

Ontologies, Protégé y OWL

Ulises Cortés, Sergio Álvarez, Ignasi Gómez-Sebastià

{ia,salvarez,igomez}@cs.upc.edu

<http://www.cs.upc.edu/~{ia,igomez,salvarez}>

SID2019

<https://kemlg.upc.edu>



Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA



Índice



Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

<https://kemlg.upc.edu>





Indice I

- Ontologías
 - Teoría básica sobre ontologías
- Ontology Web Language (OWL)
 - Introducción a OWL
 - Ciclo de diseño de la ontología
 - Taxonomía y clases
 - Propiedades de datos
 - Propiedades de objetos
 - Restricciones





Indice II

- Protégé como editor de ontologías
 - Introducción
 - Uso del software en contraste con OWL
 - Para practicar
 - u http://protege.stanford.edu/download/protege/3.4/installanywhere/Web_Installers/
- Uso práctico de Protégé



Teoría básica sobre Ontologías

<https://kemlg.upc.edu>



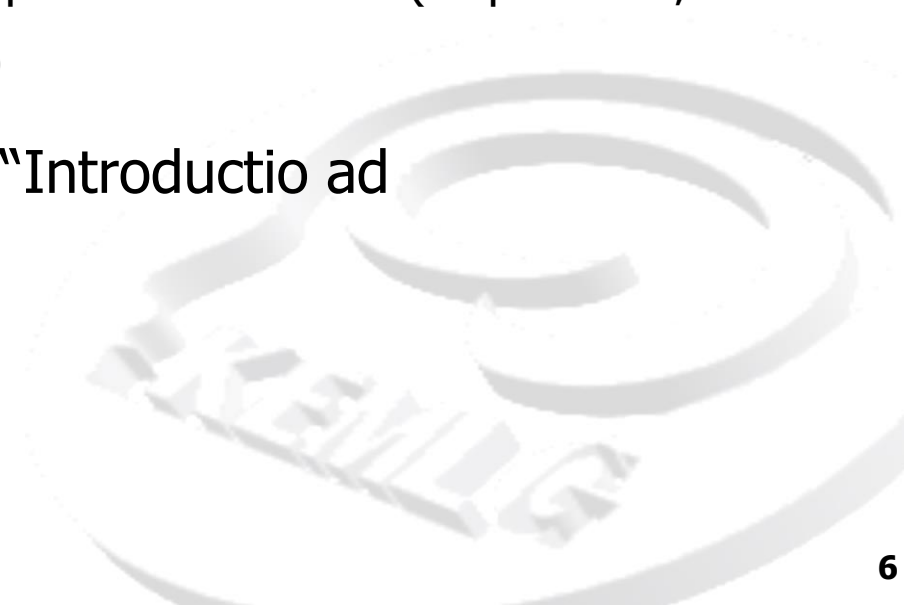
Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA





Filosofía (Aristóteles y Platón)

- Filosóficamente: Parte de la metafísica que estudia lo que hay
 - Lo que existe. Entidades o **conjuntos** de entidades
 - Formas en que se relacionan las entidades que existen
 - u Formas en que se relacionan los universales (Químico) y los particulares a los que se aplica ese universal (Arquímedes, El mago Merlín, Manel Poch)
- Término moderno: Leibniz "Introductio ad Encyclopaediam arcanam"
 - ¿Que hay?
 - El "Todo"





Informàtica

- Esquema conceptual (detallado) de un domino (o varios)
- Facilita la comunicación (intercambio de información) entre sistemas heterogéneos
- Representación del conocimiento. Estructura que contiene todas las entidades y relaciones relevantes para el dominio tratado
 - Cyc (Ontología con conocimiento genérico)
- Uso en Inteligencia Artificial:
 - **Razonamiento**
 - **Clasificación**





Base tecnológica

- DAML
 - Fuertemente asociado a WebServices
 - DARPA
- OIL
 - Comisión Europea
 - Primer lenguaje de intercambio de Ontologías
 - Anticuoado 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 + XML



Ontology Web Language (OWL)

