Redes semánticas

Redes semánticas: contenidos

- Introducción
- Definición de redes semánticas (o asociativas)
  - Características
  - Tipos de arcos
- Mecanismos de inferencia (o razonamiento)
  - Herencia de propiedades
  - Búsqueda de la intersección entre dos conceptos
  - Contestar preguntas / recuperar información
- Representación con redes semánticas
  - Representación de relaciones no binarias
  - Representación de sucesos
- Conclusiones

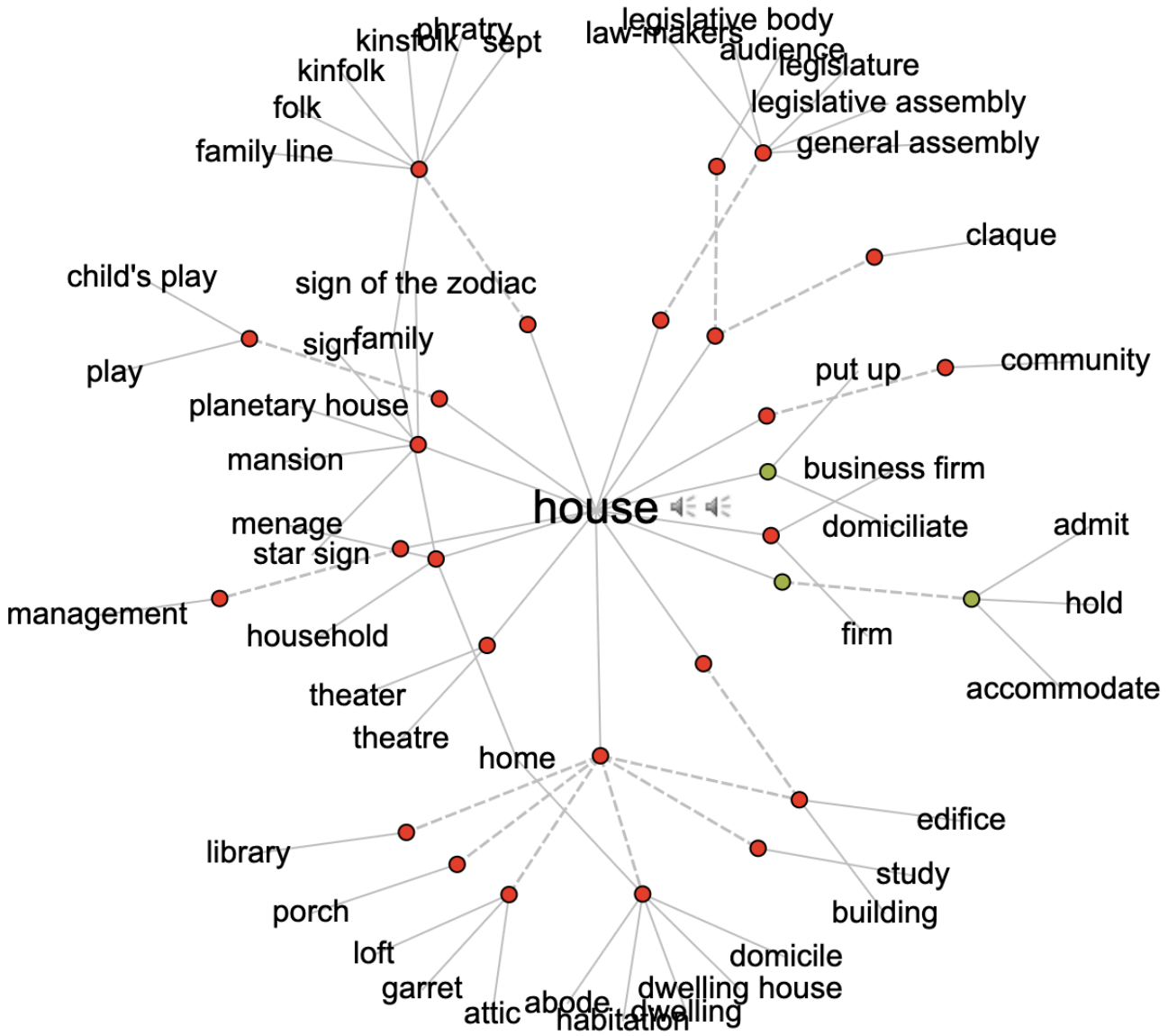
Introducción

- Ya conocemos la representación en lógica de primer orden
  - Principios del razonamiento lógico: correcto y completo
  - ¿Cómo representar y razonar con muchos conceptos y relaciones?
    - Ej: conocimiento de sentido común con 1,4 millones (ConceptNet)
    - Más datos → menos semántica formal

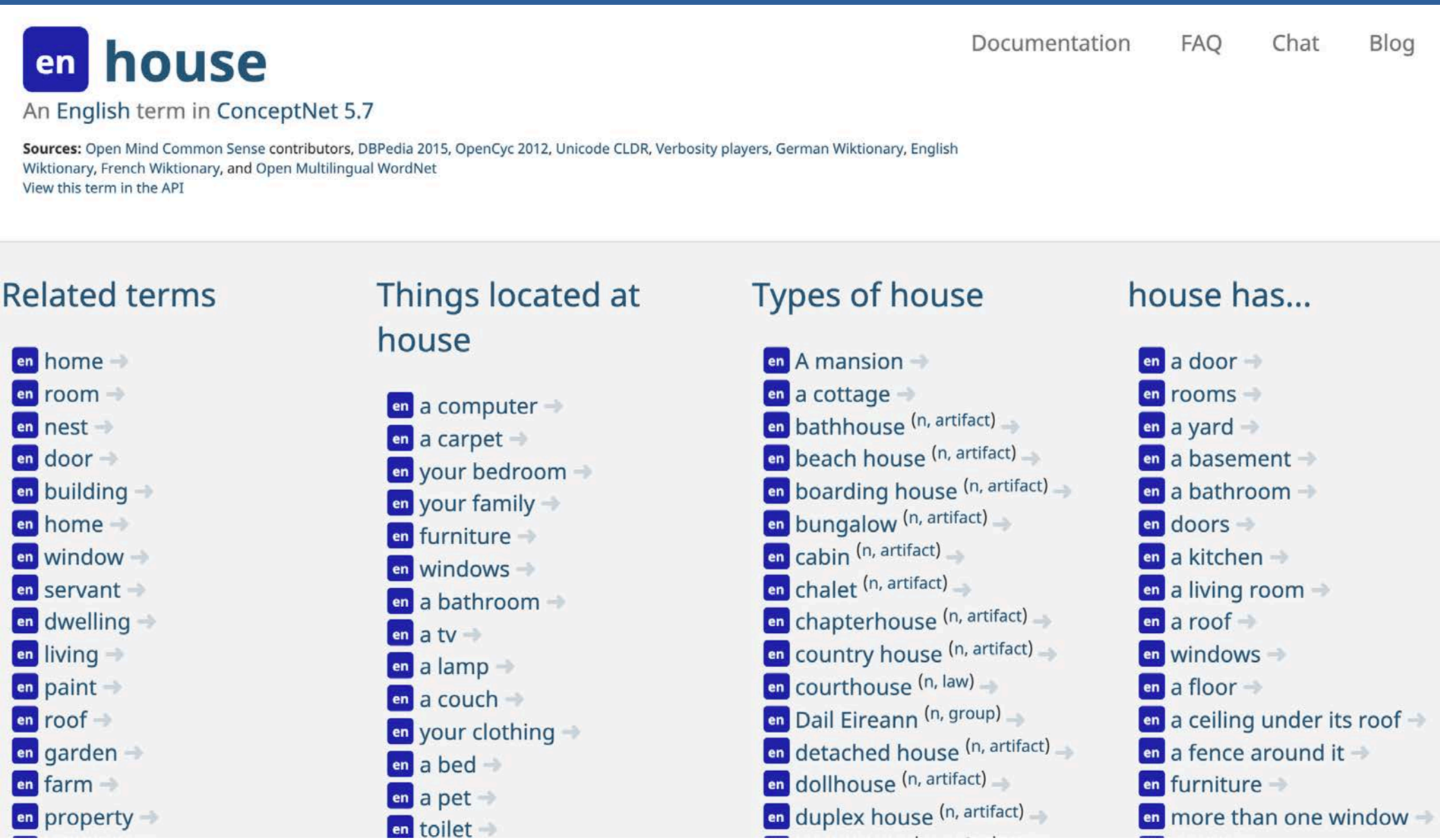
- Redes semánticas (fundamentación psicológica, Quillian 1968)
  - Representación basada en grafos dirigidos
    - Nodos para conceptos y aristas para relaciones
  - Permite razonar con categorías y relaciones entre conceptos
  - Representación descriptiva intuitiva para las personas

- Ejemplos on-line : Aplicaciones
  - Visualthesaurus: un diccionario visual
  - ConceptNet: BD conocimiento de sentido común (colaborativamente)

Introducción: Ejemplo Visualthesaurus



Introducción: Ejemplo ConceptNet

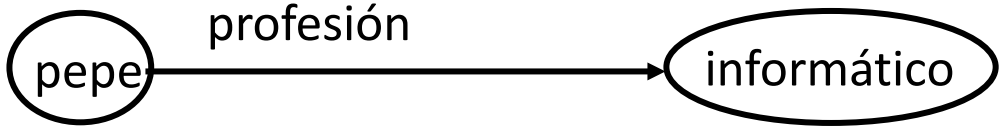


Introducción: Teoría Asociativa

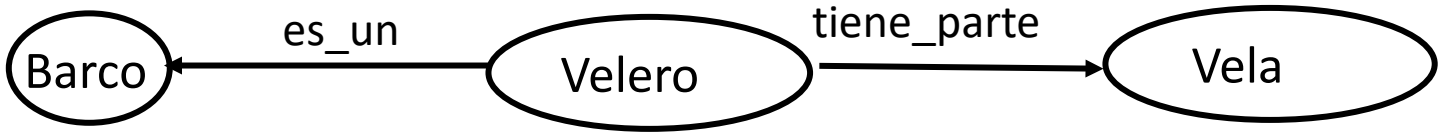
- Damos significado a un concepto mediante relaciones con otros conceptos
  - Relaciones sobre propiedades, comportamientos, usos, etc.
  - Ej: nieve → fría, blanca, hielo, muñeco de nieve...
- Organizamos nuestro conocimiento de forma jerárquica
    - *canario → ave → vertebrado → animal*
- Aprendemos:
  - Conectamos el nuevo concepto a otros mediante relaciones
    - *Canario ? Lo conecto con ave*
  - Ponemos propiedades en el nivel más abstracto posible
    - *Volar conectado con ave (no con canario)*
  - Ponemos excepciones directamente en los conceptos
    - *No vuela conectado con avestruz (no con ave)*
- Recordamos y razonamos: responder a una pregunta
  - Recorriendo/navegando por esas redes de conceptos y relaciones
  - Usamos la herencia y las intersecciones de conceptos en la navegación

Definición de Red Semántica (o asociativa)

- Representa conocimiento mediante un grafo dirigido
  - Nodos: conceptos
  - Arcos etiquetados: relaciones específicas entre conceptos
    - Concepto/propiedad, clase/subclase, agente/verbo/objeto...
    - Tipo (etiqueta): espacial, temporal, causal, rol desempeñado, etc…
    - Etiquetas “primitivas”: relaciones estándar (es_un, tiene_parte, instancia, …)
  - Inferencia: la herencia, navegación organizada
- Técnica de representación declarativa
- Significado de un concepto: sus conexiones con otros conceptos

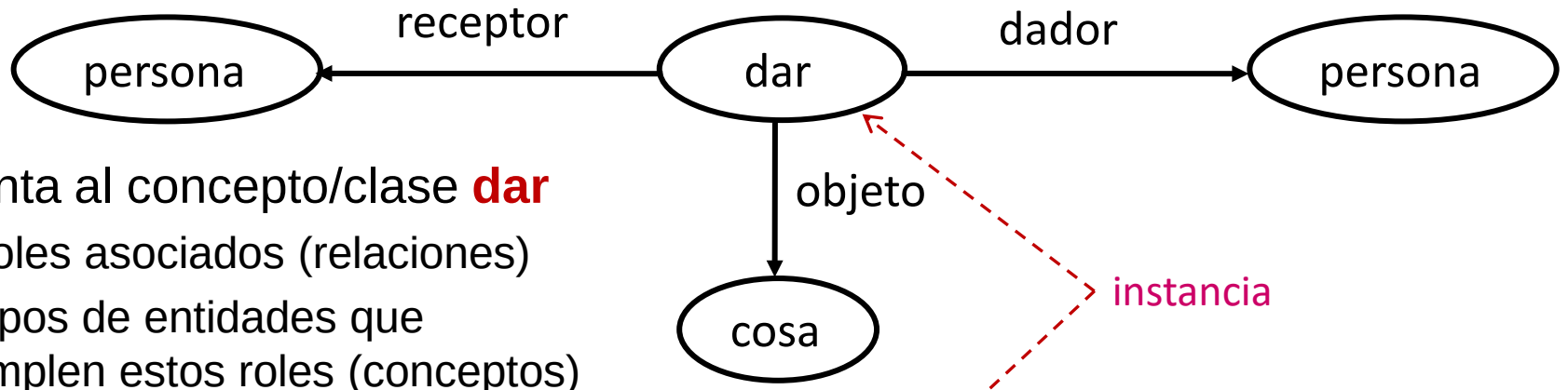


- “un velero es un barco con velas”



Ejemplo: Fragmentos de una red semántica

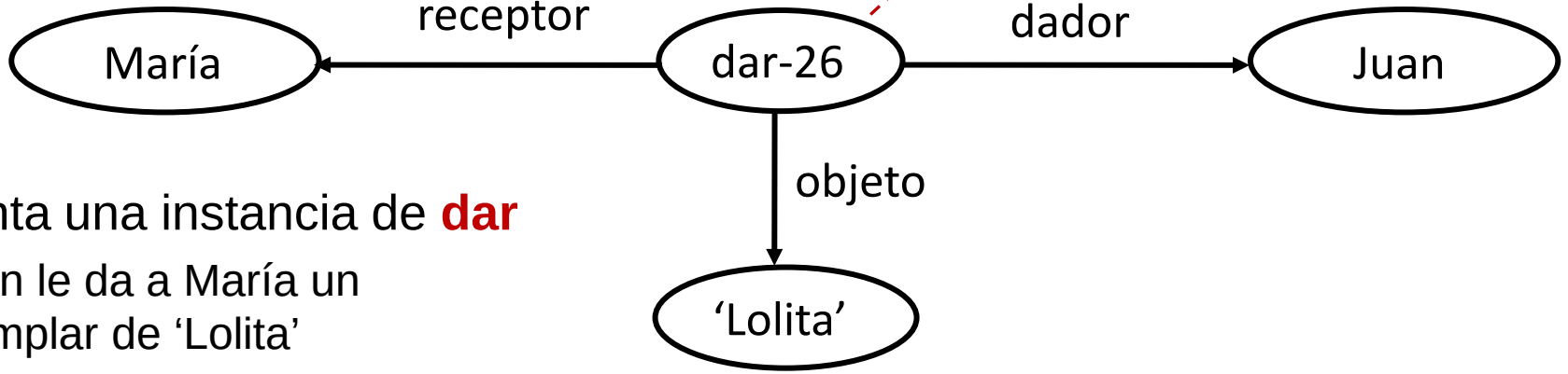
A)-



Representa al concepto/clase **dar**

- 3 roles asociados (relaciones)
- 3 tipos de entidades que cumplen estos roles (conceptos)

