

FIN DE LA PRIMERA SESION

SEGUNDA SESION

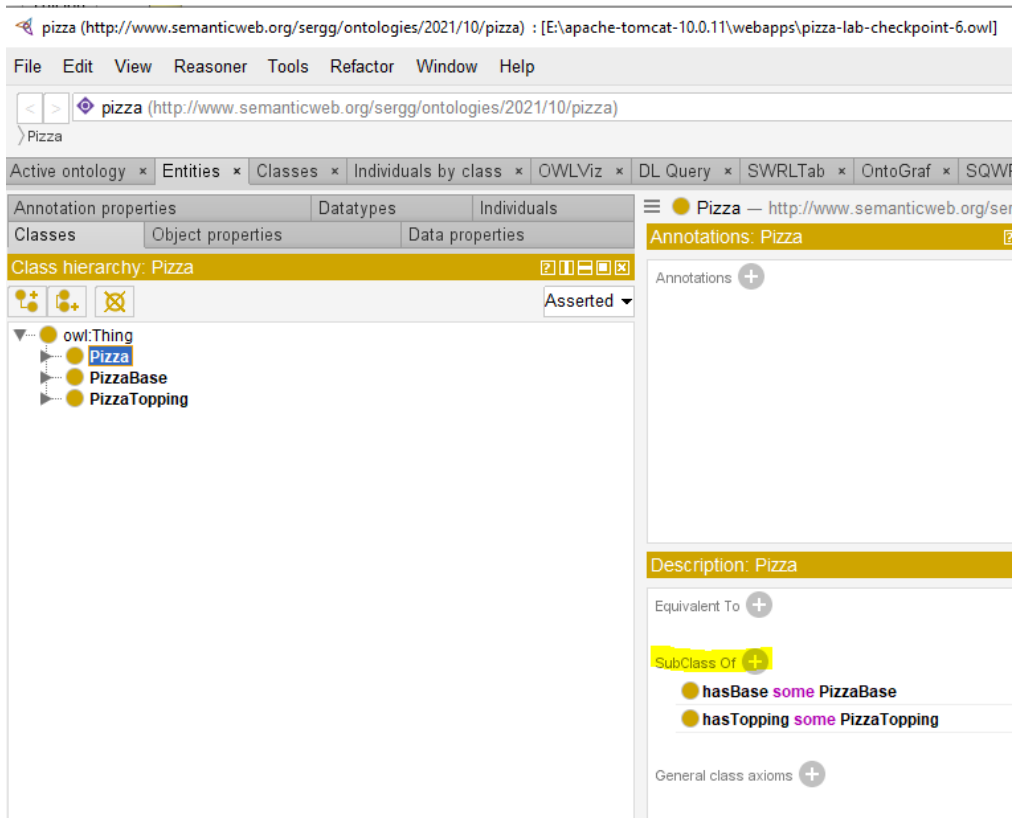
Sesión 2 - Agenda

- Definiciones de clases más completas
- Funcionamiento de razonador
- Clases Definidas y Primitivas
- Clases Enumeradas
- Restricciones sobre propiedades y datos
- Definición de propiedades transitivas

