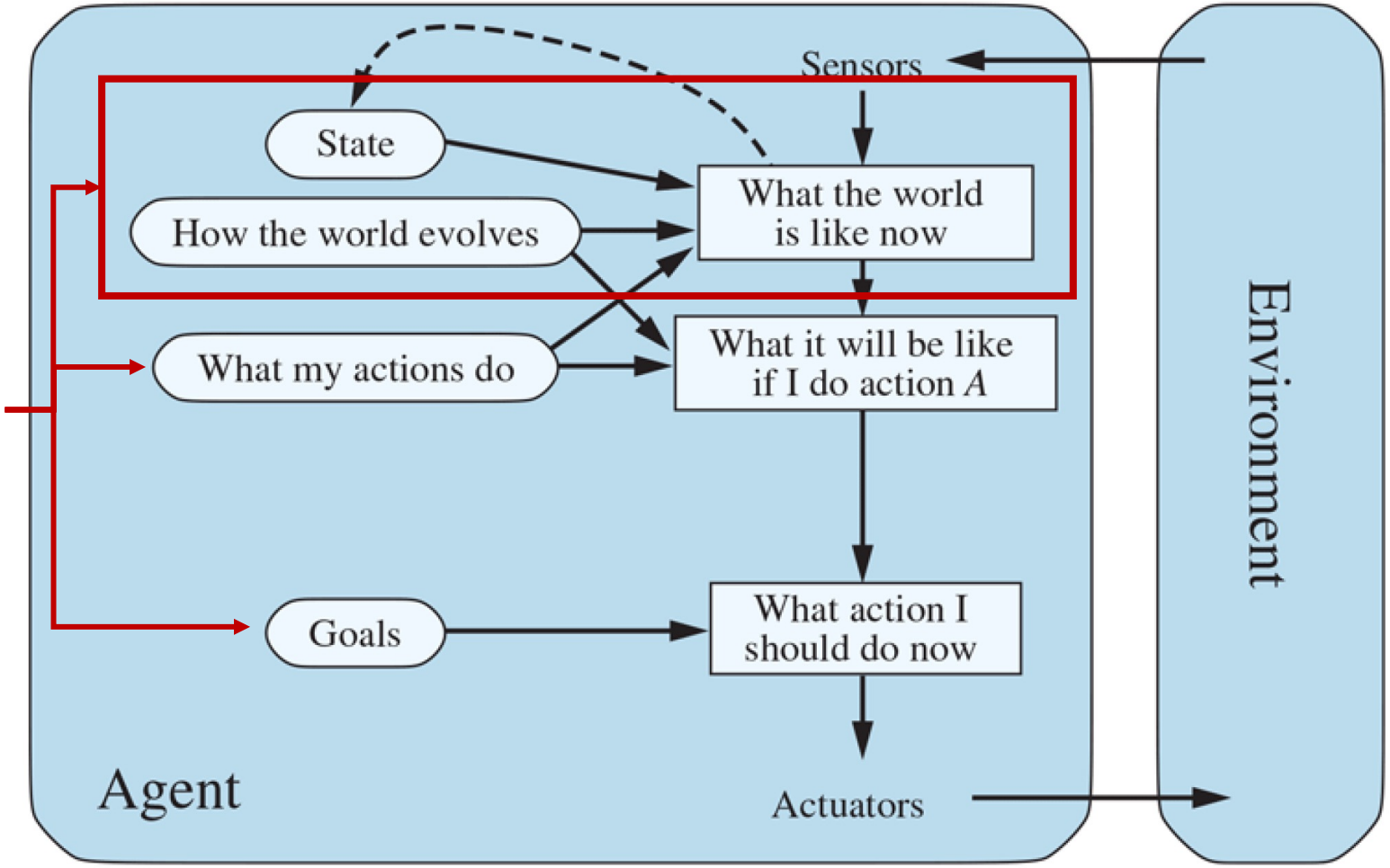
- *Artificial intelligence: a modern approach* (Russell & Norvig), cap. 8, 10
- *The Description Logic Handbook* (Baader, McGuinness, Nardi, Patel-Schneider)

Ontología y epistemología

Ontologías

Agente deliberativo por objetivos

Representaciones simbólicas



Representar el mundo

- En el tema anterior nos hemos centrado en la **representación de objetivos e intenciones**
- Otro aspecto fundamental es **cómo representar el mundo: *knowledge representation***
  - Cómo adquirir conocimiento a partir de los hechos (percepciones)
  - Cómo adquirir conocimiento a partir de conocimiento
  - Cómo hacer inferencias de manera eficiente
  - Cómo comunicar conocimiento

- Múltiples formalismos de representación del conocimiento
  - Redes neuronales (representation learning)
  - Redes bayesianas
  - Modelos markovianos
  - Redes jerárquicas (HTNs)
  - **Lógica de primer orden (FOL)**
- Nos centraremos en subconjuntos decidibles de FOL: **lógica descriptiva**
  - **Ontologías**
  - Laboratorio: RDF/OWL

¿Por qué una ontología?

- “**Sin lógica, una representación del conocimiento es incompleta**, sin un criterio para determinar si un hecho es redundante o contradictorio. **Sin una ontología, los términos y símbolos están pobremente definidos**, son ambiguos y confusos. Y sin modelos computacionales, la lógica y la ontología no se pueden implementar en programas. **La representación del conocimiento es la aplicación de la lógica y la ontología a la tarea de construir modelos computacionales para un cierto dominio**.”
  - John F. Sowa, *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, 2000.

¿Qué es una ontología?

- Merriam-Webster:
  - **Main Entry:** *on·tol·o·gy*
  - **Pronunciation:** *än-'tä-l&-jE*
  - **Function:** *noun*
  - **Etymology:** *New Latin ontologia, from ont- + -logia*
  - **Date:** *circa 1721*

  1. a branch of metaphysics concerned with **the nature and relations of being**;
  2. **a particular theory** about the nature of being or the kinds of existents.

- RAE:

  1. *f. Fil.* Parte de la metafísica que trata del **ser en general y de sus propiedades trascendentales**.
  2. *f.* En ciencias de la comunicación y en inteligencia artificial, **red o sistema de datos que define las relaciones existentes** entre los conceptos de un dominio o área del conocimiento.

Origen

- El término es relativamente moderno (siglo XIX)
  - Del griego *ontos* (del ser) y *logos* (palabra)
- Sus primeros usos fueron dirigidos a diferenciar el estudio de las categorías del ser de las categorías de la Biología
- De hecho, la categorización es una tarea común de diversas áreas del conocimiento
  - Filosofía, biología, medicina, lingüística, …