B)-



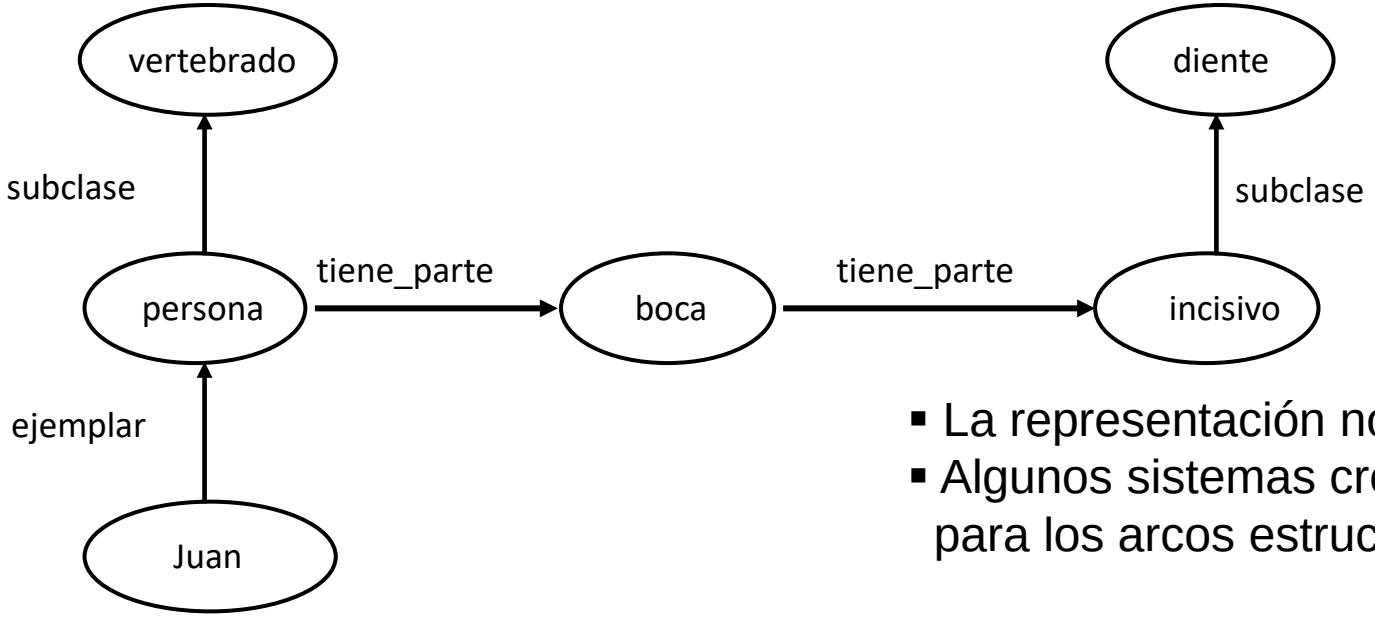
Representa una instancia de **dar**

- Juan le da a María un ejemplar de 'Lolita'
  - Juan y María son instancias de persona
  - 'Lolita' es instancia de libro, que a su vez es instancia de cosa

Tipos de arcos: relaciones entre conceptos

a)- Arcos estructurales (semántica independiente del dominio, “primitivas”)

- *instancia o ejemplar:* une un objeto con su tipo (clase)
- *subclase:* une una clase con otra más general
- *tiene_parte:* liga un objeto con sus componentes



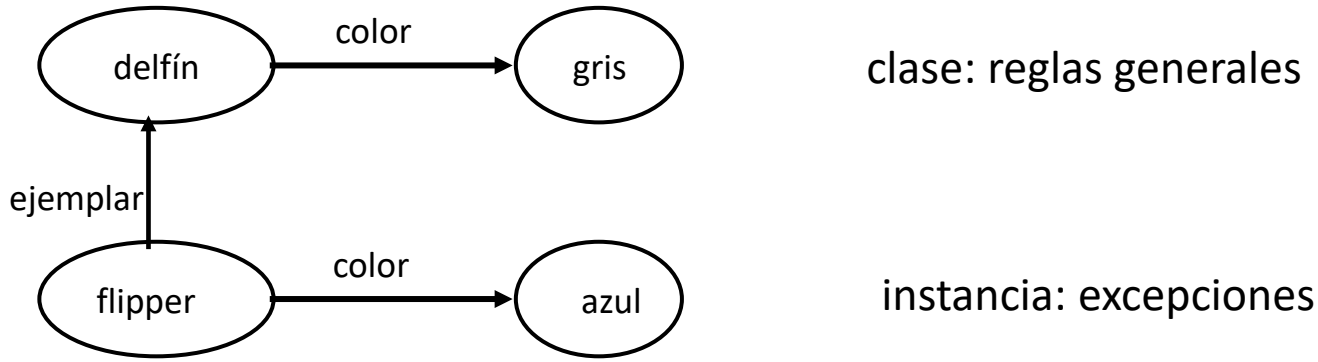
- La representación no es única
- Algunos sistemas crean inversas para los arcos estructurales

b)- Arcos descriptivos (semántica dependiente del dominio)

- Propiedades: *Profesión, Color_Pelo,* etc.
- Relaciones (no estructurales): *Amigo_de, Padre_de,* etc.

Tipos de Inferencia (1): Herencia de propiedades

- La notación de redes semánticas hace muy conveniente la utilización de razonamiento basado en herencia
- Algoritmo simple y eficiente con manejo de excepciones
  - Los nodos acceden a las propiedades definidas en otros nodos siguiendo los arcos *Instancia* (o *Ejemplar)* y *Subclase*
- Ventajas
  - Evita repetir propiedades
  - Permite compartir conocimiento entre diferentes conceptos de la red semántica



clase: reglas generales

instancia: excepciones

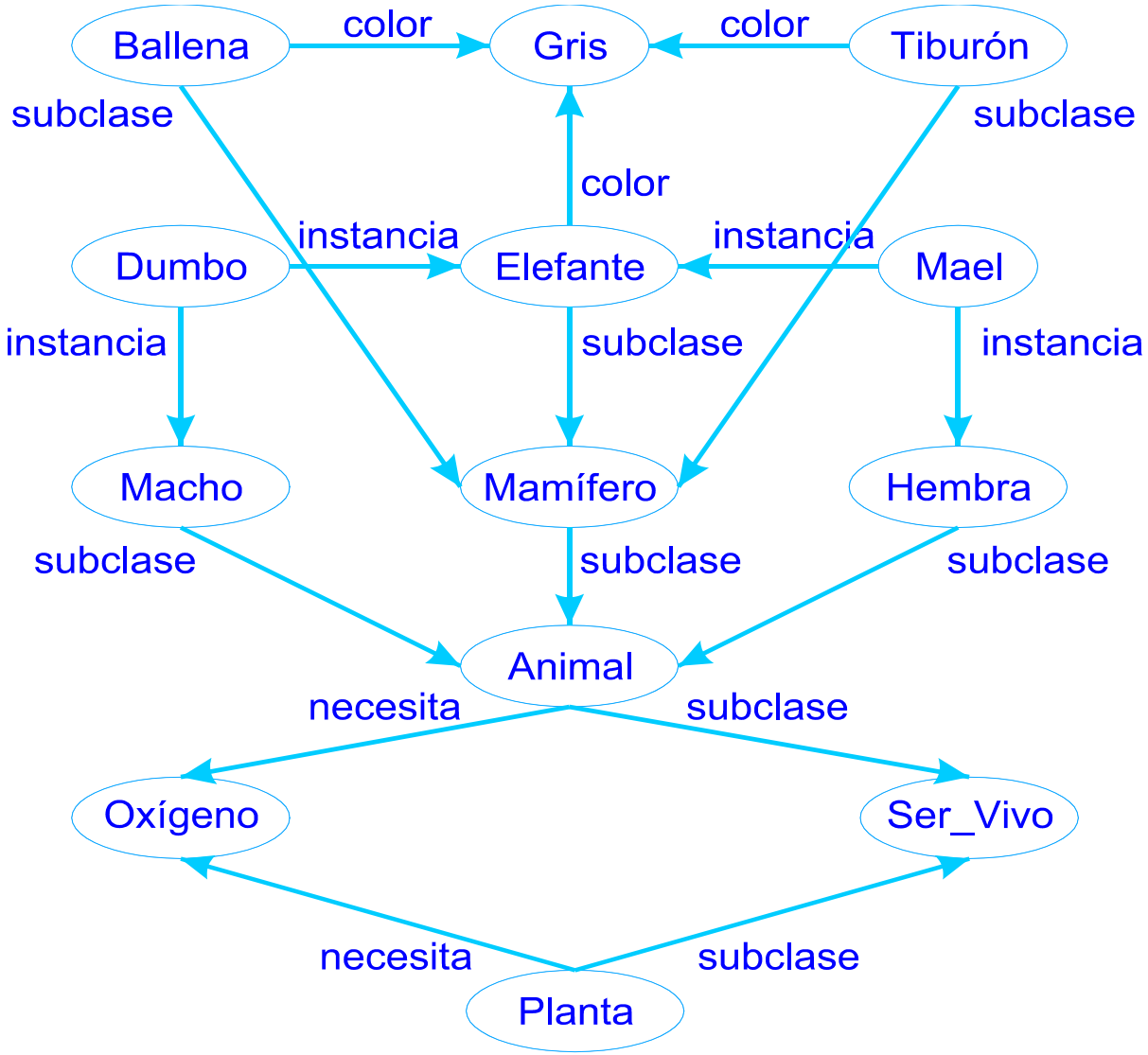
Herencia de propiedades: ejemplo

¿De qué color es Dumbo?

- Gris

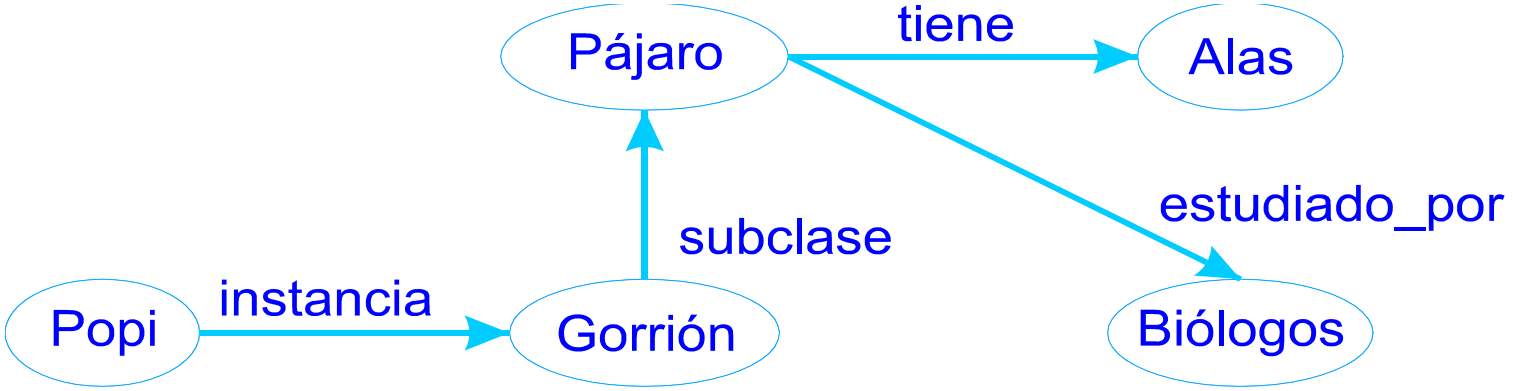
¿Qué puedo decir de Dumbo?