<https://kemlg.upc.edu>



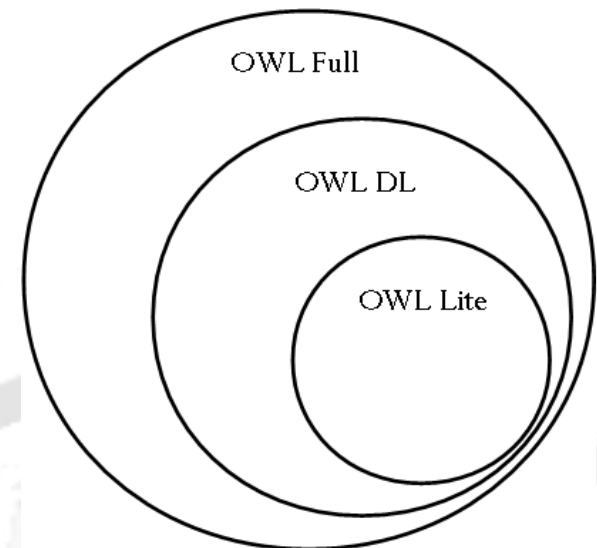
Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA





OWL es como una patata frita

- OWL-FULL
 - Sin restricciones
 - Permite bucles en las relaciones
 - Algunas sentencias pueden llevar tiempo infinito en resolverse
- OWL-DL
 - Termino medio
 - Basado en lógica descriptiva
 - Contiene algunas restricciones
- OWL-LITE
 - Básico
 - Taxonomía con restricciones simples





OWL Lite

- Restricciones de cardinalidad 0..1
- Casi tan complejo como OWL-DL de modo que apenas se usa





OWL DL

- Permite casi todos los tipos de restricciones
 - Restricciones numéricas no soportadas en propiedades transitivas
 - Restricciones en las anotaciones (annotation property)
 - u Sólo soporta datos, literales, URI o Instancias
 - u Una annotation property no puede tener sub-propiedades



Ciclo de desarrollo de una Ontología

<https://kemlg.upc.edu>



Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA





- Cual es el dominio que intentamos cubrir con la Ontología
 - Problema de la granularidad
 - Problema de la omnisciencia
- Para que vamos a usar la Ontología
- Identificar las *Competency Questions*
 - Para que tipos de preguntas debe darnos respuesta la información que contiene la Ontología
- Las decisiones no son finales, pueden cambiar durante el ciclo de desarrollo de la Ontología



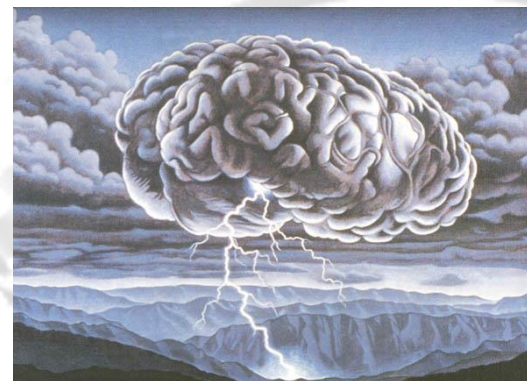


- Porque?
 - Eficiencia, menor coste de desarrollo
 - Integración directa con sistemas que usen esa Ontología
 - Uso de Ontologías que han sido validadas en casos de uso prácticos (aplicaciones)





- Cuales son los términos sobre los que vamos a hablar
- Cuales son las propiedades de esos términos
- Que queremos decir sobre esos términos
- Primer paso:
 - No organizar los términos, hacer una lista con lo que queremos incluir en la Ontología