Definiciones de clases más completas

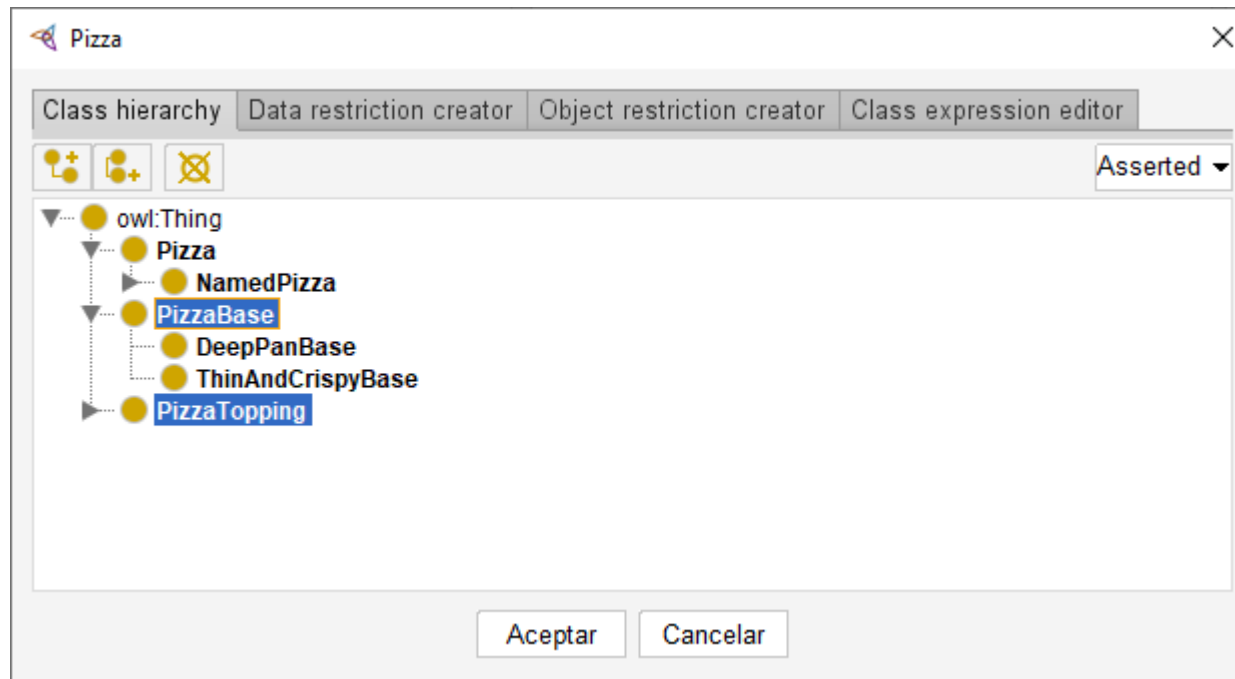
Mecanismos para especificar restricciones para una subclase (I)

- En el menú de clases, al hacer clic en el + de Subclass Of, se abre el siguiente menú, con 4 pestañas (ver página siguiente)