- Es un elefante
  - Es de color gris
- Es un macho
- Es un mamífero
- Es un animal
  - Necesita oxígeno
- Es un ser vivo



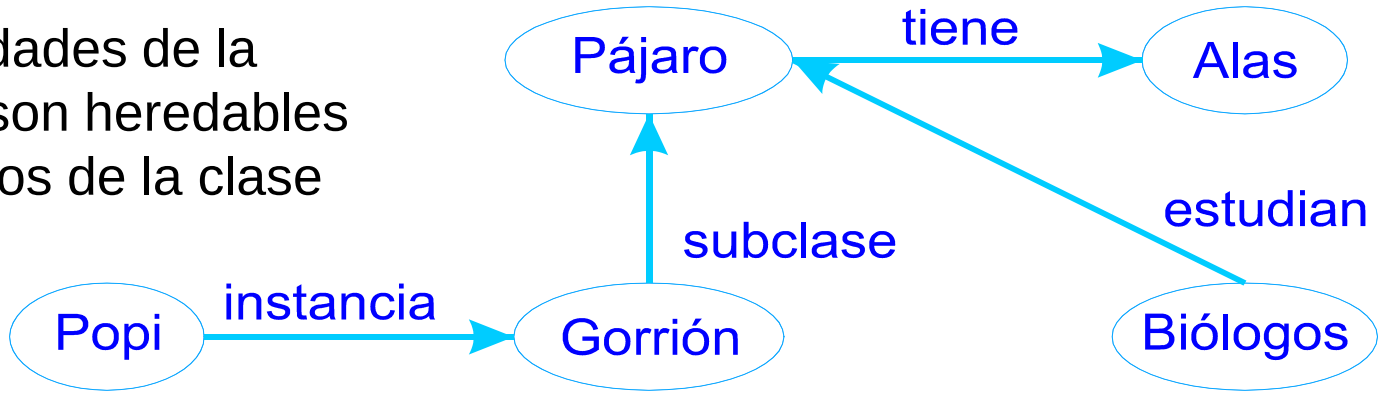
Herencia de propiedades: Problemas

- Herencia de propiedades que no son ciertas *(inferencias inválidas)*

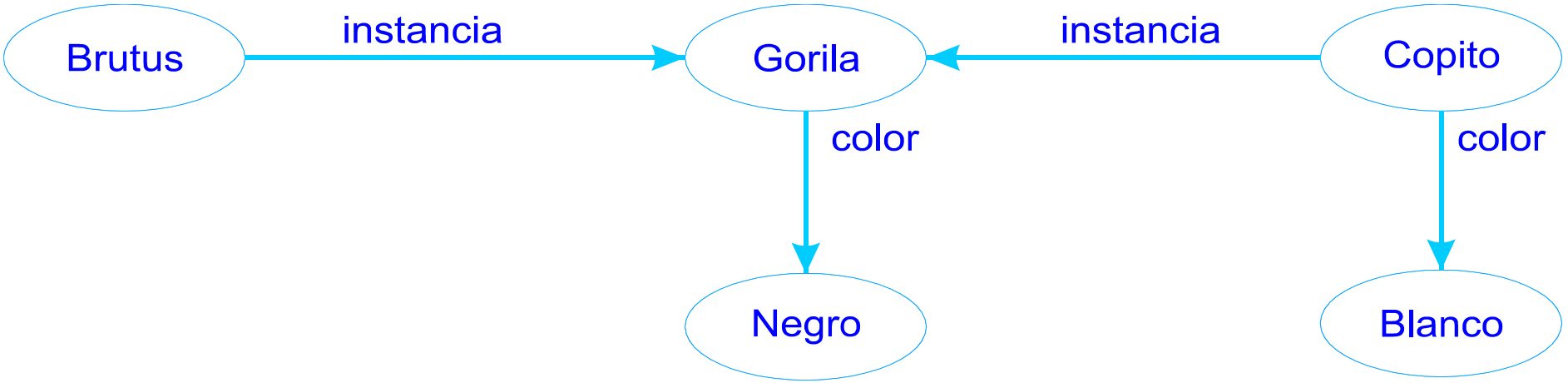


- Causa:
  - Algunas propiedades de la clase en sí, no son heredables por los individuos de la clase

Otra posibilidad mejorada:



Herencia de propiedades: Excepciones



- Se hereda el valor de la propiedad del nodo más cercano:
  - Brutus es de color negro *(hay herencia de la clase Gorila)*
  - Copito es de color blanco *(no hay herencia)*

- Si hay varios valores distintos a la misma distancia
  - Respuesta múltiple o no determinada
  - La herencia múltiple es habitual en las representaciones estructuradas

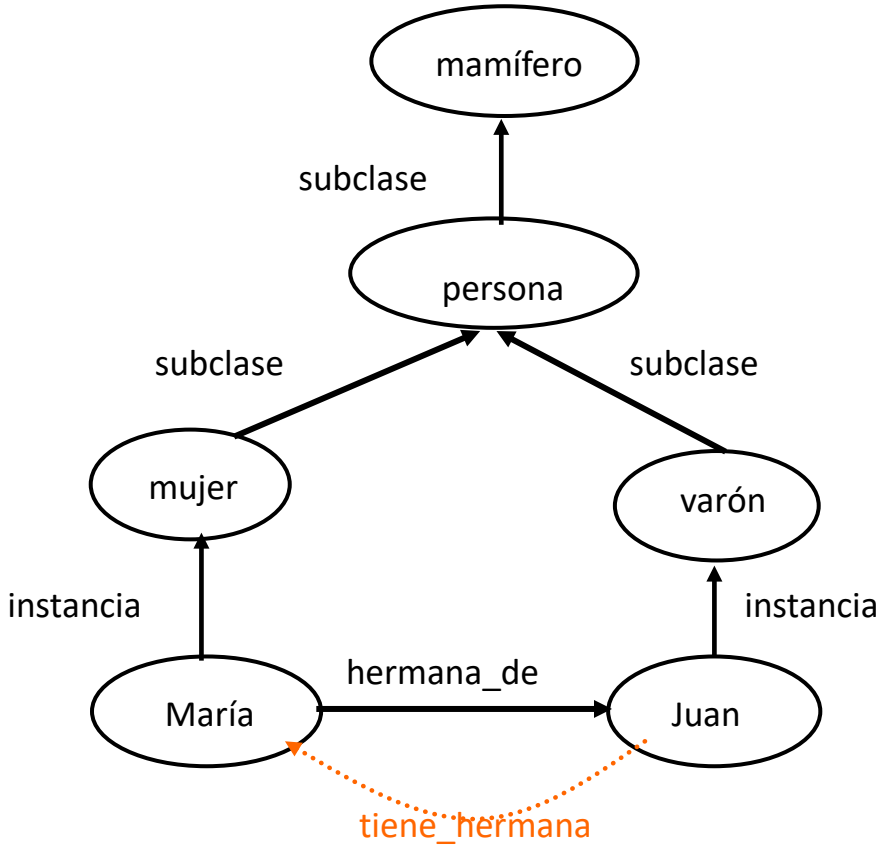
Tipos de Inferencia (2): intersección entre dos conceptos

- Para conocer la **relación entre dos conceptos C1 y C2** se utiliza un mecanismo de propagación de la activación
  - Inicialmente activamos ambos conceptos
  - La activación se propaga a los nodos adyacentes de forma sucesiva formando ”ondas” concéntricas (distancia 1, 2, 3, …) hasta que las ondas intersecan
  - La relación entre *C1* y *C2* queda determinada por los caminos (etiquetas y nodos) de *C1* al punto de intersección y de *C2* al punto de intersección
  - Si hay varios puntos de intersección a la misma distancia indica que existen varias relaciones distintas entre *C1* y *C2*
  - **Hipótesis**: los conceptos están más relacionados cuando más cerca están en el grafo de relaciones
  - **Nota**: no es exactamente lo mismo que el camino mínimo entre dos nodos por la dirección de las aristas (de C1 y C2 al nodo de intersección).

Uso de enlaces inversos

- Para buscar la intersección o contestar preguntas a veces es necesario generar la inversa de una relación
  - Algunos sistemas lo hacen automáticamente con los arcos estructurales

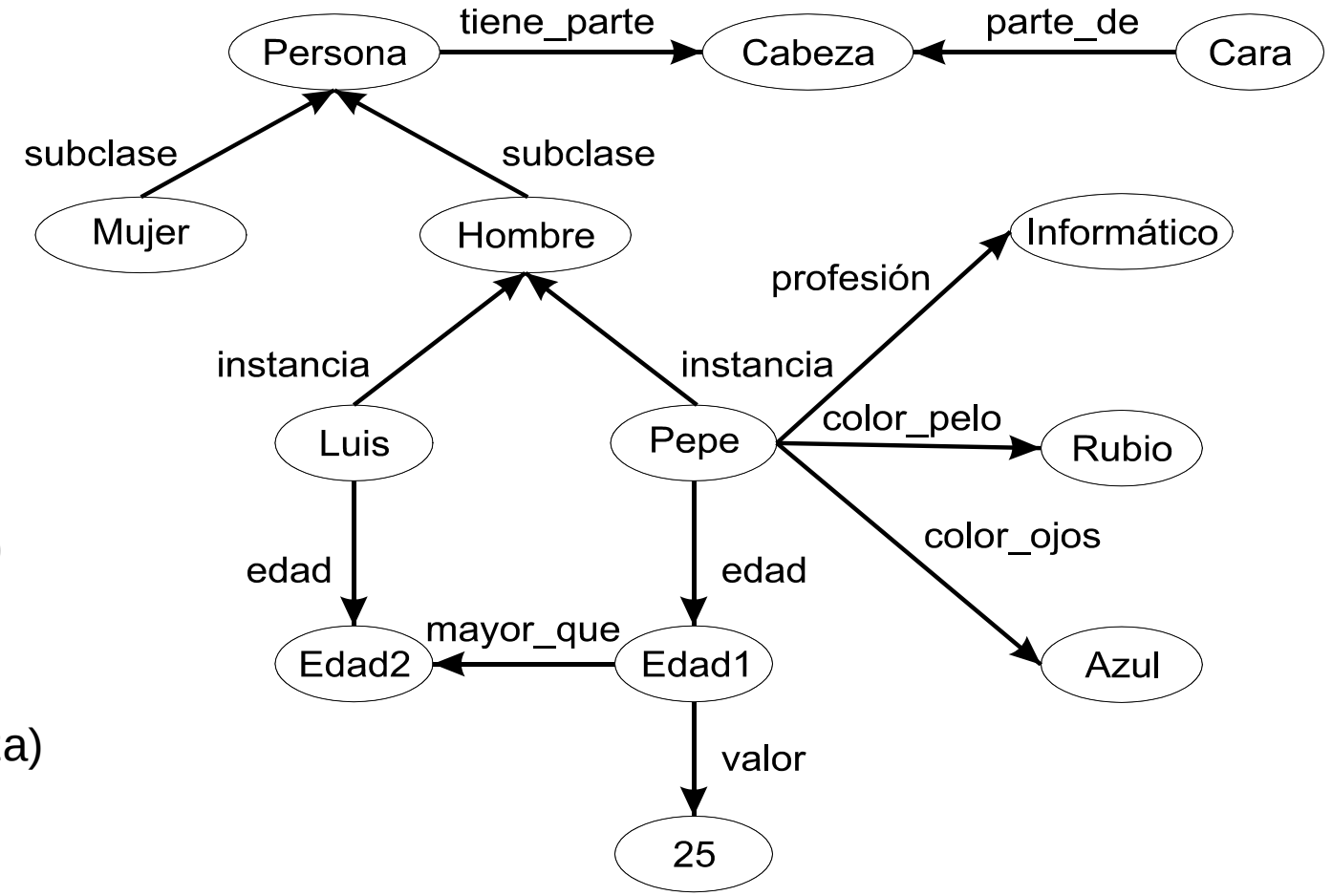
- ¿Quién es hermana de Juan?
  - El algoritmo de inferencia podría deducir que *tiene_hermana* es inversa de *hermana_de* y responder siguiendo el enlace de *Juan* a *María*
  - Si no, comprobaría cada mujer para ver si tiene un enlace *hermana_de* hacia *Juan*
    - Indexación directa sólo para los enlaces que salen de un nodo



Representación de relaciones binarias

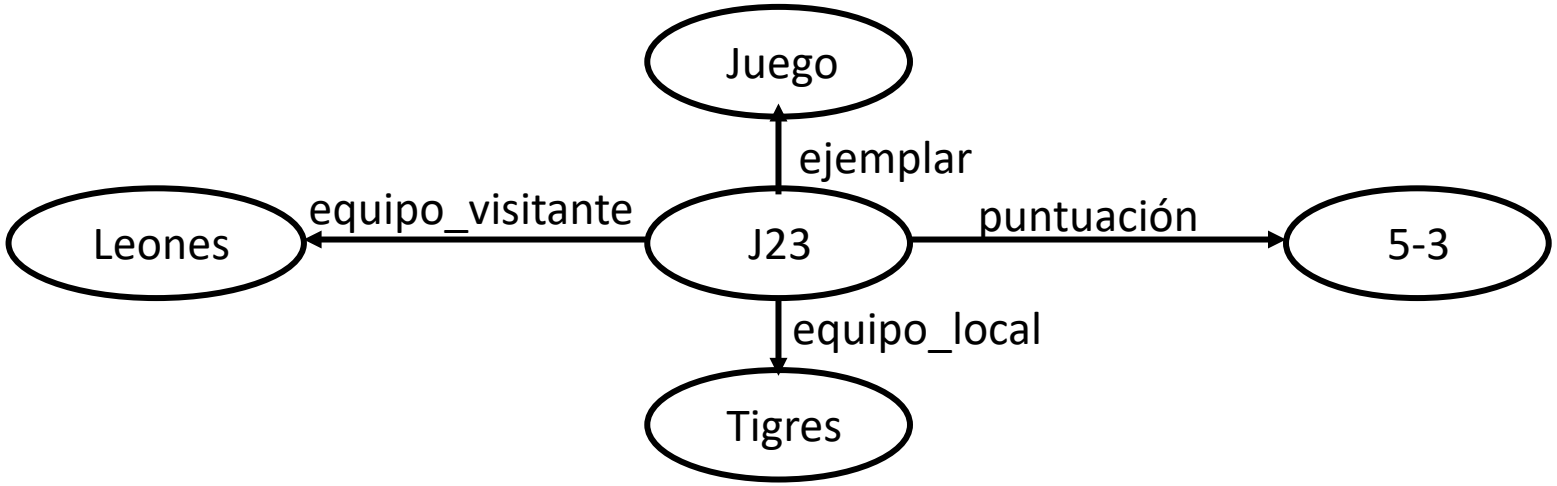
- Una red semántica es la forma natural de representar relaciones correspondientes a predicados binarios en lógica (sin variables)

subclase(Mujer, Persona)
subclase(Hombre, Persona)
instancia(Pepe, Hombre)
instancia(Luis, Hombre)
edad(Pepe, Edad1)
edad(Luis, Edad2)
valor(Edad1, 25)
mayor_que(Edad1, Edad2)
profesión(Pepe, Informático)
color_pelo(Pepe, Rubio)
color_ojos (Pepe, Azul)
tiene_Parte(Persona, Cabeza)
parte_de(Cara, Cabeza)