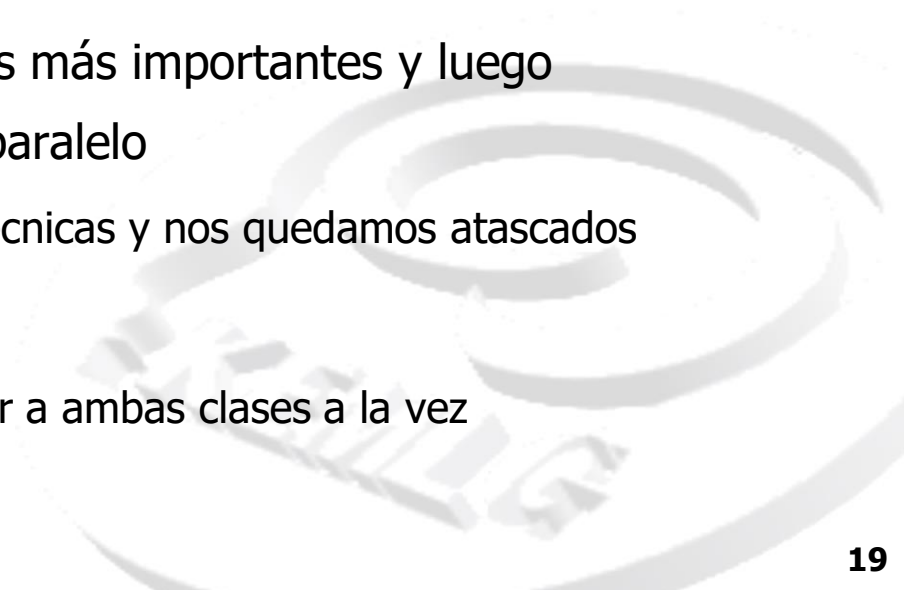
- Definir las clases y la taxonomía
 - Una clase es un concepto del dominio, no un objeto
 - u No sólo entidades, pueden ser propiedades!
 - Una clase es un conjunto de elementos con propiedades similares
 - u Y que es cada elemento entonces?
- La taxonomía es la jerarquía de clases
 - Cuando agrupamos dos clases en la misma superclase?
 - u La respuesta la tenéis en esta transparencia







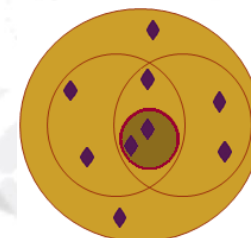
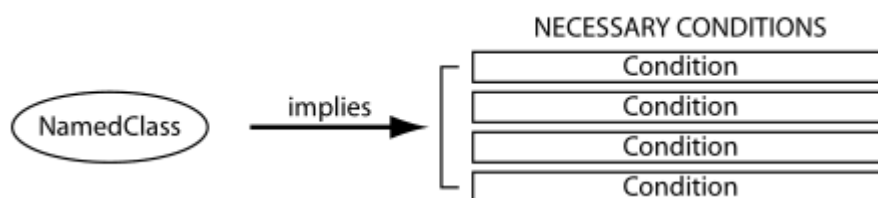
- Definir las clases y la taxonomía
 - Top-Down: Definir primero los conceptos más generales, y luego especializar
 - u Y cual es el concepto más general?
- Bottom-up: Definir los conceptos más específicos y luego agruparlos en clases más generales
 - Y cuando los agrupamos?
- Combinación: Definir los conceptos más importantes y luego generalizarlos y especificarlos en paralelo
 - Útil si aplicamos otra de las dos técnicas y nos quedamos atascados
- Clases disjuntas
 - Una instancia no puede pertenecer a ambas clases a la vez



Tipos de clases: Primitiva

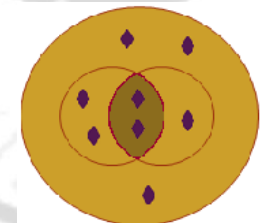
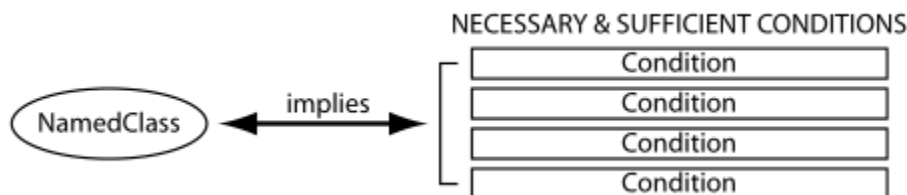
- *Necessary conditions*

- Si algo reúne las condiciones no es necesariamente obligatorio que sea un miembro de la clase
- **PERO:** Un elemento escogido al azar que sabemos que es miembro de la clase, sabemos que reúne las condiciones
- Que ocurre con un elemento escogido al azar que sabemos que reúne las condiciones?