Origen

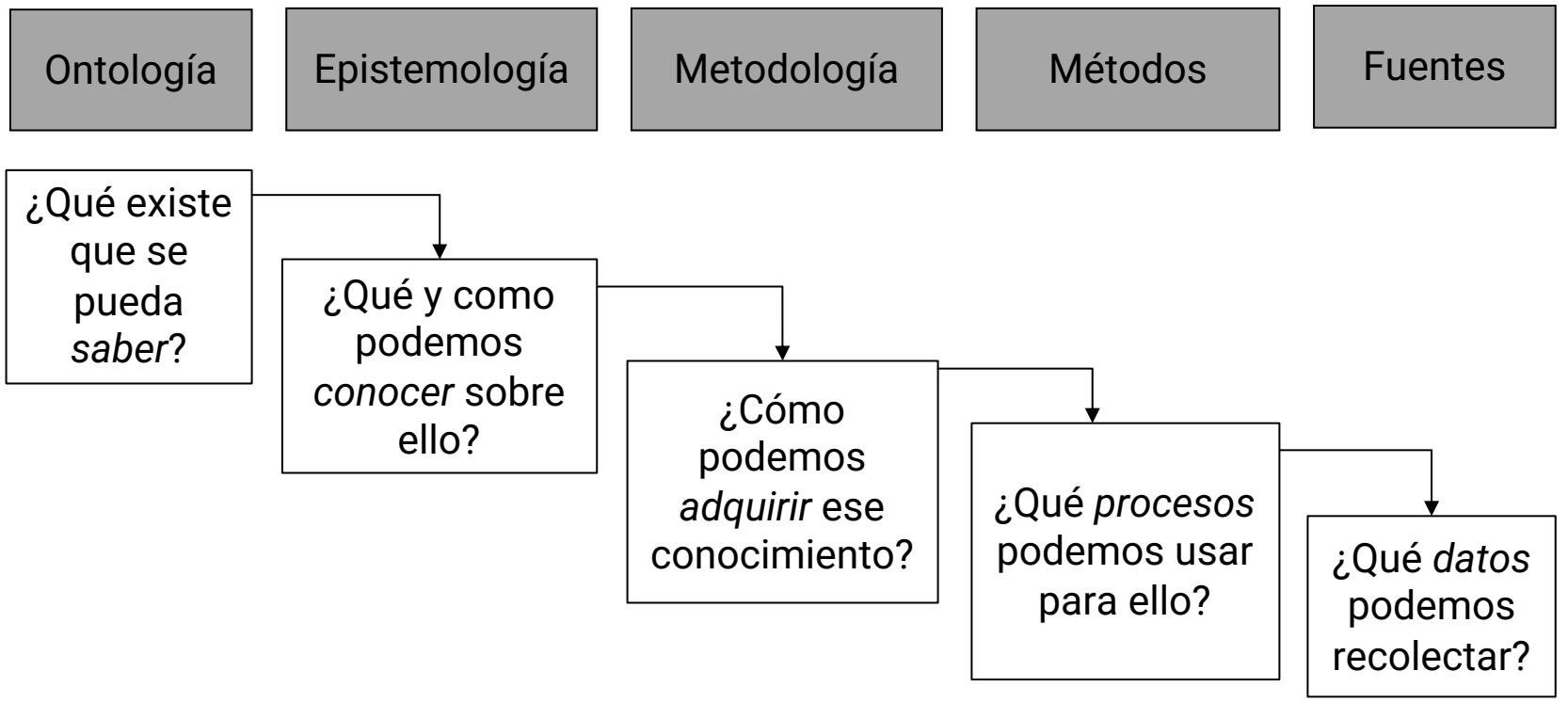
- El desarrollo de ontologías está íntimamente relacionado con la Filosofía
- Aristóteles acuño el término *Categoría* para describir las diferentes clases de cosas en que se puede dividir el mundo
- El primer sistema de clasificación de Aristóteles se dedica a los seres (τὰ ὄντα), dividiendo en dos conceptos: *cosas predicadas sobre otras cosas* (said-of) y *cosas halladas en otras cosas* (present-in)

No predicado sobre / No presente en Substancias primarias Particulares no accidentales e.g. una persona específica (Sócrates)	**No predicado sobre / Presente en** No sustanciales Particulares accidentales e.g. el color de la piel de Sócrates
Predicado sobre / No presente en Substancias secundarias Universales no accidentales e.g. concepto de persona	**Predicado sobre / Presente en** Sustancias primarias Universales accidentales e.g. conocimiento

Ontología y Epistemología

- Ontología: cómo es la realidad, qué elementos básicos contiene
- Epistemología: estudio del criterio por el cual podemos saber qué constituye conocimiento justificado o científico
- La manera como pensamos que el mundo es (ontología) influencia:
  - lo que pensamos que puede conocerse sobre él (epistemología)
  - cómo pensamos que se puede investigar sobre él (metodología)
  - los tipos de teorías que se pueden construir sobre él
- *If it looks like a duck, swims like a duck, and quacks like a duck, then it probably is a duck*

Ontología y Epistemología



Hay C. (2002), *Political Analysis. A Critical Introduction*

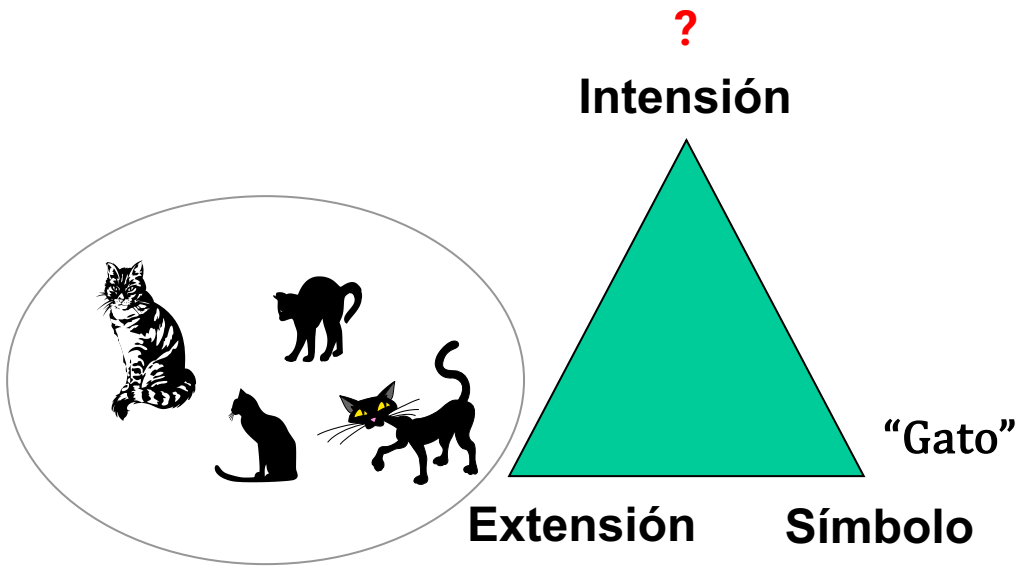
Ontología / ontología

- La Ontología tiene como objetivo el **estudio de las categorías** que existen en un determinado dominio
- El resultado de este estudio es una ontología
  - **Un catálogo de los diferentes tipos de objetos que se asume que existen en un dominio D, desde la perspectiva de alguien que usa el lenguaje L para hablar sobre D**
- Una ontología se puede ver como un **vocabulario** que los agentes necesitan para **intercambiar información** en el contexto de un dominio

Definiciones de ontología

- Una ontología define **los términos y relaciones básicos que abarcan el vocabulario** del área de un determinado tema, así como las **reglas para combinar términos y relaciones** para obtener extensions del vocabulario
  - Neches, R; Fikes, R; et al *Enabling Technology for Knowledge Sharing* AI Magazine. Winter 1991. pp36-56
- Una ontología es la **especificación explícita de una conceptualización**
  - Gruber, T. *A translation approach to portable ontology specifications* Knowledge Acquisition. Vol. 5. 1993. 199-220.
- Una ontología es una descripción (e.g. la **especificación formal de un programa**) de los conceptos y relaciones que pueden existir **para un agente o para una comunidad de agentes**
  - Gruber, T. *What is an Ontology?*

“Triángulo semiótico"

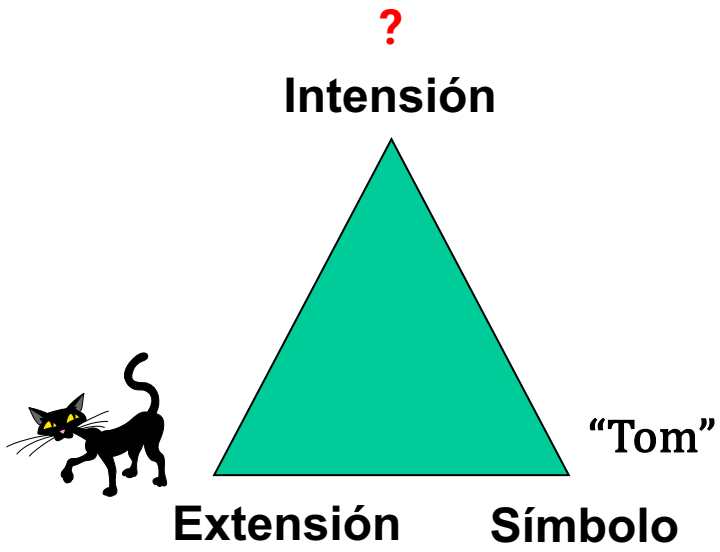


Símbolo: termino lingüístico usado para hacer referencia al concepto

Intensión (referencia, concepto o connotación): conjunto de atributos, características y propiedades que constituyen el significado del concepto