Representación de relaciones no binarias

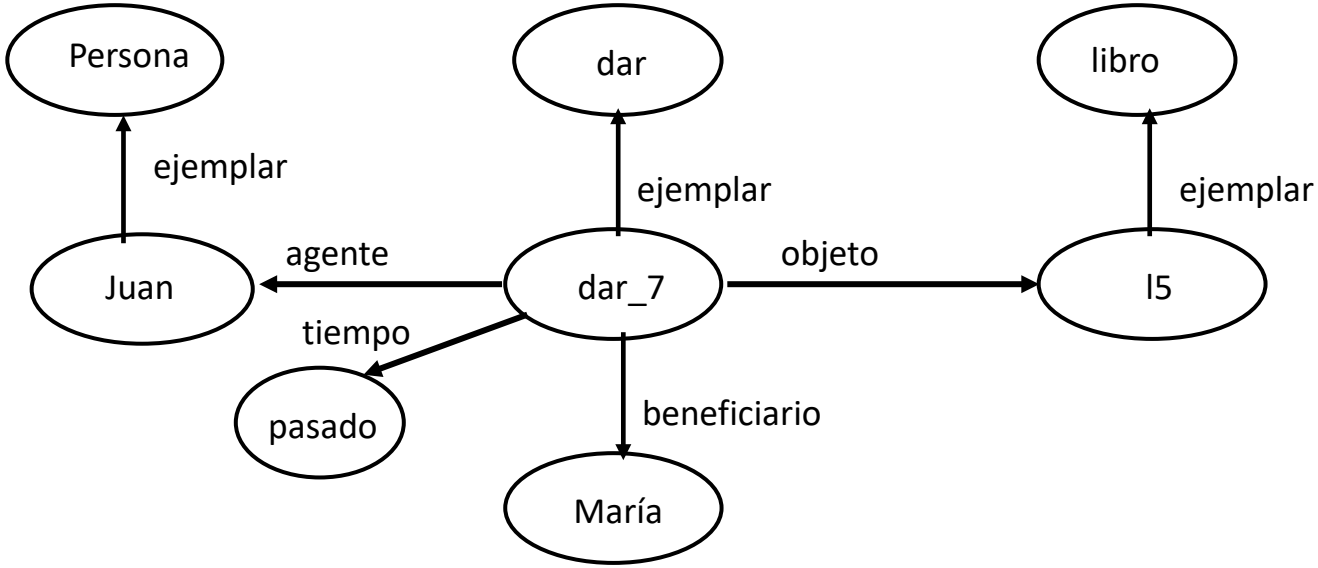
- Los enlaces representan relaciones binarias
  - ¡Un arco sólo tiene 2 extremos!
- La representación de relaciones n-arias
  - Convertirlas a formato binario
  - Se crea un nuevo objeto que representa a la relación concreta
    puntuación(Tigres, Leones, 5-3) *J23*
  - Se introducen predicados binarios para describir la relación de ese nuevo objeto con sus argumentos originales



- Esta técnica resulta útil para la representación de sucesos

Representación de sucesos

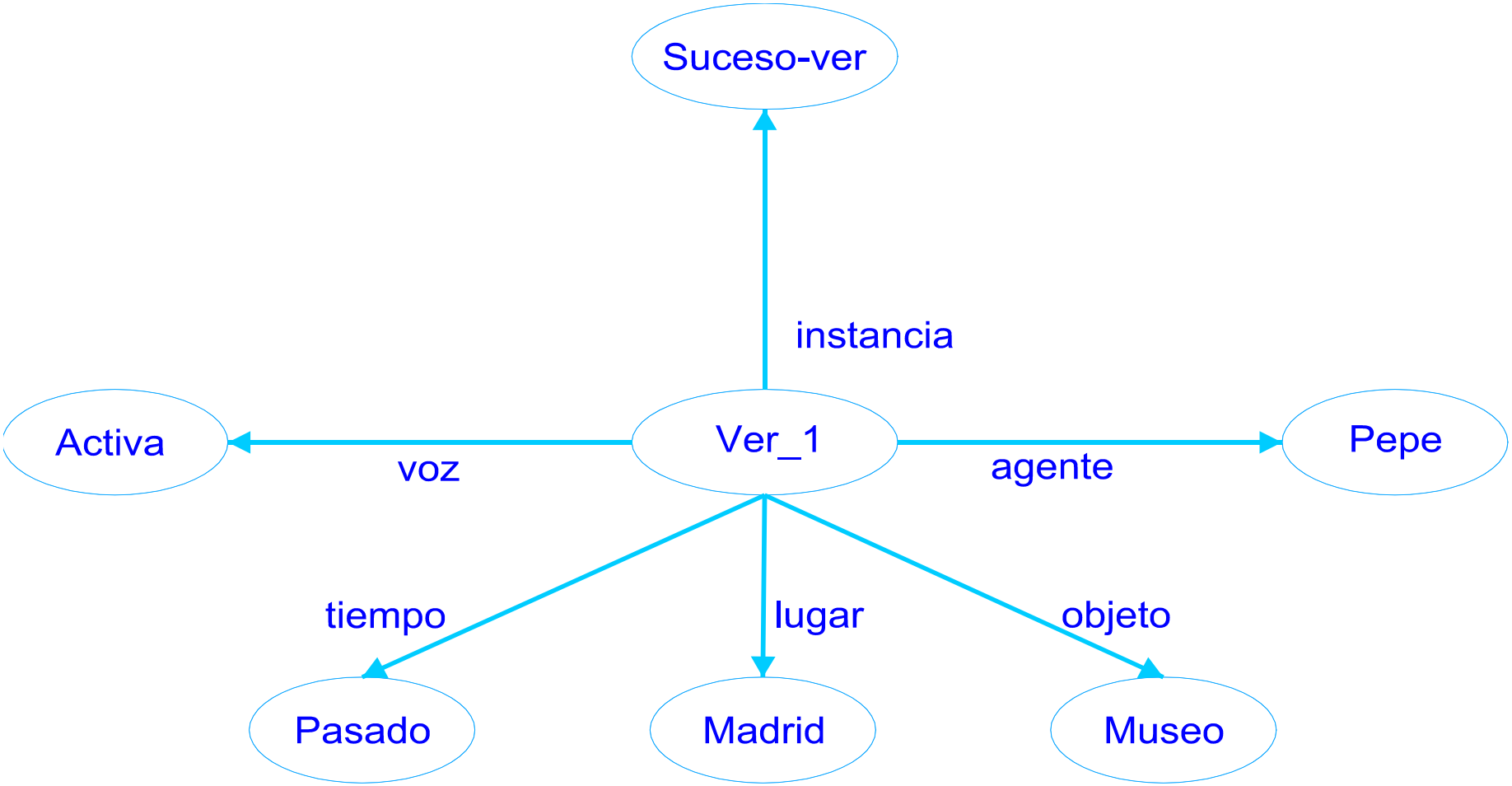
- Juan dio el libro a María



  - El objeto del suceso es un libro concreto que no está representado como tal en la frase dada por el usuario ➔ el sistema crea un objeto, ejemplar de libro y le da un nombre (*l5*)
  - *Juan* sí es un individuo concreto al igual que *María*
  - Este tipo de representación contesta preguntas de distinto tipo sobre el conocimiento que tenemos representado

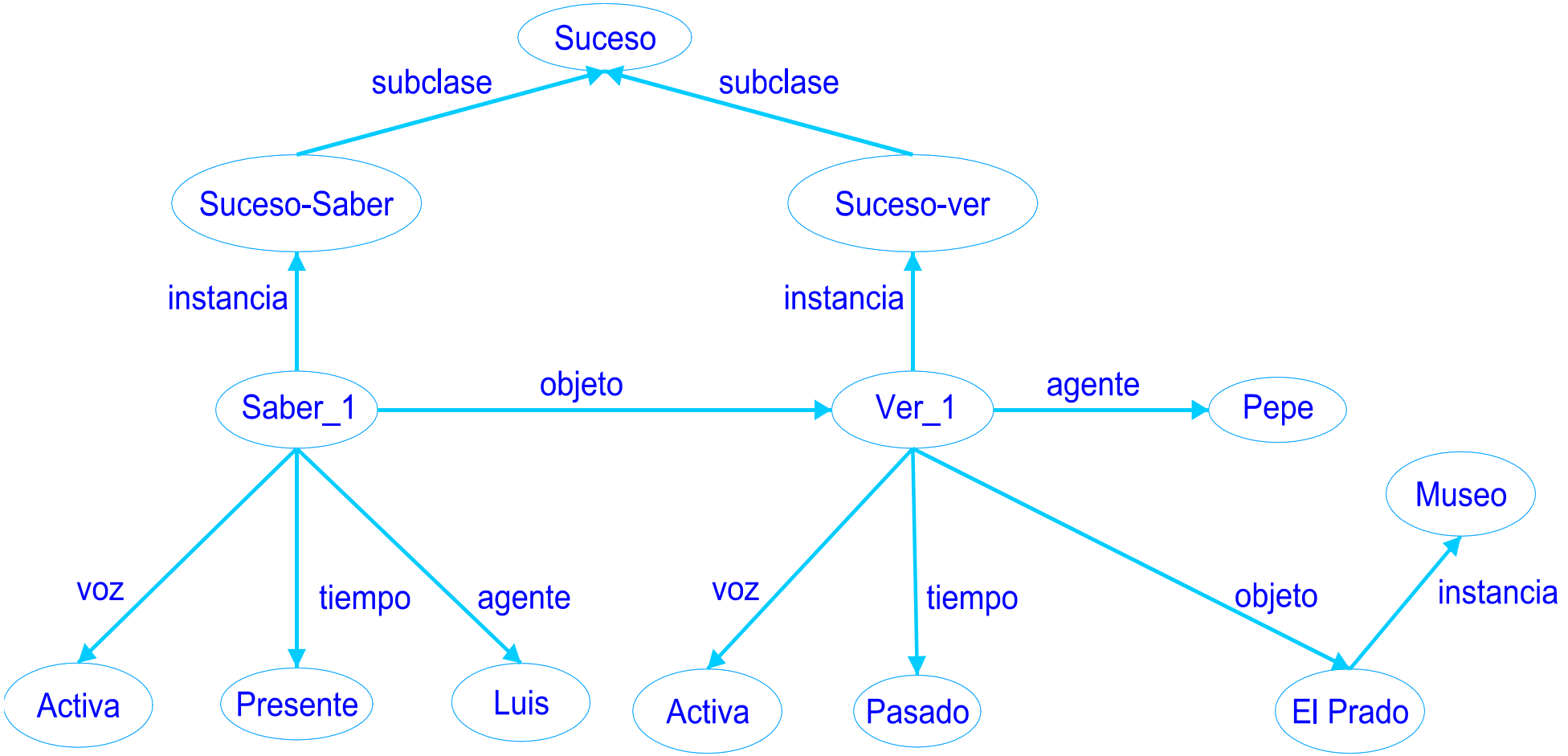
Representación de sucesos

- Pepe vio un museo en Madrid



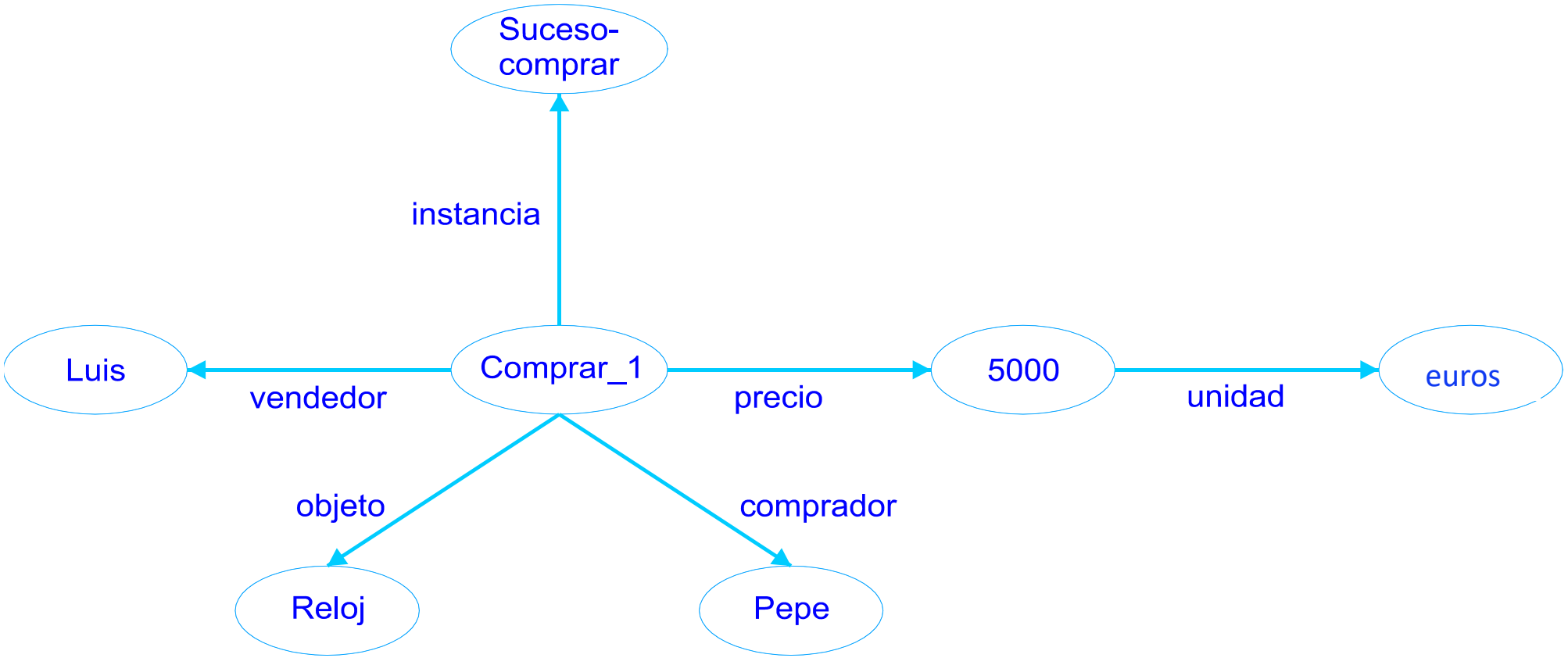
Representación de sucesos

- Luis sabe que Pepe vio el museo de El Prado



Representación de sucesos

- Pepe compra a Luis un reloj por 5000 euros
  - Lógica: compra(Pepe, Luis, Reloj, 5000, euros)