Tipos de clases: Equivalente

- *Necessary conditions and sufficient conditions*
 - *Si algo reúne las condiciones es suficiente para decir que es un miembro de la clase*
 - *Un elemento escogido al azar que sabemos que es miembro de la clase, sabemos que reúne las condiciones*
 - *Que ocurre con un elemento escogido al azar que sabemos que reúne las condiciones?*





- Asociadas a la clases (Dominio-Rango):
- Rango no es una clase
 - Data Properties
- Rango es una clase
 - Relaciones a otras instancias de la clase
 - Object Properties



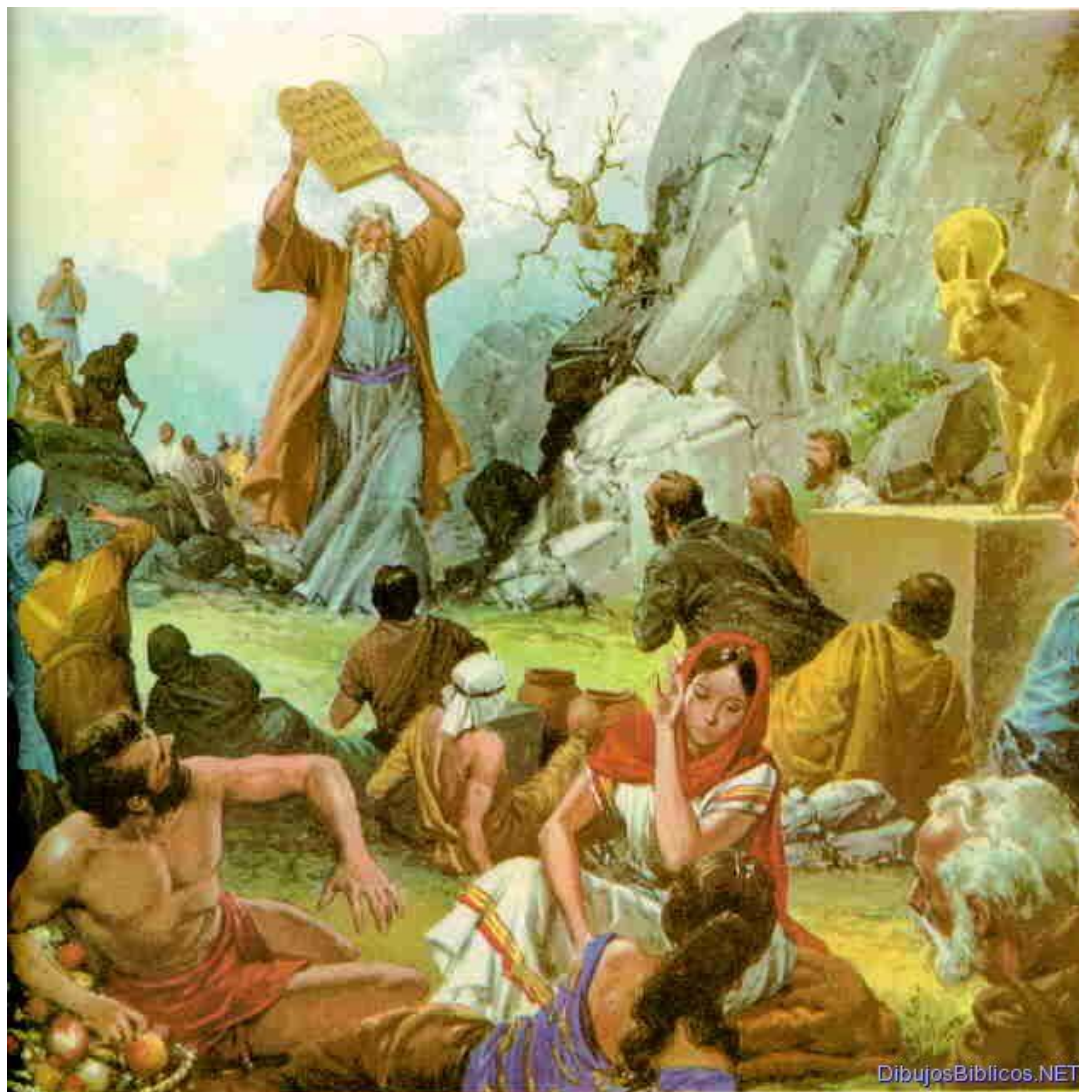


- Las restricciones definen el conjunto de valores posibles para una propiedad
- Las restricciones más comunes son
 - Dominio
 - Rango
- No son restricciones a comprobar, son axiomas





- Moisés y los creadores de Ontologías inconsistentes



DibujosBiblicos.NET



- Crear instancias de las clases
 - La clase se convierte en un tipo directo de la instancia
 - Las superclases del tipo directo son tipos de la instancias
- Asignar valores a las propiedades
 - Los valores asignados deben cumplir las restricciones impuestas
 - Se pueden usar razonadores para comprobar que las restricciones se cumplan