Extensión (referente o denotación): conjunto de casos particulares que están cubiertos por el concepto

“Triángulo semiótico"

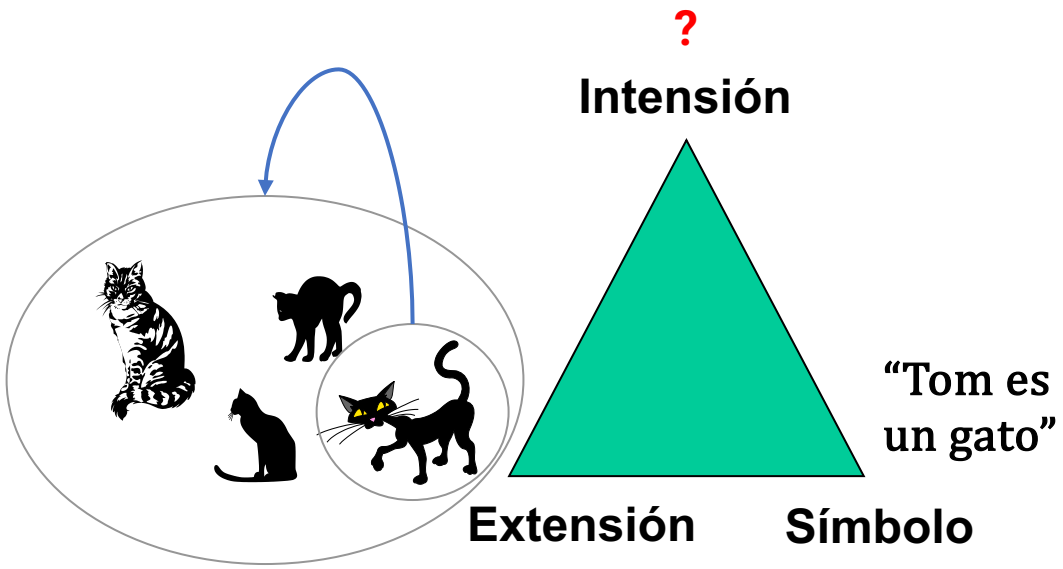


Símbolo: termino lingüístico usado para hacer referencia al concepto

Intensión (referencia, concepto o connotación): conjunto de atributos, características y propiedades que constituyen el significado del concepto

Extensión (referente o denotación): conjunto de casos particulares que están cubiertos por el concepto

“Triángulo semiótico"



Símbolo: termino lingüístico usado para hacer referencia al concepto

Intensión (referencia, concepto o connotación): conjunto de atributos, características y propiedades que constituyen el significado del concepto

Extensión (referente o denotación): conjunto de casos particulares que están cubiertos por el concepto

Elementos de una ontología

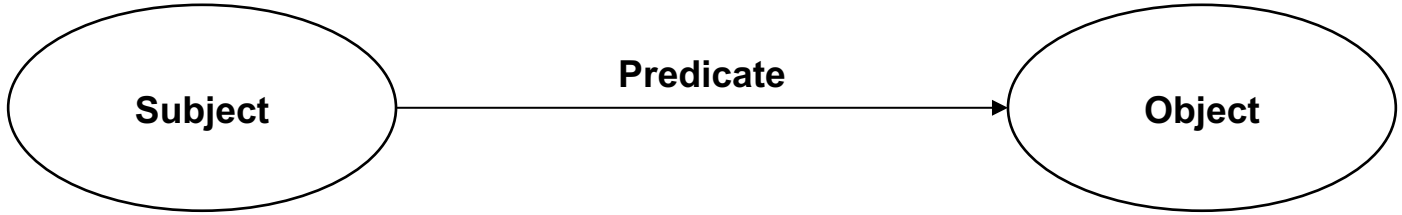
- **Conceptos:** permiten modelar elementos como objetos, tareas, funciones, acciones, estrategias, planes, etc.
  - Algunos lenguajes de ontología se refieren a ellos como **clases,** compuestas de **propiedades**
- **Relaciones:** modelan un tipo de interacción entre conceptos del dominio
- **Funciones:** un tipo especial de relación
- **Restricciones:** reglas que pueden restringir el dominio de los valores para algunos conceptos y relaciones
- **Instancias:** constituyen los individuos concretos representados por los conceptos de la ontología
- **Axiomas:** proposiciones y reglas, sobre todos los demás elementos, que se consideran ciertas

OWL

Ontologías

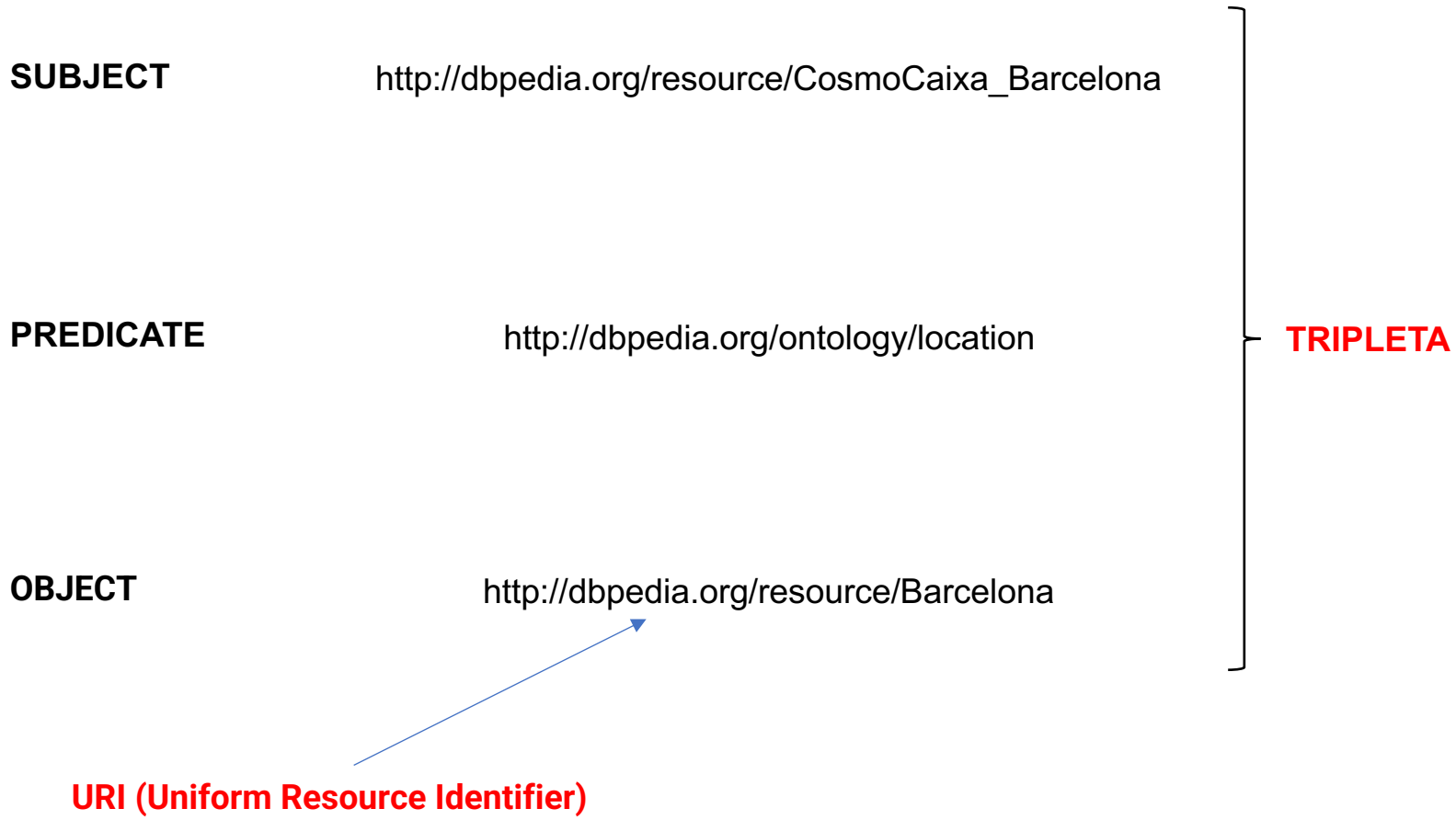
RDF

- Las ontologías y grafos de conocimiento (*knowledge graphs*) se suelen representar como **tripletas** (*triples*) **<sujeto, predicado, objeto>**