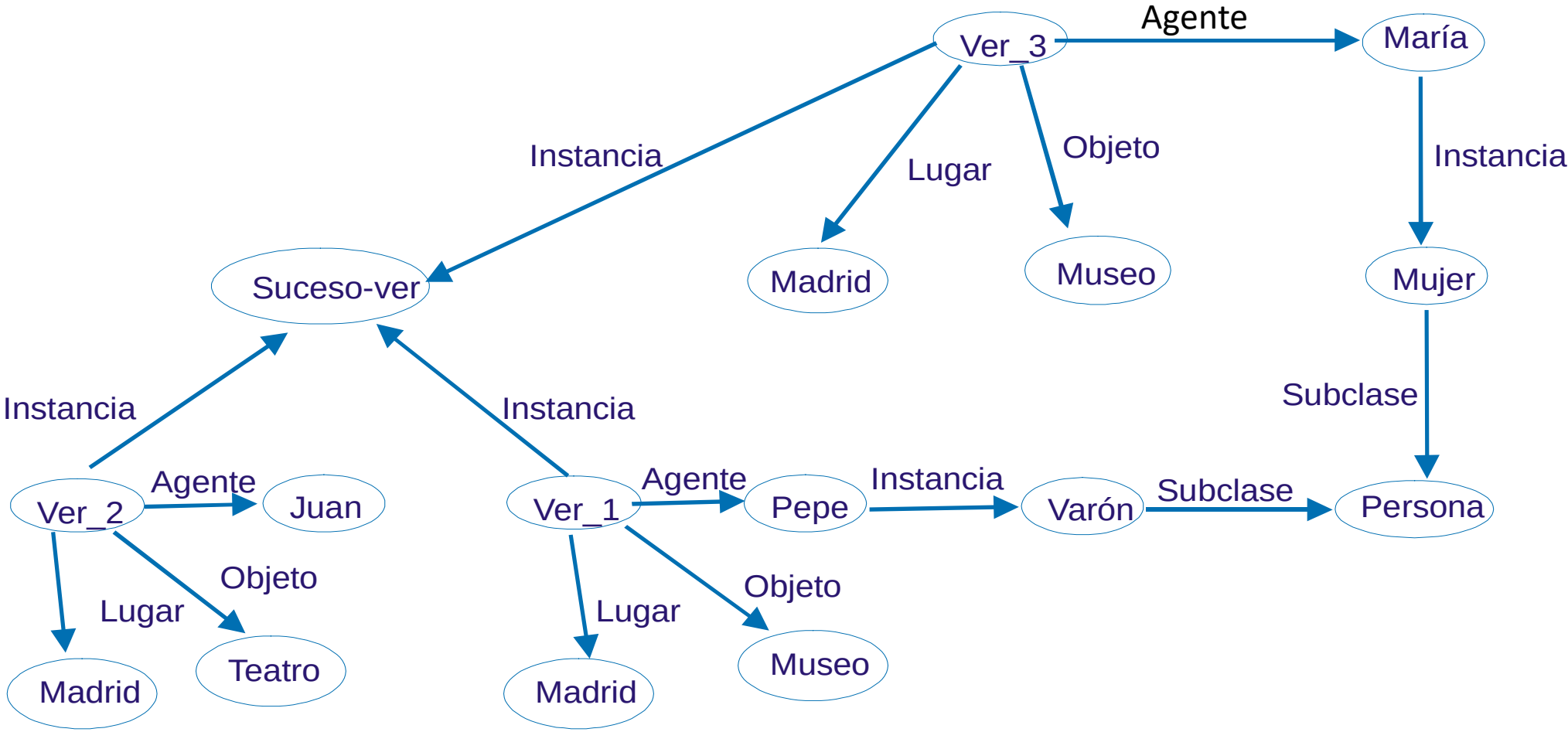
Tipos de Inferencia (3): Contestar preguntas / recuperar información

- Para contestar una pregunta
  - Se representa la pregunta en forma de grafo consulta
    - Pequeña red semántica que puede contener nodos variable
    - Mismo criterio de construcción de la base de conocimiento
  - Se buscan fragmentos de la BC que encajen con la consulta
    - **Cuidado**: directa o indirectamente a través de inferencias
    - Se puede ver como un problema de encaje de grafos sobre la red semántica que contiene todas las relaciones asertadas e inferidas
  - Cada posible encaje encontrado es una respuesta a la consulta

- La complejidad del proceso es importante
  - Si pregunto algo falso (o que el sistema no sepa) puede ser necesario analizar la red semántica entera

Contestar preguntas: ejemplo 1

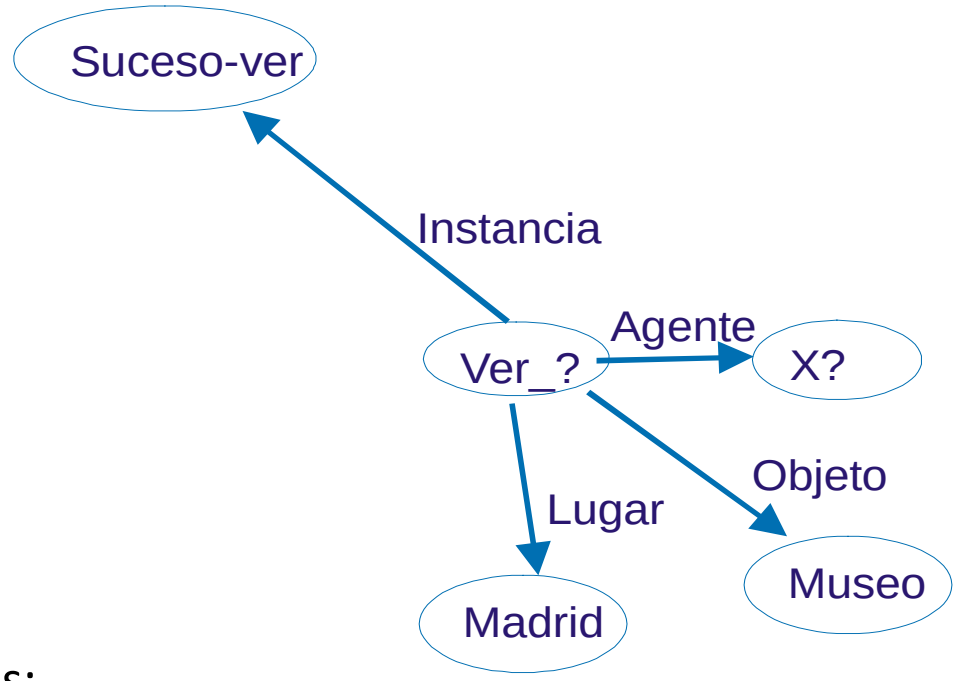
- Base de conocimiento: tres instancias del Suceso ver



🕮 **Transparencias de redes semánticas. Asunción Gómez Pérez. Facultad de Informática. UPM.**

Contestar preguntas: ejemplo 2

- Consulta: ¿quién vio un museo en Madrid?



Dos respuestas:

Equiparación 1:

Ver_? ≡ Ver_1

X? ≡ Pepe

Equiparación 2:

Ver_? ≡ Ver_3

X? ≡ María

Contestar preguntas: ejemplo 3

- Consulta: ¿algún varón vio algún museo en Madrid?

Una respuesta:

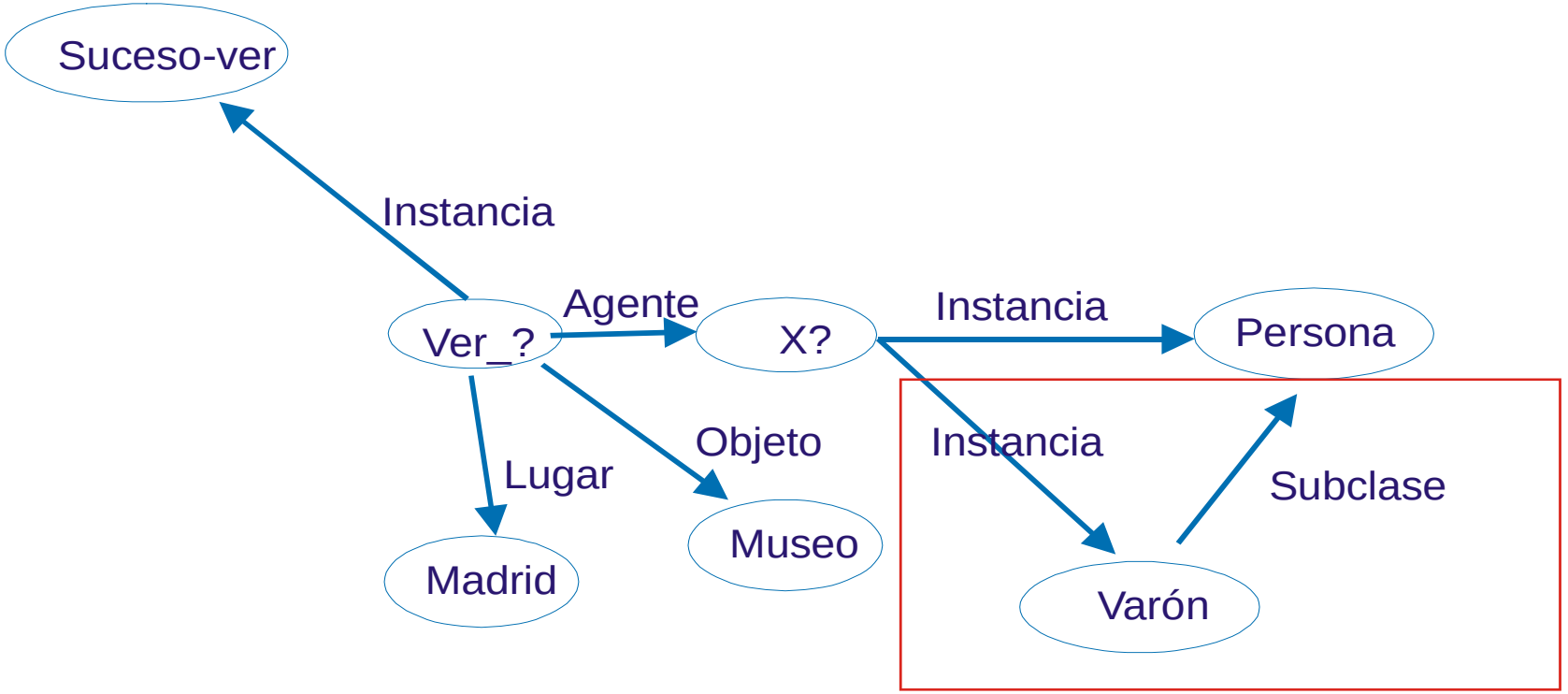
Equiparación:

Ver_? ≡ Ver_1

Varón? ≡ Pepe

Contestar preguntas: ejemplo 4

- Consulta: ¿alguna persona vio algún museo en Madrid?



(igual para Mujer)

Respuesta:

No existe equiparación directa con la consulta, pero puede inferirse para el caso de María y de Pepé

Redes semánticas: ventajas

- Más intuitiva y cercana al pensamiento humano que la lógica
  - Mismas ideas que la lógica, pero con la ventaja de que el conocimiento se organiza en torno a conceptos (y no a relaciones)
- Ayuda gráfica para visualización, algoritmo eficiente de herencia
  - Permite fácilmente el mecanismo de herencia con excepciones, siendo el proceso transparente (facilidad de visualizar los pasos)
- Mecanismo específico para obtener la relación entre dos conceptos: búsqueda de la intersección
  - Fue uno de los usos iniciales de redes semánticas en IA (Quillian, 1968): operación básica de recuperación de información, memoria asociativa
  - A menudo, necesita la generación de las relaciones inversas
- Contribución a la investigación en representación del conocimiento
  - Abrió una década de investigación en formalismos basados en redes
  - Éxito limitado como modelo psicológico de la memoria humana
  - Vuelven a coger fuerza con la Red Semántica (Semantic Web) y la iniciativa de Datos enlazados (Linked data)

Redes semánticas: inconvenientes

- Falta de estándares a la hora de nombrar nodos y arcos
  - Como en la lógica de predicados
    - Dificulta crear grandes bases de conocimiento de forma colaborativa
  - Sólo los arcos estructurales tienen una semántica bien definida
    - Diferentes interpretaciones de una misma red
- Problemas de eficiencia: aunque las inferencias se reduzcan a buscar intersecciones y responder preguntas
  - Respuestas negativas: cantidad descomunal de búsqueda
    - Esto prueba su no adecuación como modelo psicológico
    - ¿Hay un equipo de fútbol en Plutón? (una persona responde rápido)
- Imposibilidad de distinguir entre características propias del conjunto y características heredables por sus elementos
  - El cardinal del conjunto *delfín* es característica de la clase y NO de los individuos de la clase (como *flipper*)

Redes semánticas: inconvenientes

- Semántica (formal) limitada
  - Faltan negación, disyunción, símbolos de función anidados, cuantificadores.
    - Para cuantificadores: redes semánticas particionadas (qué parte está cuantificada existencialmente y cuál lo está universalmente)
  - Significados de nodos y arcos dependientes de las capacidades del sistema: confusión de semántica con detalles de implementación

- Dificultad para representar conocimiento procedimental
  - Imposibilidad de incluir meta-conocimiento (heurística) para dirigir la búsqueda
    - Extraer información puede ser muy ineficiente

- Escasez de estructura. Evolucionan para
  - Incluir estructuras para dar profundidad a la representación plana: Sistemas de marcos
  - Incluir relaciones estándar bien definidas: Linked data y RDFS
  - Usar formalismos lógicos que garantice su corrección y completitud: lógicas descriptivas y ontologías.