Restricciones



Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

<https://kemlg.upc.edu>



Restricciones: Quantifier restrictions

- *Existencial*
 - *Una pizza picante es una pizza que contiene al menos un ingrediente picante*
 - *Some*
- *Universal*
 - *Una pizza vegana es una pizza donde todos los ingredientes son vegetales*
 - *Only*





Restricciones: Cardinality restrictions

- Mínimo número de relaciones
 - Pizza Beverly Hills
- Máximo número de relaciones
 - Pizza de Oferta
- Número exacto de relaciones
 - Pizza Binaria
- Restricciones qualificadas
 - Limitan el rango de la relación
 - Pizza triqueseada





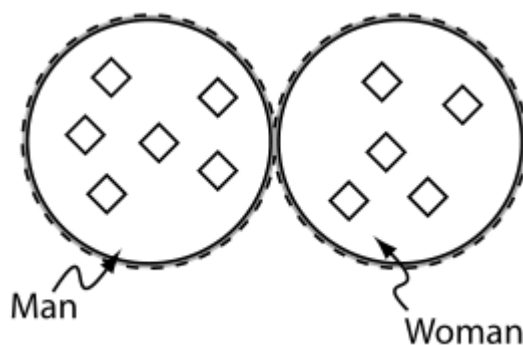
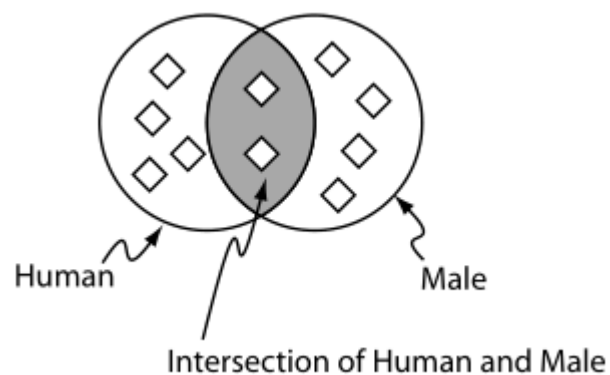
Restricciones: por valor

- hasValue restrictions
 - Equivalente a enumeraciones
 - Usa el símbolo \exists
 - u Pizza EuroMediterranea =
 - u Pizza.tienePaisOrigen \exists {Italia, España, Grecia, Francia}





Restricciones: operaciones conjuntos



Propiedades



Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

<https://kemlg.upc.edu>





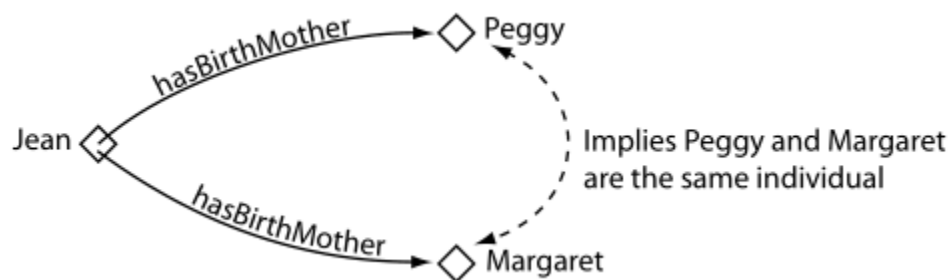
Propiedades

- Propiedades y subpropiedades
 - Cuando agrupamos propiedades en subpropiedades?
- Domino y rango de la propiedad
 - Recordad que son tratadas como axiomas!
- Propiedad Inversa
 - Del tipo *hasComponent* *isComponentOf*
 - Donde el dominio y el rango, se intercambian



Propiedades

- Propiedad funcional
 - Cuando A y B están relacionados mediante una propiedad funcional sólo una instancia de B puede estar relacionada con A
 - u Que ocurre si más de una instancia de B está relacionada con A?