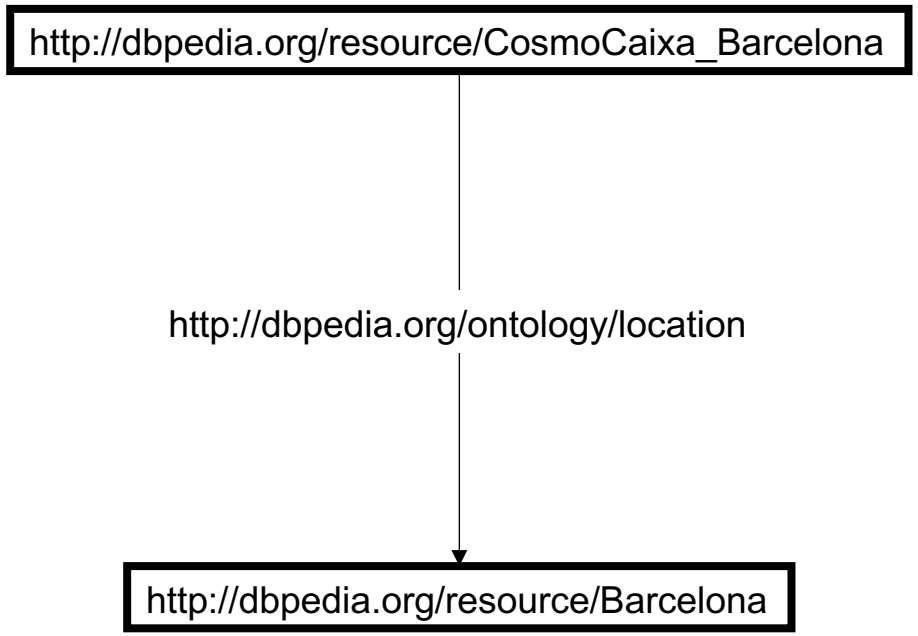


- El formato estándar más común es **RDF** (Resource Description Framework)
- En RDF, cada uno de los tres elementos es una URI, e.g.
  - **S: <http://dbpedia.org/resource/Tetris>**
  - **P: <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>**
  - **O: <http://dbpedia.org/ontology/VideoGame>**

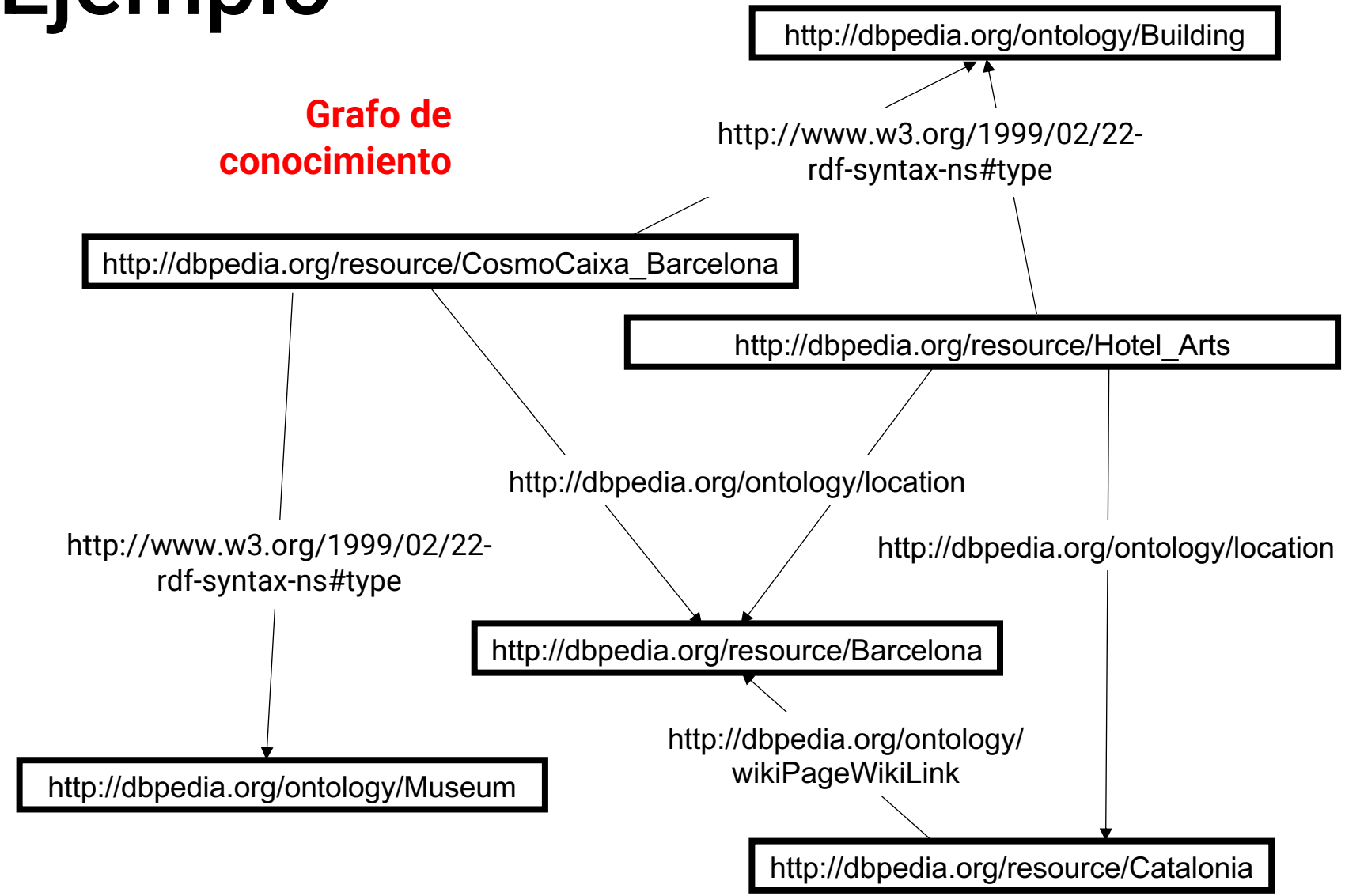
RDF: Ejemplo


SUBJECT
http://dbpedia.org/resource/CosmoCaixa_Barcelona
PREDICATE
http://dbpedia.org/ontology/location
OBJECT
http://dbpedia.org/resource/Barcelona
TRIPLETA
URI (Uniform Resource Identifier)

RDF: Ejemplo


http://dbpedia.org/resource/CosmoCaixa_Barcelona
http://dbpedia.org/ontology/location
http://dbpedia.org/resource/Barcelona