Web Semántica - Linked data - SPARQL

Web Semántica: contenidos

- Introducción
- RDF
- RDFS
- Linked Data
- SPARQL

Web Semántica

- En 1990 Tim Berners-Lee y Robet Cailliau crean la World Wide Web en el CERN
  - Navegador, HTML, HTTP, URI/URL
  - La Web se basa en documentos de texto (conocimiento no estructurado)
  - Pensada para personas (ambigüedad en resultados de búsquedas)

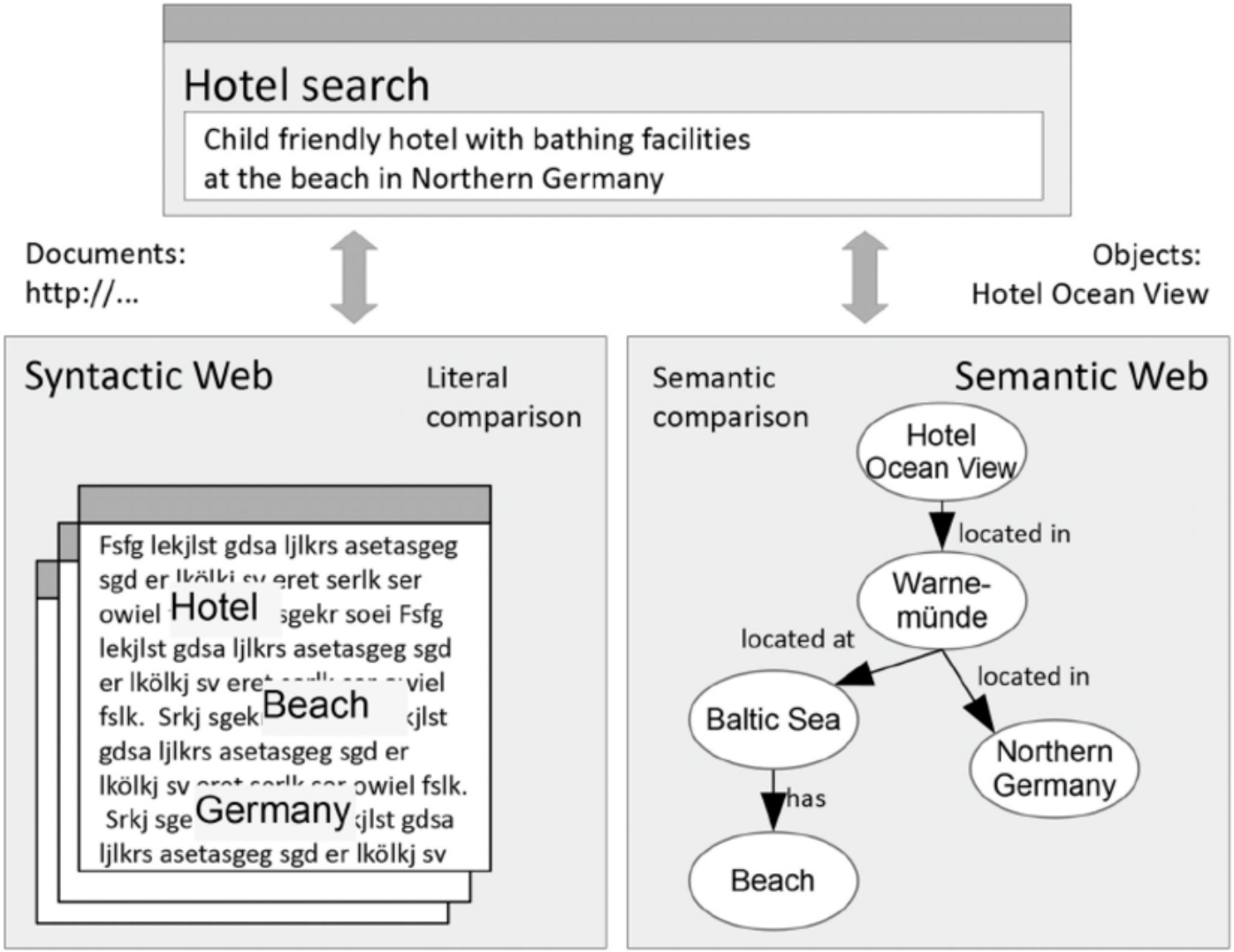
- En 2001 Tim Berners-Lee propone la idea de Web Semántica
  - Pensada para que las máquinas puedan interoperar (agentes software)
  - Conocimiento estructurado además de lenguaje natural
  - Añadir semántica para que las máquinas “entiendan” la información
  - Manipulación automática del conocimiento mediante lógica y motores de inferencia
  - Distintos formalismos de representación estándar buscando distintos puntos de equilibro
    - Cantidad de conocimiento vs capacidad de inferencia

Web Semántica

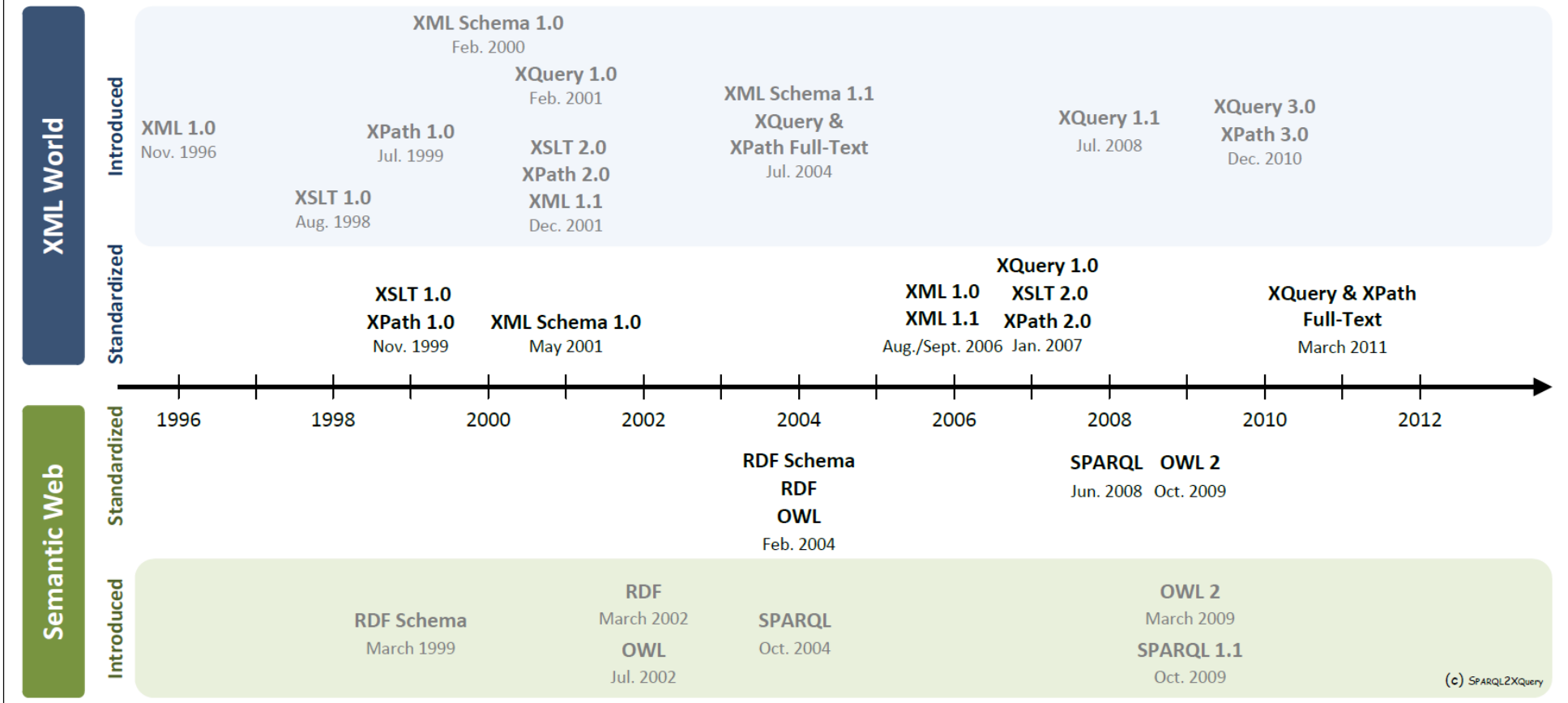
- Ventajas
  - Búsquedas más precisas de información (por significado)
  - Conocimiento accesible para programas
  - Podemos crear herramientas más potentes para personas
    - Que combinen la información disponible en distintas fuentes
    - Motores de búsqueda, asistentes personales, publicidad, …
  - Capacidad de razonar a distintos niveles de profundidad

- Desventajas
  - Es muy costoso anotar la información con etiquetas semánticas
    - ¡Y queremos anotar la Web!
  - Necesidad de estándares semánticos
  - Necesidad de vocabularios comunes
    - Debe ser un esfuerzo incremental y colaborativo
  - Coste computacional y volumen de información

Web Semántica

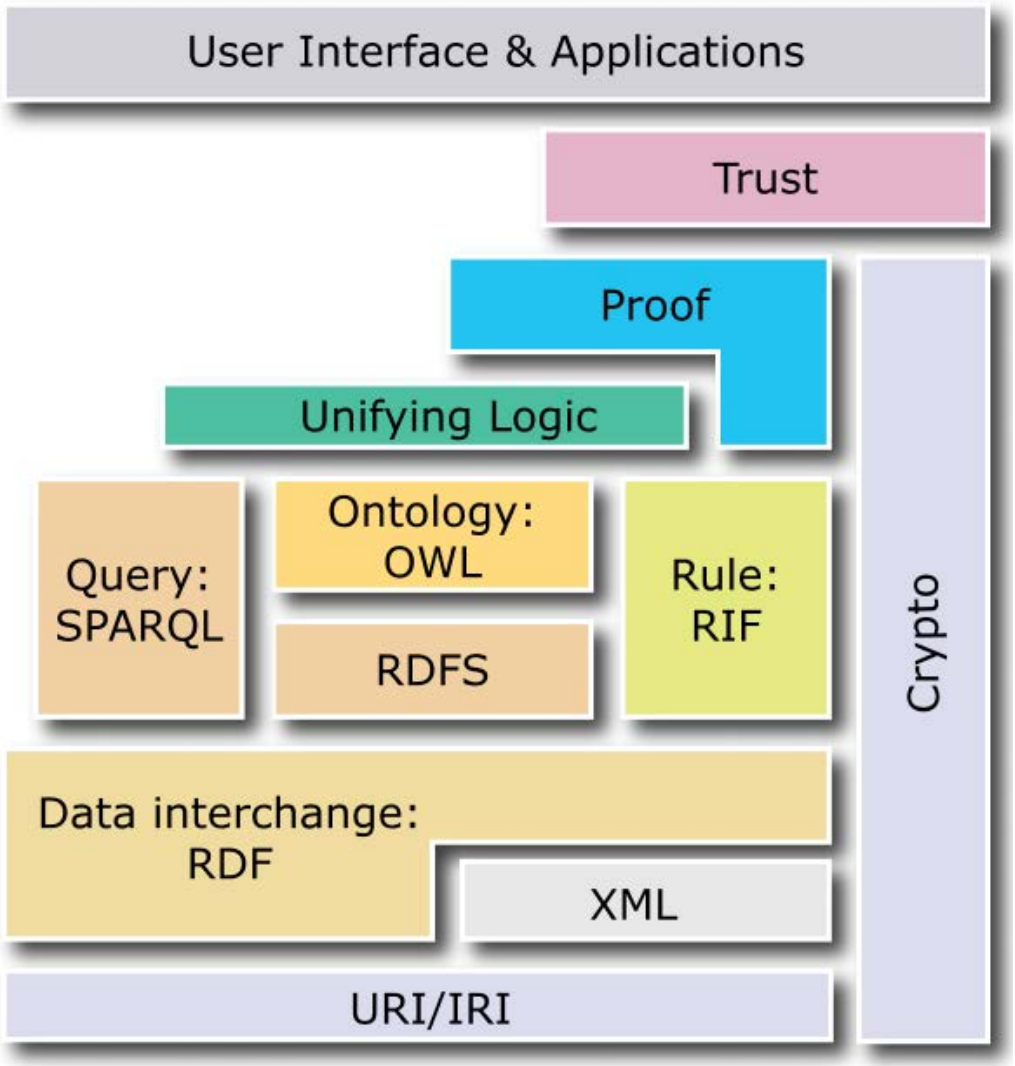


Estándares del W3C para la Web Semántica



This work is available under a CC BY-SA license. This means you can use/modify/extend it under the condition that you give proper attribution.
Please cite as: Bikakis N., Tsinaraki C., Gioldasis N., Stavrakantonakis I., Christodoulakis S.:
"The XML and Semantic Web Worlds: Technologies, Interoperability and Integration. A survey of the State of the Art"
To Semantic Hyper/Multi-media Adaptations: Schemes and Applications, Springer 2012 (to appear)

Estándares del W3C para la Web Semántica



- Identificadores no ambiguos para recursos (URI)
- Tripletas / redes semánticas (RDF)
- Ontologías con vocabulario predefinido (RDFS)
- Ontologías generales (OWL)
- Lenguajes de consultas (SPARQL)
- Lenguajes de reglas (RIF)
- Bases de conocimiento, razonadores, etc.

Resource Description Framework (RDF)

- Estándar de la W3C para el intercambio de datos en la Web (2004)
  - Permite definir redes semánticas como las que hemos visto
  - Pensado para ordenadores, no para personas
  - Facilita la integración de distintas bases de conocimiento

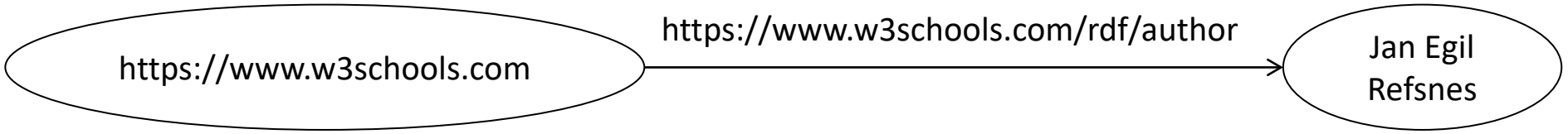
- Una tripleta (*statement*) consta de:
  - **Sujeto**: recurso web que se está describiendo
  - **Predicado / Propiedad**: propiedad o relación entre el sujeto y el objeto
  - **Objeto**: valor de la propiedad, puede ser un literal u otro recurso

- Los recursos se identifican mediante URIs

- Una BC es un conjunto de tripletas que forma un grafo dirigido o red semántica

RDF - Sentencias

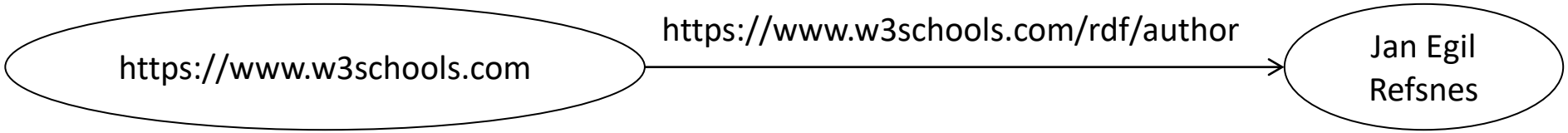
- "The autor of https://www.w3schools.com is Jan Egil Refsnes"
  - Sujeto: https://www.w3schools.com
  - *Predicado: author*
  - *Objeto:* Jan Egil Refsnes



- Existen conjuntos de relaciones estándar que facilitan integrar distintas bases de conocimiento distribuidas
  - ¡Estamos hablando de anotar la Web!

- Existen distintos formatos estándar de representación de grafos RDF (XML, N3, JSON-LD, etc.)