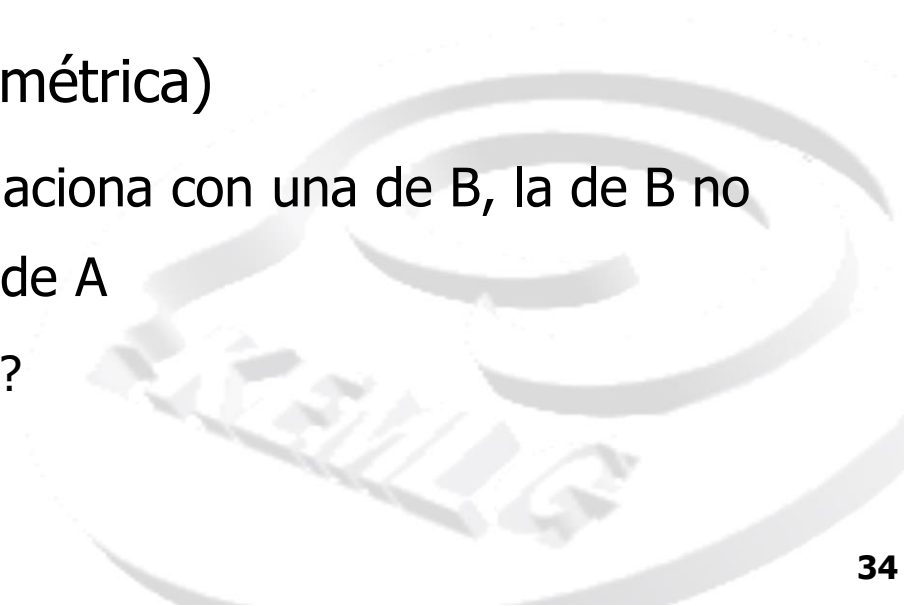


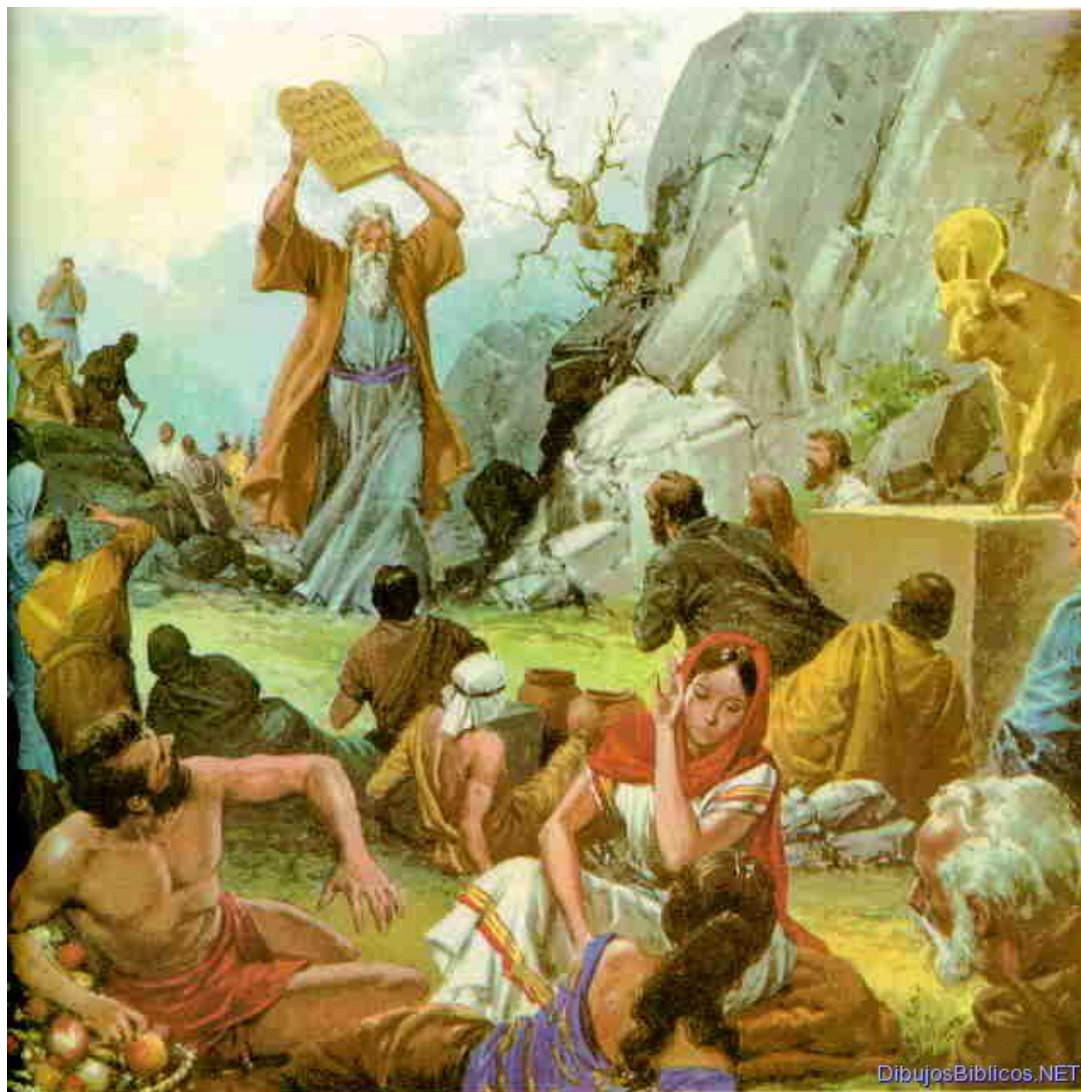
- Propiedad funcional inversa



Propiedades

- Propiedad transitiva
 - Si una instancia de A se relaciona con una de B y una de B con C, la instancia de A se relaciona con la de C
- Propiedad simétrica
 - Si una instancia de A se relaciona con una de B, la de B se relaciona con la de A
- Propiedad asimétrica (antisimétrica)
 - Si una instancia de A se relaciona con una de B, la de B no se puede relacionar con la de A
 - u Y que pasa si se relaciona?





- Moisés y los creadores de Ontologías inconsistentes



Propiedades

- Propiedad reflexiva
 - Si P es reflexiva y una instancia A tiene P , P se aplica a la instancia de A
- Propiedad irreflexiva