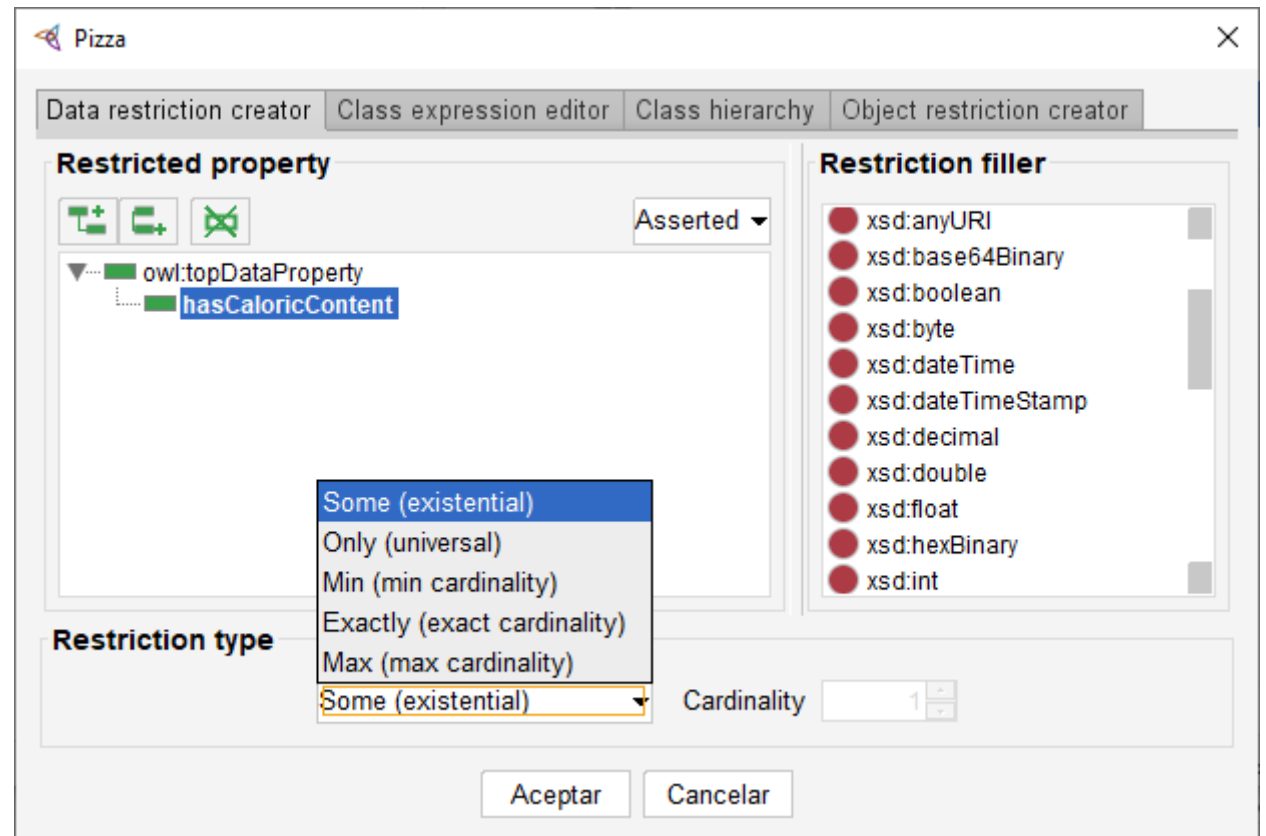
Mecanismos para especificar restricciones para una subclase (II)

- Class hierarchy, permite seleccionar una o más clases para elaborar la restricción de que la subclase deberá heredar de esas clases.