RDF - Formatos de representación



RDF/XML

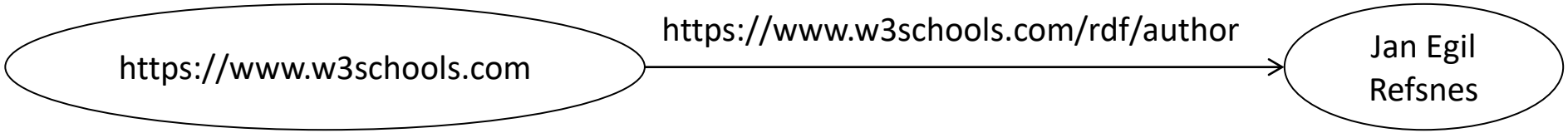
```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:si="https://www.w3schools.com/rdf/"
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <rdf:Description rdf:about="https://www.w3schools.com">
    <si:author>Jan Egil Refsnes</si:author>
  </rdf:Description>
</rdf:RDF>
```

prefijos

datos

- Los prefijos o espacios de nombres permiten escribir el nombre de las entidades de forma compacta
  - si:autor en lugar de https://www.w3schools.com/rdf/author

RDF - Formatos de representación



N3

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix si: <https://www.w3schools.com/rdf/> .
<https://www.w3schools.com> si:author "Jan Egil Refsnes" .
```

prefijos

datos

JSON-LD

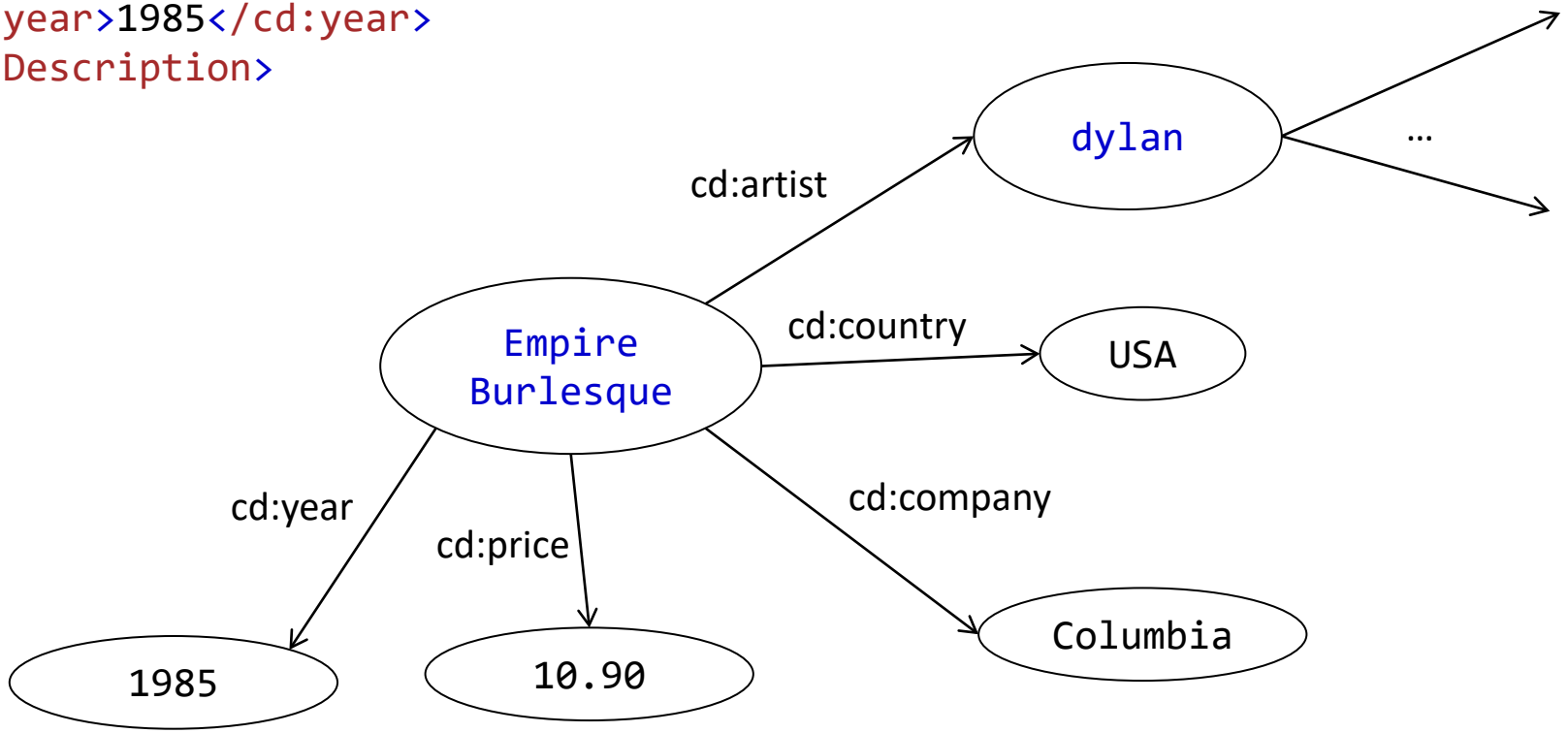
```
{
  "@context": {
    "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
    "si": "https://www.w3schools.com/rdf/",
  },
  "@id": "https://www.w3schools.com",
  "si:author": "Jan Egil Refsnes"
}
```

prefijos

datos

RDF – Un ejemplo un poco más complejo

```
<rdf:Description rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
  <cd:artist rdf:resource="http://www.recshop.fake/cd/dylan" />
  <cd:country>USA</cd:country>
  <cd:company>Columbia</cd:company>
  <cd:price>10.90</cd:price>
  <cd:year>1985</cd:year>
</rdf:Description>
```



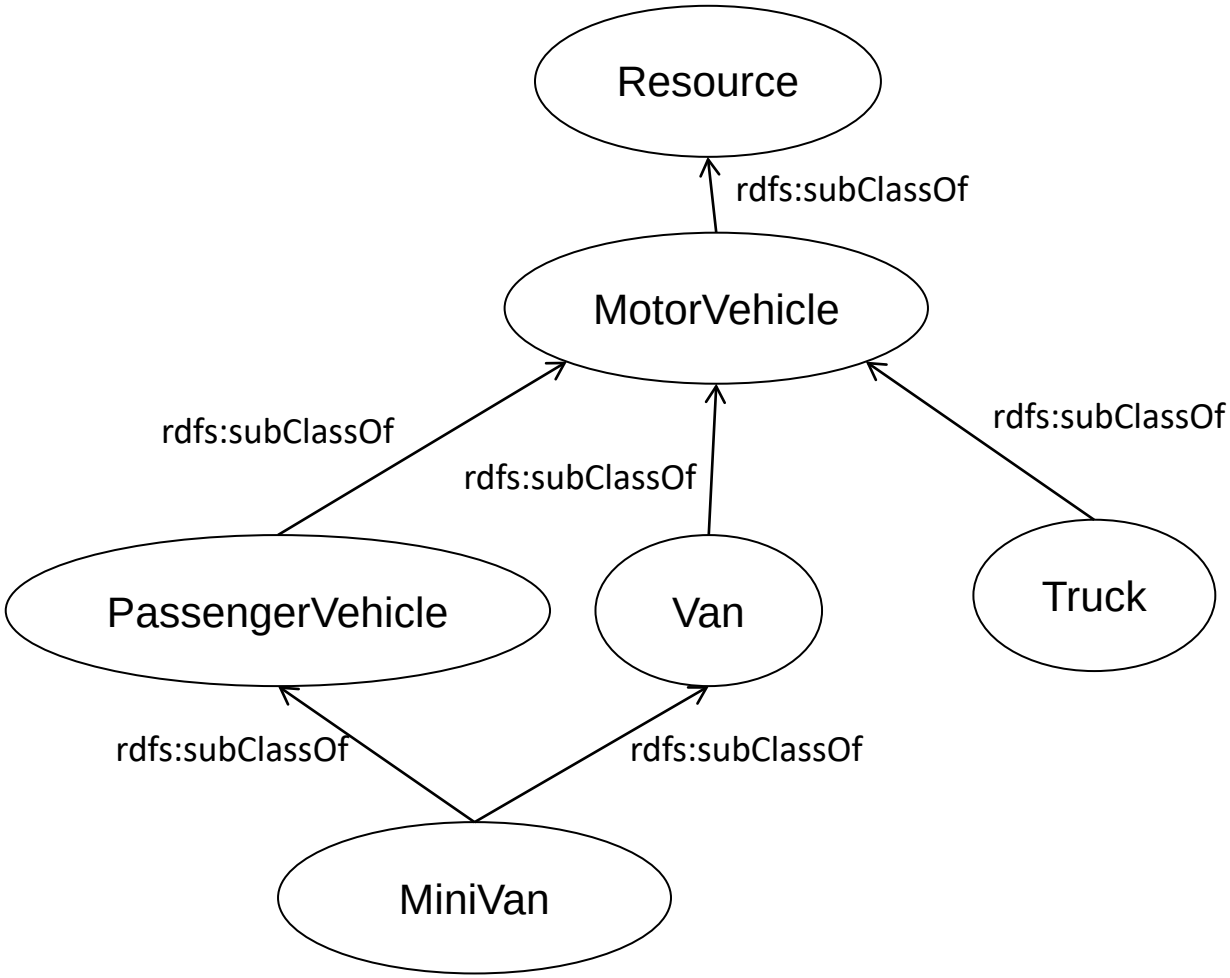
RDF Schema (RDFS)

- RDFS extiende RDF con un vocabulario estándar con el que definir tipos
  - Clases y propiedades
  - Jerarquías de clases y propiedades (taxonomías)
  - Rango y dominio de las propiedades (básicos)
  - En resumen: ontologías básicas para definir el vocabulario de un dominio

- Aún tiene muchas limitaciones porque carece de mecanismos para establecer:
  - Restricciones de existencia y cardinalidad
  - Propiedades transitivas, inversas o simétricas
  - Restricciones complejas para el rango y dominio de las propiedades
    - Conectivas lógicas: and, or, not

- OWL es el estándar de la W3C para definir ontologías más generales

RDFS – Ejemplo de jerarquía de clases



Ejemplo de RDF Schema (RDFS)

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
<rdfs:Class rdf:about="#MotorVehicle">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class >
<rdfs:Class rdf:about="#PassengerVehicle">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class >
<rdfs:Class rdf:about="#Van">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class >
<rdfs:Class rdf:about="#MiniVan">
  <rdfs:subClassOf rdf:resource="#Van"/>
  <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
</rdfs:Class >
<rdfs:Class rdf:about="#Truck">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class >
</rdf:RDF>
```

RDF /RDFS - Clases

Element	Class of	Subclass of
rdfs:Class	All classes	
rdfs:Datatype	Data types	Class
rdfs:Resource	All resources	Class
rdfs:Container	Containers	Resource
rdfs:Literal	Literal values (text and numbers)	Resource
rdf:List	Lists	Resource
rdf:Property	Properties	Resource
rdf:Statement	Statements	Resource
rdf:Alt	Containers of alternatives	Container
rdf:Bag	Unordered containers	Container
rdf:Seq	Ordered containers	Container
rdfs:ContainerMembershipProperty	Container membership properties	Property
rdf:XMLLiteral	XML literal values	Literal

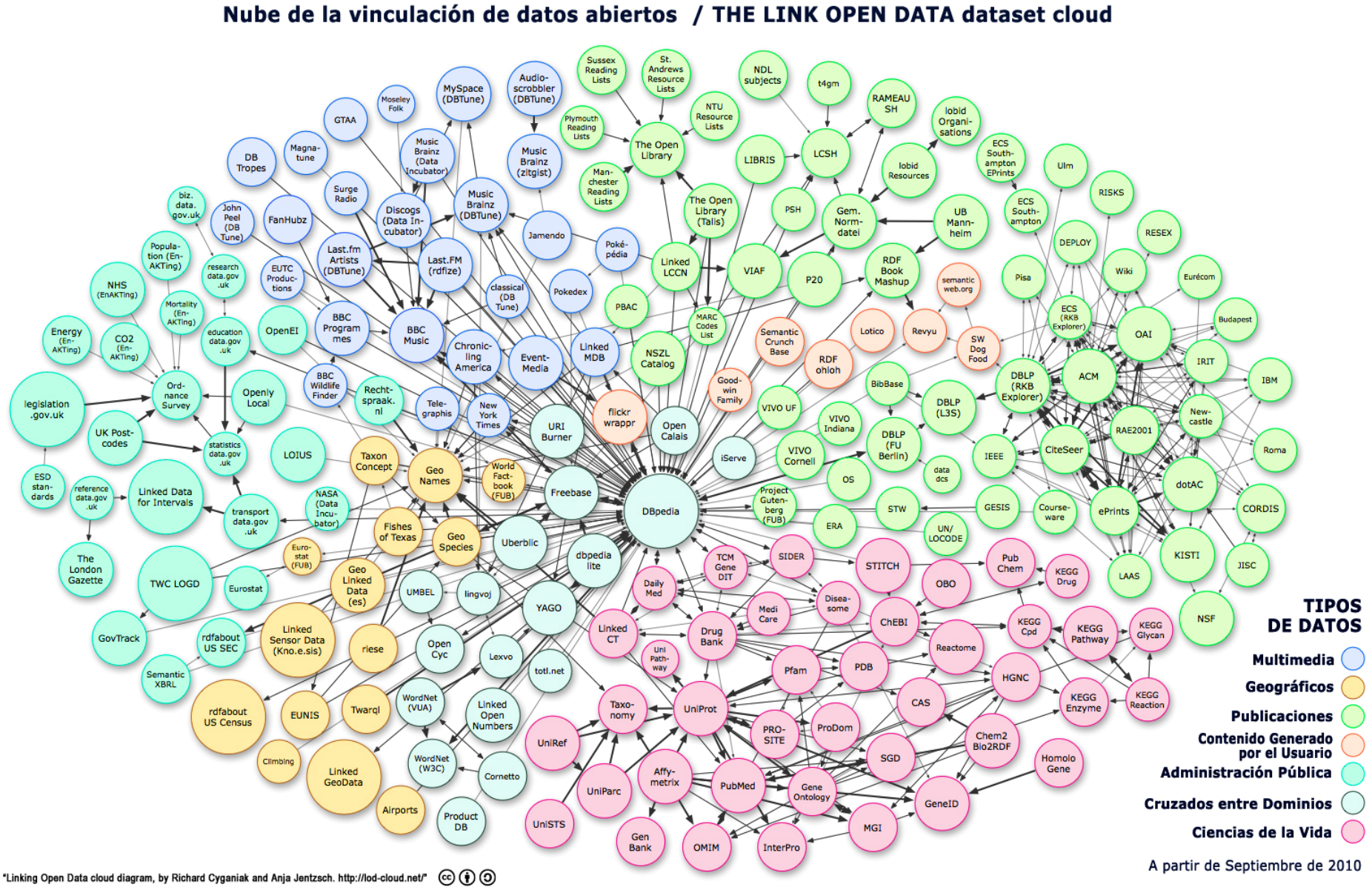
RDF / RDFS - Propiedades

Element	Domain	Range	Description
rdfs:domain	Property	Class	The domain of the resource
rdfs:range	Property	Class	The range of the resource
rdfs:subPropertyOf	Property	Property	The property is a sub property of a property
rdfs:subClassOf	Class	Class	The resource is a subclass of a class
rdfs:comment	Resource	Literal	The human readable description of the resource
rdfs:label	Resource	Literal	The human readable label (name) of the resource
rdfs:isDefinedBy	Resource	Resource	The definition of the resource
rdfs:seeAlso	Resource	Resource	The additional information about the resource
rdfs:member	Resource	Resource	The member of the resource
rdf:subject	Statement	Resource	The subject of the resource in an RDF Statement
rdf:predicate	Statement	Resource	The predicate of the resource in an RDF Statement
rdf:object	Statement	Resource	The object of the resource in an RDF Statement
rdf:value	Resource	Resource	The property used for values
rdf:type	Resource	Class	The resource is an instance of a class