RDF: Ejemplo

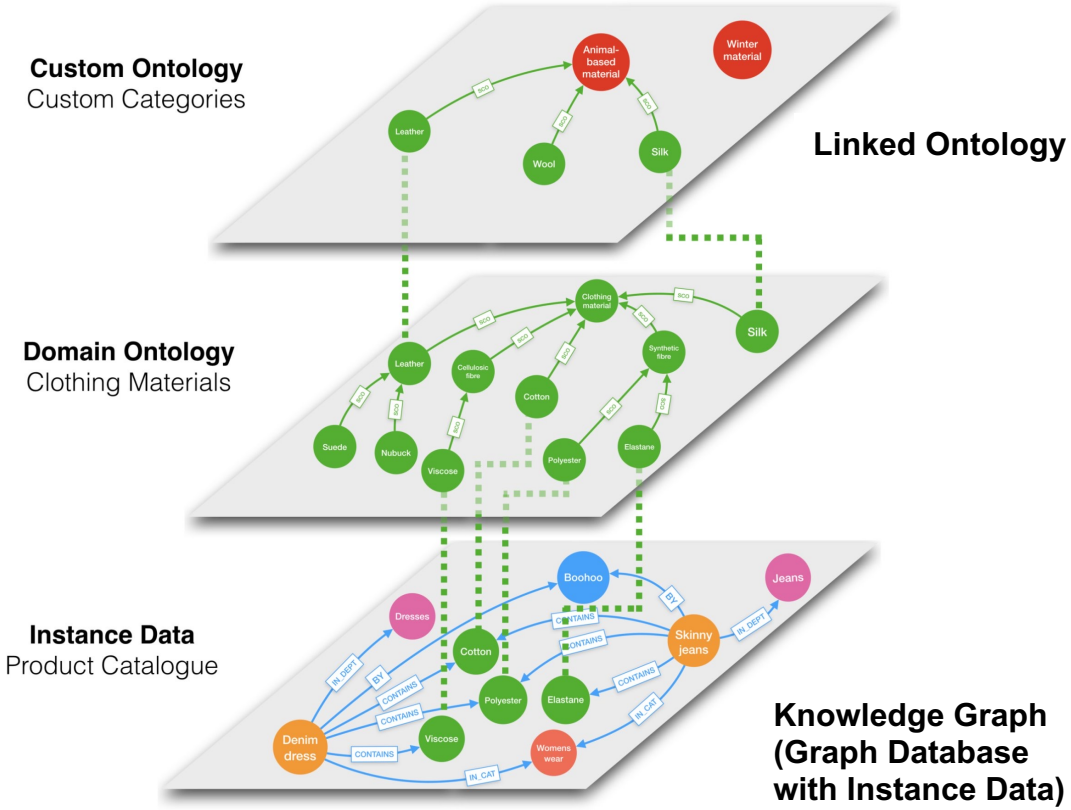

Grafo de conocimiento
http://dbpedia.org/ontology/Building
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://dbpedia.org/resource/CosmoCaixa_Barcelona
http://dbpedia.org/resource/Hotel_Arts
http://dbpedia.org/ontology/location
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://dbpedia.org/ontology/location
http://dbpedia.org/resource/Barcelona
http://dbpedia.org/ontology/wikiPageWikiLink
http://dbpedia.org/ontology/Museum
http://dbpedia.org/resource/Catalonia


https://lod-cloud.net

Grafos de conocimiento

Un grafo de conocimiento (o red semántica) representa una red de entidades del mundo (como objetos, eventos, situationes o conceptos) e ilustra las relaciones entre ellos

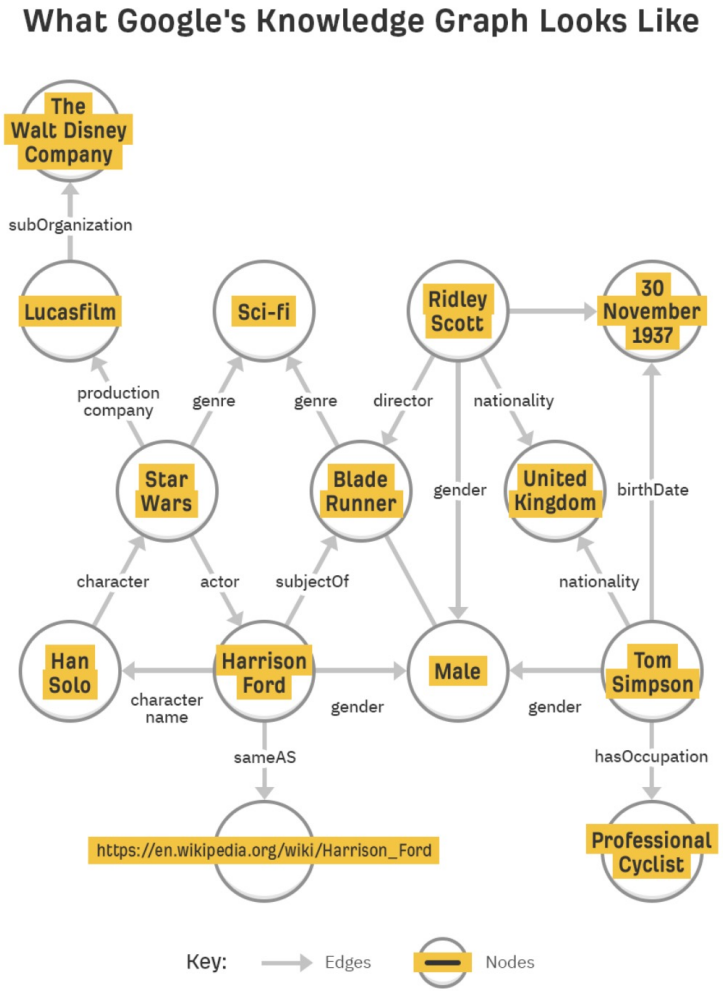
Esta información se guarda generalmente en bases de datos de grafos y se visualizan como una estructura en forma de grafo