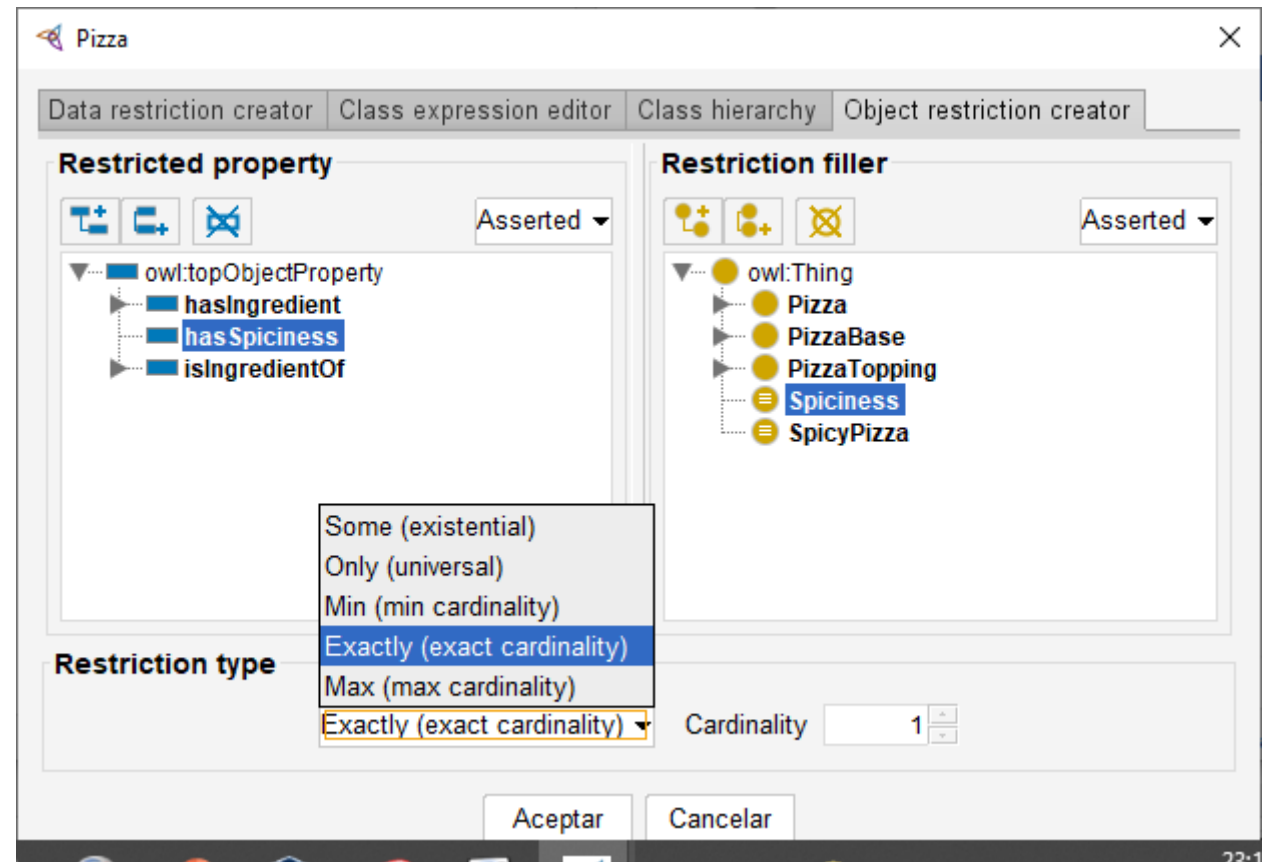
Mecanismos para especificar restricciones para una subclase (III)

- Data restriction creator: permite establecer una restricción sobre datos.
- Pueden ser de varios tipos:
 - Existencial:
 - Some
 - Universal:
 - Only
 - De Cardinalidad:
 - Min
 - Exactly
 - Max



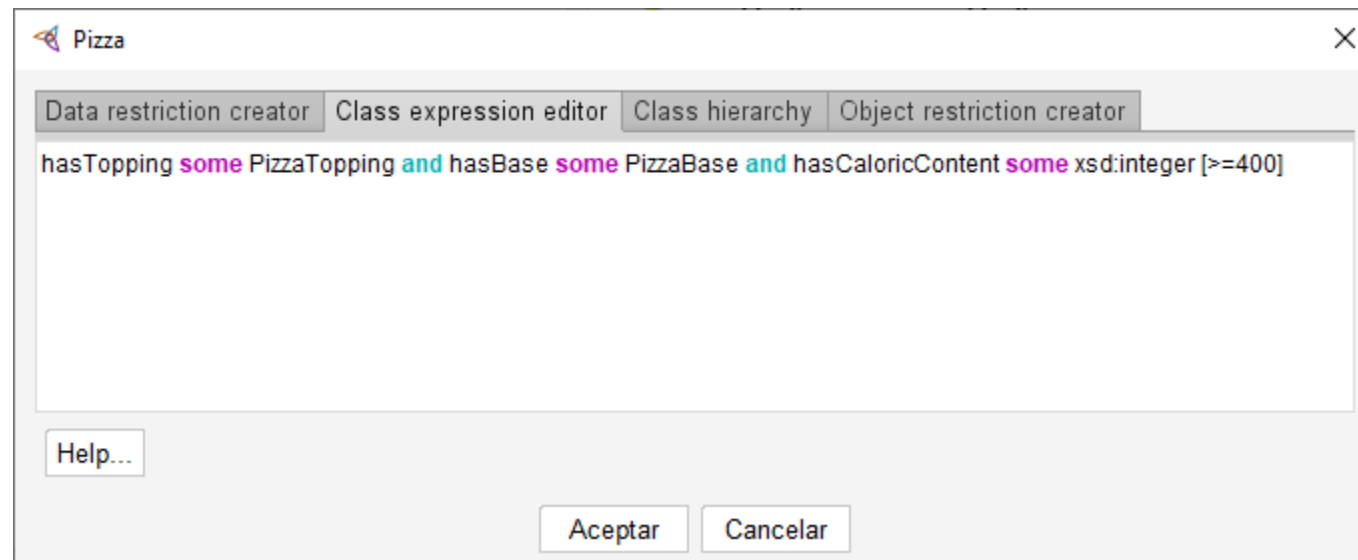
Mecanismos para especificar restricciones para una subclase (IV)