Linked Data

- Publicación de datos estructurados usando estándares de la Web Semántica (URI, HTTP, RDF)
  - El objetivo es poder construir programas que “beban” de muchas fuentes y puedan combinar la información para realizar tareas complejas
- Ventajas
  - Facilita el intercambio de información entre distintas organizaciones
  - Facilita el desarrollo de aplicaciones que integren distintas bases de conocimiento
  - Promueve la transparencia y la confianza (por ej. datos de la administración pública)
- Normalmente, los datos se almacenan en formatos más eficientes (bases de datos relacionales) pero se ofrecen interfaces de consulta SPARQL

Linked Data


Nube de la vinculación de datos abiertos / THE LINK OPEN DATA dataset cloud
TIPOS DE DATOS
Multimedia
Geográficos
Publicaciones
Contenido Generado por el Usuario
Administración Pública
Cruzados entre Dominios
Ciencias de la Vida
A partir de Septiembre de 2010
"Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. http://lod-cloud.net/"

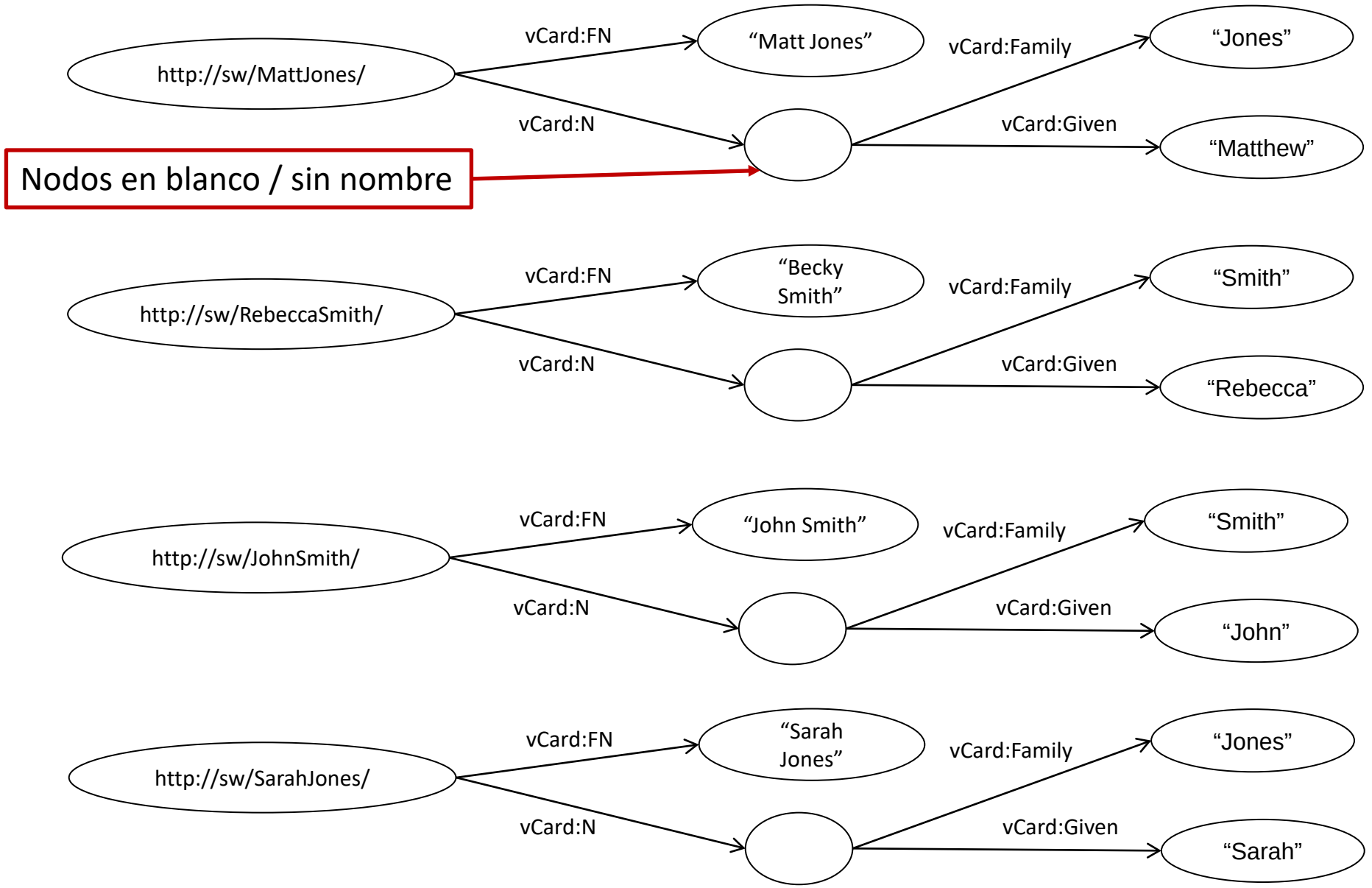
SPARQL

- SPARQL Protocol and RDF Query Language

- Lenguaje de consultas sobre bases de conocimiento / grafos RDF
  - Similar a SQL para bases de datos relacionales
  - Estándar de la W3C desde 2008

- Las consultas definen grafos con variables. Responder a una consulta consiste en encontrar las variables que permiten hacer *matching* con partes del grafo RDF (base de conocimiento).

SPARQL – Ejemplo de base de conocimiento



SPARQL – Consultas sencillas

- Estructura de una consulta
  - **PREFIX** prefijos
  - **SELECT** variables a recuperar
  - **WHERE** condiciones / tripletas
  - Modificadores (ORDER BY, DISTINCT, LIMIT, …)

- Recuperar los usuarios con nombre completo “John Smith”

```
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
SELECT ?x
WHERE {
  ?x vcard:FN "John Smith"
}
```

x
<http://sw/JohnSmith/>

SPARQL – Consultas sencillas

- Recuperar todos los usuarios y sus nombres completos

```
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
SELECT ?x ?fname
WHERE {
  ?x vcard:FN ?fname
}
```

x	fname
<http://sw/RebeccaSmith/>	"Becky Smith"
<http://sw/SarahJones/>	"Sarah Jones"
<http://sw/JohnSmith/>	"John Smith"
<http://sw/MattJones/>	"Matt Jones"

SPARQL – Consultas sencillas

- Recuperar el nombre de pila de los usuarios con apellido “Smith”

```
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
SELECT ?givenName
WHERE {
  ?y vcard:Family “Smith” .
  ?y vcard:Given ?givenName .
}
```

Los tripletes se separan con un punto

La variable ?y no se muestra

givenName
“John”
“Rebecca”

SPARQL – Consultas sencillas

- Recuperar el nombre de pila de los usuarios con apellido “Smith”

```
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
SELECT ?givenName
WHERE {
  ?y vcard:Family “Smith” ;
     vcard:Given ?givenName .
}
```

Podemos usar ; para describir varias relaciones a partir de la misma entidad

givenName
“John”
“Rebecca”

SPARQL – Filtros

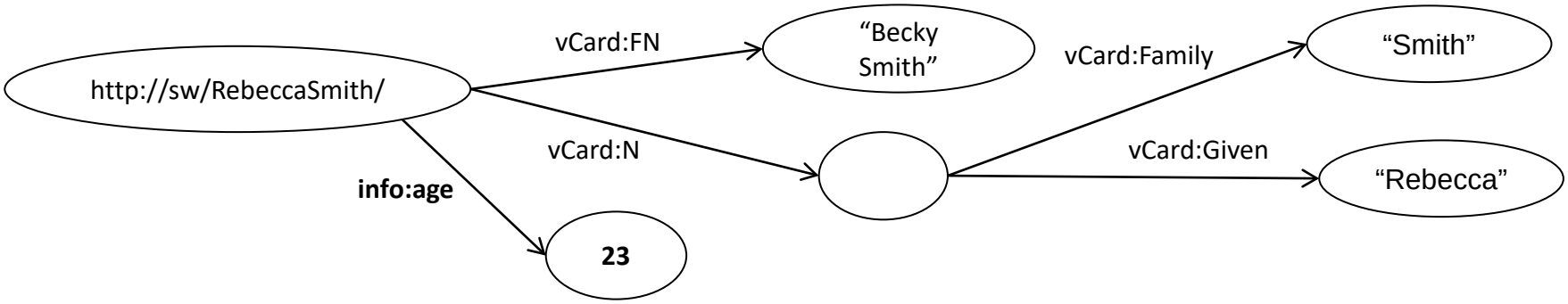
- La palabra **FILTER** permite restringir los resultados de la consulta
- La función *regex(?x, "pattern", "flags")* comprueba si una cadena es reconocida por una expresión regular
  - Los *flags* son opcionales. Por ej, "i" indica *case-insensitive*
- Recuperar nombres de pila que contengan la letra "R" o "r"

```
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
SELECT ?g
WHERE {
  ?y vcard:Given ?g .
  FILTER regex(?g, "r", "i")
}
```

g
"Rebecca"
"Sarah"

SPARQL – Filtros

- Supongamos que modificamos a Rebecca añadiendo su edad:



- Recuperar los usuarios con más de 20 años

```
PREFIX info: <http://somewhere/peopleInfo#>
SELECT ?x
WHERE {
  ?x info:age ?age .
  FILTER (?age > 20)
}
```

x
http://sw/RebeccaSmith/

SPARQL – Contar resultados

- Para poder contar el número de resultados de una consulta usamos la función **COUNT**.

- Contar el número de usuarios con más de 20 años.

```
PREFIX info: <http://somewhere/peopleInfo#>
SELECT (COUNT(DISTINCT(?x) as ?n)
WHERE {
   ?x info:age ?age .
   FILTER (?age > 20)
}
```

n
1

SPARQL – Opcionales

- La palabra OPTIONAL permite describir subgrafos que pueden o no formar parte de la solución

- Encontrar los nombres de todos los usuarios y, si está definida, su edad

```
PREFIX info: <http://somewhere/peopleInfo#>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
SELECT ?name ?age
WHERE {
   ?person vcard:FN ?name .
   OPTIONAL { ?person info:age ?age }
}
```

Sin OPTIONAL sólo se habría recuperado un resultado

name	age
"Becky Smith"	23
"Sarah Jones"	
"John Smith"	
"Matt Jones"	

SPARQL - Uniones

- La palabra **UNION** permite unir las respuestas de varias consultas

- Puede ser útil, por ejemplo si se usan varias propiedades distintas en la base de conocimiento para representar una información:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .

_:a foaf:name "Matt Jones" .
_:b foaf:name "Sarah Jones" .
_:c vcard:FN "Becky Smith" .
_:d vcard:FN "John Smith" .
```

- Este este ejemplo el nombre completo puede aparecer usando las propiedades foaf:name ó vcard:FN

SPARQL - Uniones

- Recuperar los nombres completos de todos los usuarios

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vCard: <http://www.w3.org/2001/vcard-rdf/3.0#>
SELECT ?name
WHERE {
  { [] foaf:name ?name } UNION { [] vCard:FN ?name }
}
```

[] significa nodo en blanco

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vCard: <http://www.w3.org/2001/vcard-rdf/3.0#>
SELECT ?name
WHERE {
  [] ?p ?name
  FILTER ( ?p = foaf:name || ?p = vCard:FN )
}
```

name
"Becky Smith"
"Sarah Jones"
"John Smith"
"Matt Jones"

SPARQL - Modificadores

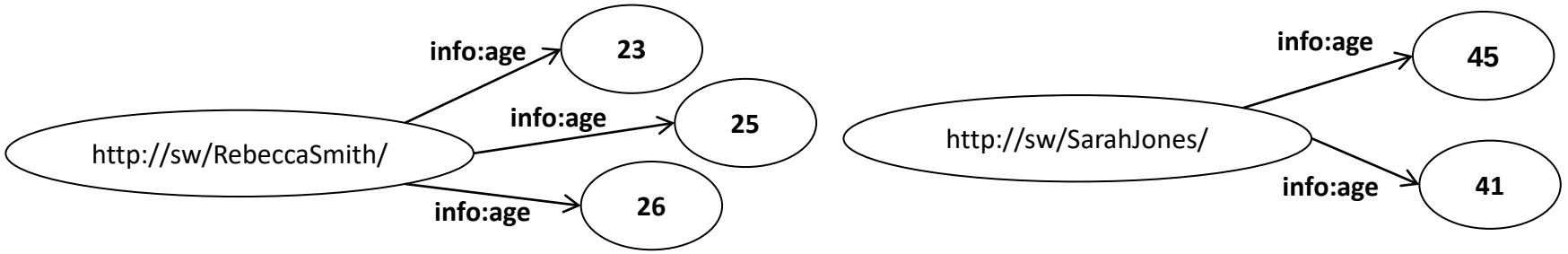
- **LIMIT:** limita el número de resultados a recuperar
- **OFFSET**: indica el primer resultado que se debe mostrar
- **ORDER BY**: orden en el que se muestran los resultados (ASC o DESC, por defecto ASC).
- **DISTINCT**: eliminar resultados duplicados

```
PREFIX vCard: <http://www.w3.org/2001/vcard-rdf/3.0#>
SELECT DISTINCT ?name
WHERE { ?x foaf:name ?name }
ORDER BY ASC(?name)
LIMIT 5
OFFSET 10
```

Recupera los nombres sin duplicados
Los ordena alfabéticamente
Muestra los resultados 11-15 de la lista

SPARQL – Funciones de agregación

- Supongamos que algunos nodos tienen varias edades asignadas (propiedades multivaluadas o inconsistencias en la base de conocimiento)



- Podemos agrupar resultados con propiedades comunes y aplicar funciones sobre cada agregación para calcular un valor "resumen"

```
PREFIX info: <http://somewhere/peopleInfo#>
SELECT ?node (MAX(?age) AS ?maxAge)
WHERE {
  ?node info:age ?age .
}
GROUP BY ?node
ORDER BY ?maxAge
```

node	maxAge
http://sw/RebeccaSmith/	26
http://sw/SarahJones/	45