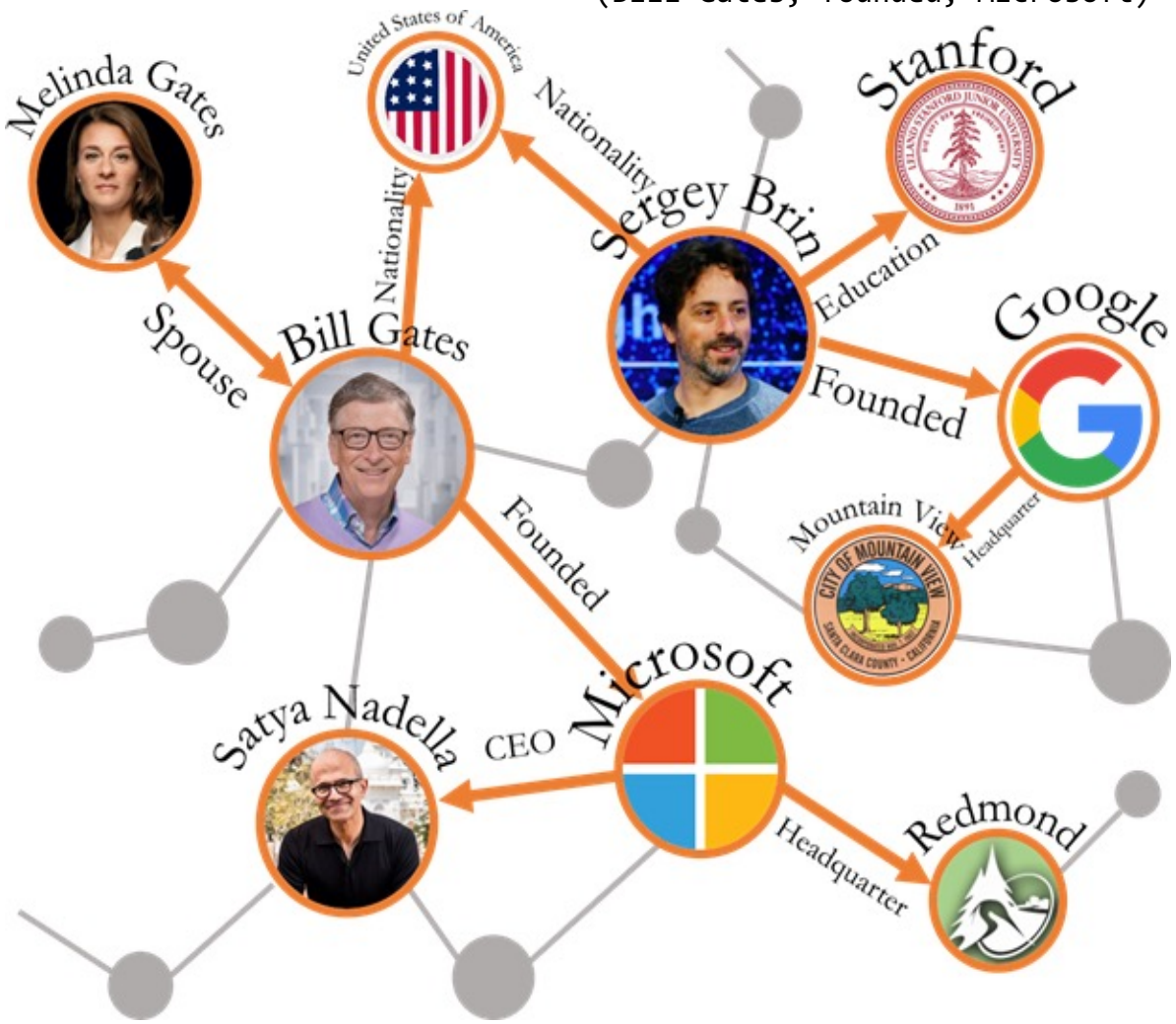


[Jesús Barrasa (2019) *"The fashion Knowledge Graph. Inferencing with Ontologies in Neo4j"*]

Grafos de conocimiento

“Bill Gates fundó Microsoft” se representa como
`(Bill Gates, founded, Microsoft)`





{salvarez,jvazquez}@cs.upc.edu

RDF Schema

- **Classes**
  - rdfs:Resource
  - rdfs:Class
  - rdfs:Literal
  - rdfs:Datatype
  - rdf:langString
  - rdf:HTML
  - rdf:XMLLiteral
  - rdf:Property
- **Properties**
  - rdfs:range
  - rdfs:domain
  - rdf:type
  - rdfs:subClassOf
  - rdfs:subPropertyOf
  - rdfs:label
  - rdfs:comment

https://www.w3.org/TR/rdf-schema

RDF Schema

- La especificación es demasiado débil para poder describir recursos con suficiente detalle, **NO permite**:
  - **Restricciones con rangos y dominios condicionales**
    - E.g. no permite decir que el rango de *utilizaEnergía* es *Eléctrica* cuando se aplica a *VehículoElectrico* y que su rango es *Gasolina* cuando se aplica a *VehículoDeCombustión*
  - **Combinar clases por unión, intersección ni complemento**
    - E.g. no permite formalizar que *utilizaEnergía* se puede aplicar tanto a vehículos como a seres vivos
  - **Restricciones de existencia o cardinalidad**
    - E.g. no permite formalizar que las bicicletas tienen dos ruedas
  - **Propiedades transitivas, inversas o simétricas**
    - E.g. no permite modelar que la relación *esAncestroDe* es transitiva
- **Tampoco permite realizar razonamiento automático**
  - No hay ningún razonador nativo disponible para RDFS

La utilidad de un razonador

Con acceso a un razonador, podemos

- **Asegurarnos** de que una base de conocimiento es
  - Coherente: todas las clases pueden tener instancias
  - Correcta: las suposiciones sobre el mundo se trasladan correctamente a las inferencias
  - Mínimamente redundante: existe una ambigüedad manejable
  - … y, en general, **cualquier propiedad deseable** que podamos convertir en fórmulas lógicas
- **Responder** a preguntas sobre clases e instancias, por ejemplo:
  - Encontrar nuevas clases por inferencia
  - Obtener instancias que se correspondan con un patrón determinado
  - …

Lenguajes de ontología

- **DAML**
  - Fuertemente asociado a WebServices
  - DARPA
- **OIL**
  - Comisión Europea
  - Primer lenguaje de intercambio de Ontologías
  - Anticuado y poco expresivo
- **RDF**
  - Resource description framework
  - Esquema general de representación de información
  - Sin semántica asociada
- **OWL**
  - Inspirado en DAML y OIL
  - Construido sobre RDF

Ontology Web Language

- Motivación principal:
  - Sintáctica bien definida
  - Semántica formal
  - Posibilidad de razonamiento eficiente
- Recomendación W3C
  - OWL (2004)
  - OWL2 (2012)
    - https://www.w3.org/TR/owl2-overview/
    - https://www.w3.org/TR/2012/REC-owl2-quick-reference-20121211/
- El fundamento lógico de OWL es la **lógica descriptiva (DL)**

Lógica descriptiva

Ontologías

Lógica de primer orden

- La lógica proposicional es un lenguaje declarativo con una semántica clara
  - Con expresividad insuficiente para modelar conjuntos de entidades
  - Y por lo tanto insuficiente para expresar clases, propiedades, relaciones o axiomas
- La lógica de primer orden sí permite expresar todos los elementos identificados como importantes para una ontología:
  - Instancias (constantes o literales), conceptos (predicados unarios), relaciones (predicados binarios)
- Sin embargo, la lógica de primer orden **NO es decidible**
  - Intuición de la demostración: cualquier máquina de Turing es reducible a FOL

Lógica descriptiva

- **Familia** de lenguajes de representación del conocimiento
  - Elementos principales: conceptos, roles, instancias
  - Semántica formal
  - Fragmentos **decidibles** de la lógica de primer orden (FOL)
- Elementos básicos:
  - **Individuos** (instancias)
  - **Conceptos** (clases)
  - **Roles** (propiedades, relaciones)
- Es posible crear razonadores para cada lenguaje de la familia

- Los dialectos reciben un nombre basado en ciertas extensiones a añadir al lenguaje base
- El lenguaje base se llama $\mathcal{AL}$ (Attributive Language)
  - negación atómica, pero no en la LHS de axiomas de inclusión o igualdad
  - intersección de conceptos
  - restricciones universales
  - restricciones existenciales, pero sólo para el concepto universal

Algunas extensiones

- $\mathcal{F}$ propiedades funcionales
- $\mathcal{E}$ restricciones existenciales cualificadas
- $\mathcal{C}$ negación de concepto
- $\mathcal{U}$ unión de concepto
- $\mathcal{H}$ jerarquía de roles
- $\mathcal{R}$ axiomas sobre inclusión de roles, reflexividad, irreflexividad, roles disjuntos
- $\mathcal{O}$ uso de individuos en la descripción de conceptos (nominales)
- $\mathcal{I}$ propiedades inversas
- $\mathcal{N}$ restricciones de cardinalidad (no cualificadas)
- $\mathcal{Q}$ restricciones de cardinalidad cualificadas
- $(\mathcal{D})$ uso de tipos de datos, valores de datos y propiedades sobre tipos de datos

Lenguajes DL populares

- $\mathcal{ALC}$ (*Attributive Language with Complement*):

$$C, D \rightarrow A \mid \top \mid \bot \mid \neg C \mid C \sqcap D \mid \exists R.C \mid \forall R.C$$

  donde A es un concepto atómico, C y D son conceptos y R es un rol

- $\mathcal{S}$ es $\mathcal{ALC}$ con roles transitivos ($\mathcal{R}_+$)
- $\mathcal{SHOIN}(\mathcal{D})$ es el formalismo correspondiente a OWL-DL (OWL 1)
- $\mathcal{SROIQ}(\mathcal{D})$ es el formalismo correspondiente a OWL2
  - Razonadores: HermiT, Pellet