 - Si P es reflexiva y una instancia A tiene P , P no se puede aplicar a la instancia de A
 - u Y que pasa si se aplica?





Propiedades

- Data properties
 - Tienen sentido todas las restricciones aplicadas a propiedades en las data properties?
 - Si es que no, cuales tienen sentido?



Protégé



Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

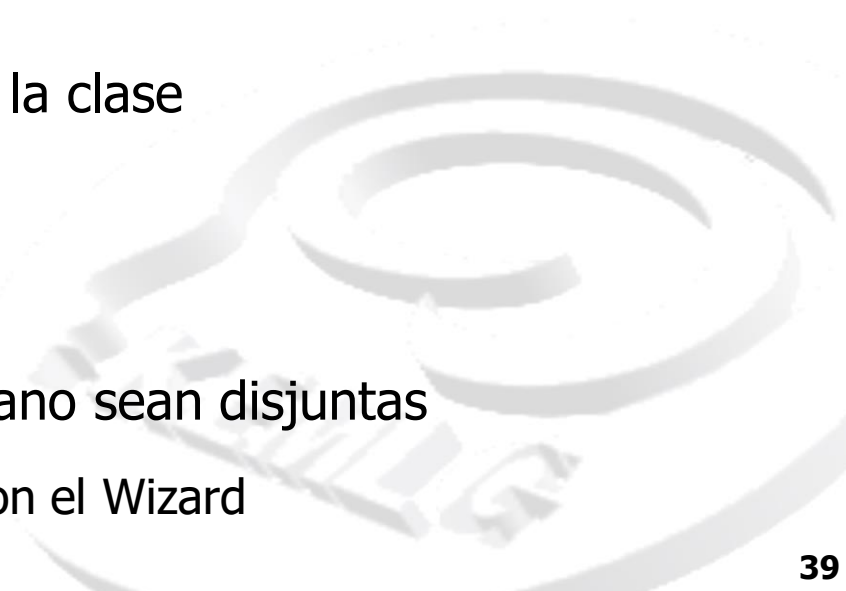
<https://kemlg.upc.edu>





Protégé

- Crear una Ontología nueva
 - De que tipo?
 - Save project as y volver a abrir
 - Añadir comentario a la Ontología
- Crear clases
 - Crear una subclase de Thing
 - Crear hermano y subclase de la clase
 - Crear otro hermano
 - Borrar el segundo hermano
 - Hacer que la clase y su hermano sean disjuntas
 - u Creación masiva de clases con el Wizard