Construcciones básicas

- **Conjunto** de individuos que no son parte de un concepto
  - $\neg \text{Estudiante}$ (todos los individuos que no son estudiantes)
- Intersección de dos conceptos
  - $\text{Estudiante} \sqcap \text{Profesor}$ (todos los individuos que estudian e imparten clases a la vez)
- Unión de dos conceptos (no soportado en muchos lenguajes DL)
  - $\text{Estudiante} \sqcup \text{Profesor}$ (todos los individuos que o bien estudian o bien imparten clases, o ambas cosas a la vez)
- Restricción universal
  - $\forall \text{asiste}.\, \text{Clase}$ (todos los individuos que, siempre que asisten a algo, ese algo es a alguna clase)
- Restricción existencial
  - $\exists \text{imparte}.\, \text{Clase}$ (todos los individuos que como mínimo imparten alguna clase)

Axiomas

- Inclusión
  - $\text{EstudianteUniversitario} \sqsubseteq \text{Estudiante}$
- Equivalencia
  - $\text{Humano} \sqsubseteq \text{Persona} \sqcap \text{Persona} \sqsubseteq \text{Humano} \Leftrightarrow \text{Humano} \equiv \text{Persona}$
  - $\text{Madre} \equiv \text{Progenitor} \sqcap \text{Mujer}$
  - $\text{Estudiante} \equiv \text{Persona} \sqcap \exists \text{atiende}.\,\text{Clase}$
- En $\mathcal{ALC}$ no hay axiomas sobre roles pero en $\mathcal{SROIQ}(\mathcal{D})$ sí, e.g.
  - $\text{esHijoDe} \equiv \text{es ProgenitorDe}^{-1}$
  - Por la extensión $\mathcal{I}$ (propiedades inversas)

Semántica

- La semántica se define a partir de **interpretaciones**
- Una interpretación I es un par $(\Delta^I, .^I)$, donde

  Δ^I es el dominio (un conjunto no vacío de individuos)

  $.^I$ es una función de interpretación que mapea:

  Cada concepto a un subconjunto de Δ^I

  Cada rol a un subconjunto de $\Delta^I \times \Delta^I$

  Cada individuo a un elemento de Δ^I

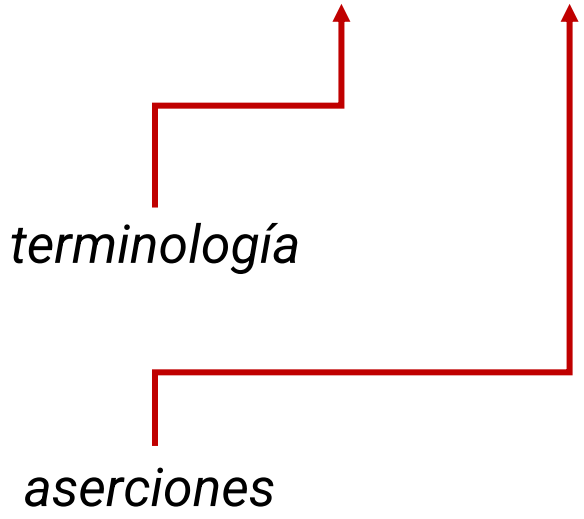
Semántica

Description	Syntax	Semantics
Top (everything)	$\top$	Δ^I
Bottom (empty concept)	$\bot$	$\emptyset$
Atomic concept ($\in N_C$)	A	$A^I \subseteq \Delta^I$
Atomic role ($\in N_R$)	r	$r^I \subseteq \Delta^I \times \Delta^I$
Concept conjunction	$C \sqcap D$	$C^I \cap D^I$
Concept disjunction	$C \sqcup D$	$C^I \cup D^I$
Concept negation (complement)	$\neg C$	$\Delta^I \backslash C^I$
Universal restriction	$\forall r.C$	$\{x \mid x \in \Delta^I \wedge \forall y(r(x,y) \in r^I \rightarrow y \in C^I)\}$
Existential restriction	$\exists r.C$	$\{x \mid x \in \Delta^I \wedge \exists y(r(x,y) \in r^I \wedge y \in C^I)\}$
General Concept Inclusion	$C \sqsubseteq D$	$C^I \subseteq D^I$
Concept Equivalence	$C \equiv D$	$C^I = D^I$

Tabla extraída de Christophe Debruyne, *Description Logics (Lecture)*

TBox, ABox

Base de Conocimiento = TBox + ABox



KB
TBox
$\text{Estudiante} \equiv \text{Persona} \sqcup \exists \text{atiende}.\,\text{Clase}$
ABox
$\text{Persona}(\text{Victor})$ $\text{atiende}(\text{Victor}, \text{SID})$

TBox: la terminología

- Un TBox es un conjunto finito formado por inclusiones ($\sqsubseteq$) y equivalencias ($\equiv$) de conceptos
- Una **definición** es una **equivalencia** entre conceptos donde, en la parte izquierda, el concepto es atómico: ayudan a darle un nombre simbólico a descripciones complejas
  - $\text{PizzaCuatroEstaciones} \equiv \text{Pizza} \sqcap \exists\text{tieneIngrediente}.\text{Alcachofa} \sqcap \exists\text{tieneIngrediente}.\text{Albahaca} \sqcap \exists\text{tieneIngrediente}.\text{Jamon} \sqcap \exists\text{tieneIngrediente}.\text{Seta} \sqcap \exists\text{tieneIngrediente}.\text{Oliva}$
  - $\text{PizzaVegana} \equiv \text{Pizza} \sqcap \forall\text{tieneIngrediente}.\text{IngredienteVegano}$
  - $\text{PizzaNoVegana} \equiv \text{Pizza} \sqcap \exists\text{tieneIngrediente}.\neg\text{IngredienteVegano}$
  - Las instancias de $\text{PizzaCuatroEstaciones}$ se podrían clasificar también como PizzaNoVegana por inferencia (*dependerá del nivel del razonador*)
- Si no podemos definir un concepto de manera completa, podemos usar la **inclusión** para modelar por lo menos sus **condiciones necesarias** (especialización):
  - $\text{PizzaHawaiana} \sqsubseteq \exists\text{tieneIngrediente}.\text{Piña}$ (sabemos seguro que lleva piña, pero… ¿es pizza?)

ABox: las aserciones

- Un ABox es un conjunto $\mathcal{A}$ de aserciones que describe los hechos específicos del dominio
  - $\mathrm{PizzaCuatroEstaciones}(\mathrm{pizza342546})$
  - $\mathrm{Pizza}(\mathrm{pizza123})$
  - $\mathrm{Rucula}(\mathrm{rucula456})$
  - $\mathrm{tieneIngrediente}(\mathrm{pizza123}, \mathrm{rucula456})$
- Dada una interpretación $I = (\Delta^I, .^I)$
  - I satisface $C(a)$ sii $a^I \in C^I$
  - I satisface $r(a, b)$ sii $(a^I, b^I) \in r^I$
- Un ABox $\mathcal{A}$ se satisface con respecto a una interpretación I sii cada aserción en $\mathcal{A}$ se satisface con respecto a I
  - O lo que es lo mismo: $I \vDash \mathcal{A}$
- *e.g.* $\mathrm{PizzaVegana}(\mathrm{pizza123})$, al no ser una aserción, será satisfactible o no… dependiendo de I

ABox: las aserciones

ABox
$\text{Gato}(\text{Tom})$ $\text{Gato}(\text{Garfield})$ $\text{tieneMascota}(\text{Victor}, \text{Tom})$

- Hay dos premisas comunes a todas las especificaciones de OWL sobre el ABox que influyen en la interpretación I:
  - **NO hay *suposición de nombre único*** (*Unique Name Assumption*, **UNA**): cada nombre de individuo en I puede no ser único, ~~$a^I \neq b^I$~~
  - **Sí hay *suposición de mundo abierto*** (*Open World Assumption*, **OWA**): la información en el ABox es incompleta
    - Las aserciones en el ABox son ciertas en todos los modelos basados en él
    - Pero **no todo lo que es desconocido** en los modelos basados en el ABox **es necesariamente falso**
- Por ejemplo, si tenemos el ABox de arriba, ¿cuál es la respuesta a la pregunta "*¿Tiene Victor a Garfield como mascota?*" en OWL?
  - Primero, al no haber suposición de nombre único, **puede ser que Tom = Garfield**
  - Segundo, al ser mundo abierto, **no sabemos seguro si** $\text{tieneMascota}(\text{Victor}, \text{Tom})$ **o no**
  - Por lo tanto: **no lo sabemos**

ABox: las aserciones

La interpretación I también depende, en gran medida, del lenguaje DL específico, de las extensiones que usa y por lo tanto los axiomas que define

KB
TBox
PizzaCuatroEstaciones ≡ Pizza ⊓ ∃tieneIngrediente. Alcachofa ⊓ ∃tieneIngrediente. Albahaca ⊓ ∃tieneIngrediente. Jamon ⊓ ∃tieneIngrediente. Seta ⊓ ∃tieneIngrediente. Oliva PizzaNoVegana ≡ Pizza ⊓ ∃tieneIngrediente. ¬IngredienteVegano
ABox
PizzaCuatroEstaciones(pizza342546)

OWL 2 usa el lenguaje $\mathcal{SROIQ}(\mathcal{D})$

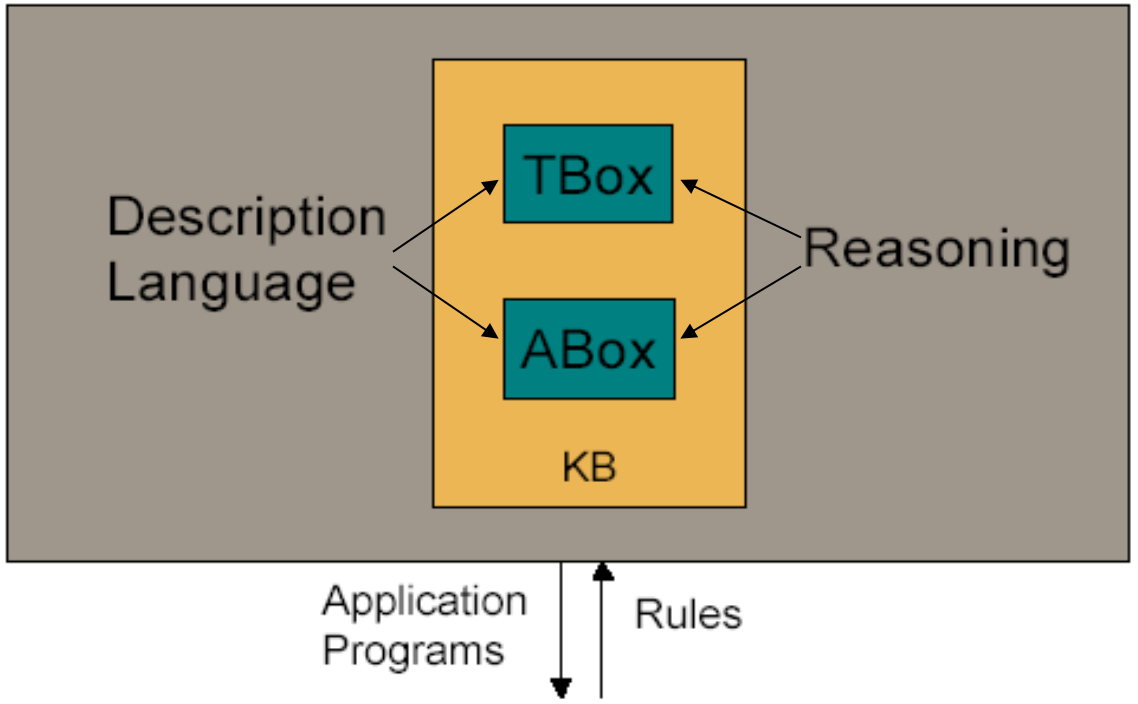
En $\mathcal{SROIQ}(\mathcal{D})$ se incluye $\mathcal{ALC}$ y por tanto negación y existenciales sobre conceptos:

$I \vDash PizzaNoVegana(\text{pizza342546})$

Si, en cambio, estuviéramos en el lenguaje $\mathcal{AL}$ y por tanto sin existenciales sobre conceptos ni negación de concepto:

$I \nvDash PizzaNoVegana(\text{pizza342546})$

Arquitectura de un sistema de DL



Ontologías en la práctica

Ontologías

¿Por qué usar ontologías?

- Para compartir una compresión mutua sobre la estructura de la información entre diferentes personas y/o agentes
- Para permitir la reutilización del conocimiento de un dominio
- Para hacer explícitas las suposiciones sobre el dominio que se establecen al desarrollar una aplicación
- Para separar el conocimiento del dominio de su operacionalización
- Para permitir el análisis (semi-)automático del conocimiento

Ejemplos de ontologías

- Tráfico y movilidad
  - DATEX II (https://datex2.eu/)
- Medicina
  - Gene Ontology (https://www.geneontology.org/)
  - Semantic DICOM (https://bioportal.bioontology.org/ontologies/SEDI)
  - Disease Ontology (http://disease-ontology.org/)
  - HL7 RDF (http://build.fhir.org/rdf)
- Música
  - The Music Ontology (http://musicontology.com/)
- Información enciclopédica
  - DBPedia (https://wiki.dbpedia.org/)
- Metadatos, en general
  - Dublin Core (http://dublincore.org/)

Repositorios de ontologías públicas

- https://lod-cloud.net/
  - Proyecto de “linked data”, con ontologías enlazadas por relaciones de equivalencias (owl:sameAs) o conceptos comunes, 1000+ grafos de conocimiento en RDF
- http://www.daml.org/ontologies/
  - 200+ ontologías en DAML+OIL/OWL
  - Temas muy diversos (departamentos académicos, cine, información geográfica, aeropuertos, bibliografía, biología, química, moda, meteorología, …)
- https://protegewiki.stanford.edu/index.php?title=Protege_Ontology_Library&oldid=13780
  - Lista de ontologías interesantes (principalmente en OWL)
- https://bioportal.bioontology.org/
  - 1000+ ontologías relacionadas con la medicina y la farmacología

Algunas herramientas

- Protégé (editor)
  - Sistema de plugins: OKBC, RDF(S), OWL, UML, Prolog, Jess, OWL-S, etc.
  - http://protege.stanford.edu/, http://protege.stanford.edu/plugins/owl/
- Owlready2
  - API en Python para manipular RDF, RDF(S) y OWL
  - https://github.com/pwin/owlready2
- Jena
  - API Java para manipular RDF, RDF(S) y OWL
  - http://www.hpl.hp.com/semweb/jena.htm, http://jena.sourceforge.net/
- Pellet
  - Razonador open-source basado en Java para OWL-DL y OWL2
  - https://github.com/stardog-union/pellet
- RDF Validator
  - Permite probar grafos RDF online: http://www.w3.org/RDF/Validator/

Algunas herramientas

- TopBraid Composer
  - TopBraid Composer es un editor commercial de ontologías y una plataforma de desarrollo de aplicaciones para web semántica
  - Suporta OWL, RDFS y el lenguaje de reglas SWRL, y proporciona funcionalidades para el desarrollo de ontologías, como visualización, depuración y prueba
- PoolParty
  - PoolParty es una suite comercial que incluye un editor de ontologías, un gestor de taxonomías y un sistema de gestión de tesauros
  - Soporta OWL, RDFS y SKOS (*Simple Knowledge Organization System*) y proporciona funcionalidades para el desarrollo de ontologías, como clasificación automática y búsqueda semántica
- WebProtege
  - WebProtege es un editor gratuito de ontologías y un sistema de gestión de conocimiento. Está basado en Protégé y proporciona una interfaz usable para crear y editar ontologías
  - Soporta OWL y RDFS y tiene funcionalidades para edición colaborativa de ontologías