Protégé

- Propiedades (Object Properties)
 - Crear una propiedad
 - Crear una subpropiedad
 - Asignar Dominio
 - Asignar Rango
 - Asignar restricciones a las propiedades
- Propiedades (Data Properties)
 - Crear propiedad
 - Asignar Dominio
 - Asignar Rango
 - Asignar Rango con conjunto de valores permitidos





Protégé

- Crear Individuals
- Restricciones en clases
 - Asignar Necessary Conditions a una clase
 - Asignar Necessary and Sufficient Conditions a una clase
- Usando el razonador para comprobar la consistencia y clasificar
- Añadir instancias



Uso práctico de Protégé

<https://kemlg.upc.edu>



Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA





Usando la ontología de Pizzas

- Uso del razonador para comprobar la consistencia y clasificar.
 - Es la ontología consistente?
 - Se ha clasificado algo?
- Crear una instancia que se vaya a clasificar (CheesyPizza) y volver a ejecutar el razonador.
 - Mirar si se infiere algo en las instancias.



Usando la ontología de Pizzas

- Super mario pizza
 - Crear el topping tortuga
 - Crear una named pizza "SUPER MARIO PIZZA" que tiene como ingredientes champiñones y tortugas. Especificar como necessary and sufficient condition. Crear instancias de la pizza ejecutar el razonador y ver como se clasifica





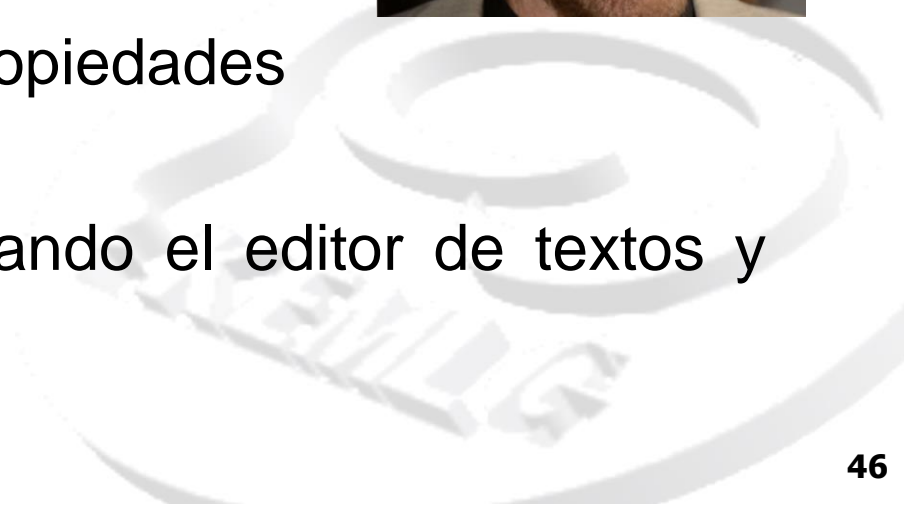
Usando la ontología de Pizzas

- Crear una necessary condition que nos indique que una pizza tiene una (y sólo una!) base de pizza.
 - Que quiere decir esto?
 - Que ocurre si creamos un individual con dos bases de pizza?



Usando la ontología de Pizzas

- OWL de Chuck Norris
 - Editar con un editor de texto plano
 - Buscar los diferentes elementos para ver que pinta tienen
 - Classes
 - Propiedades (Object)
 - Propiedades (Data)
 - Restricciones de clases
 - Restricciones de propiedades
 - Instancias
 - Replicar instancias usando el editor de textos y verlas en protege



Referencias

<https://kemlg.upc.edu>



Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA





- Cyc
 - <http://www.cyc.com/>
 - <http://www.opencyc.org/>
- DAML
 - <http://www.daml.org/>
- OIL
 - <http://www.cs.vu.nl/~frankh/postscript/IEEE-IS01.pdf>
- RDF
 - <http://www.xml.com/pub/a/2001/01/24/rdf.html>
- **OWL**
 - <http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/>



- Protégé
 - <http://protege.stanford.edu>
- Ontologia de pizzas
 - www.co-ode.org/ontologies/pizza/





Knowledge Engineering and Machine Learning Group UNIVERSITAT POLITÈCNICA DE CATALUNYA

Ulises Cortés, Sergio Álvarez, Ignasi Gómez-Sebastià

{ia,salvarez,igomez}@cs.upc.edu

<http://www.cs.upc.edu/~{ia,igomez,salvarez}>

SID2019