- Object restriction creator: permite establecer una restricción sobre propiedades de objetos.
- Pueden ser de varios tipos:
 - Existencial:
 - Some
 - Universal:
 - Only
 - De Cardinalidad:
 - Min
 - Exactly
 - Max



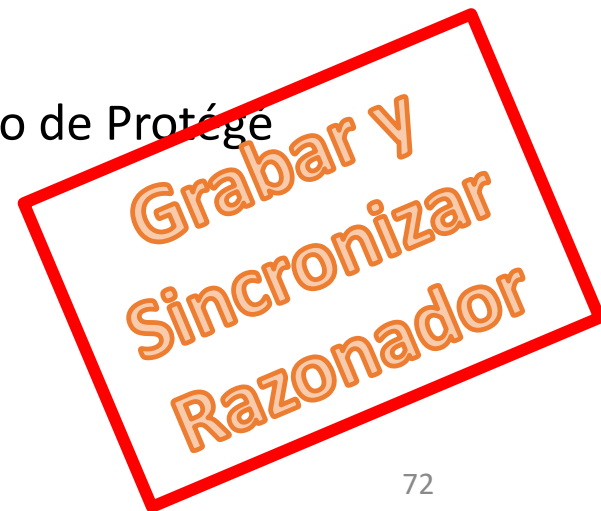
Mecanismos para especificar restricciones para una subclase (V)

- Class expression editor: permite elaborar reglas más completas usando descripción lógica.
- Aviso: para conocer más acerca de la sintaxis de las expresiones que podemos incluir, hacer clic en el botón ayuda o ir a <http://protegeproject.github.io/protege/class-expression-syntax/>



Creación de Subclases de pizza

- Creamos una jerarquía de clases bajo Pizza (ver siguiente página)
- Clase NamedPizza como subclase de Pizza (que representará a pizzas con un nombre conocido que describiremos en la anotación)
- A continuación, creamos la clase MargheritaPizza como subclase de NamedPizza (añadir anotación con un texto descriptivo)
- Seleccionando la clase MargheritaPizza, Seleccionar Subclass Of +
 - Class expression
 - hasTopping some Mo... (<CTRL>+<Space>) → para usar el autorrelleno de Protégé
 - ...MozzarellaTopping
 - Repetir con TomatoTopping
- Observar en el interfaz de Protégé las inferencias



pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-5.owl]

File Edit View Reasoner Tools Refactor

< > pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-5.owl]

> Pizza > NamedPizza > MargheritaPizza

Active ontology x Entities x Classes x Individuals

Annotation properties Datatypes

Classes Object properties

Class hierarchy: MargheritaPizza

owl:Thing

- PizzaTopping
- PizzaBase
 - ThinAndCrispyBase
 - DeepPanBase
- Pizza
 - NamedPizza
 - MargheritaPizza

MargheritaPizza

Data restriction creator Class expression editor Object restriction creator Class hierarchy

hasTopping some MozzarellaTopping

Help...

Aceptar Cancelar

SubClass Of +

- NamedPizza

General class axioms +

SubClass Of (Anonymous Ancestor)

- hasBase some PizzaBase
- hasTopping some PizzaTopping

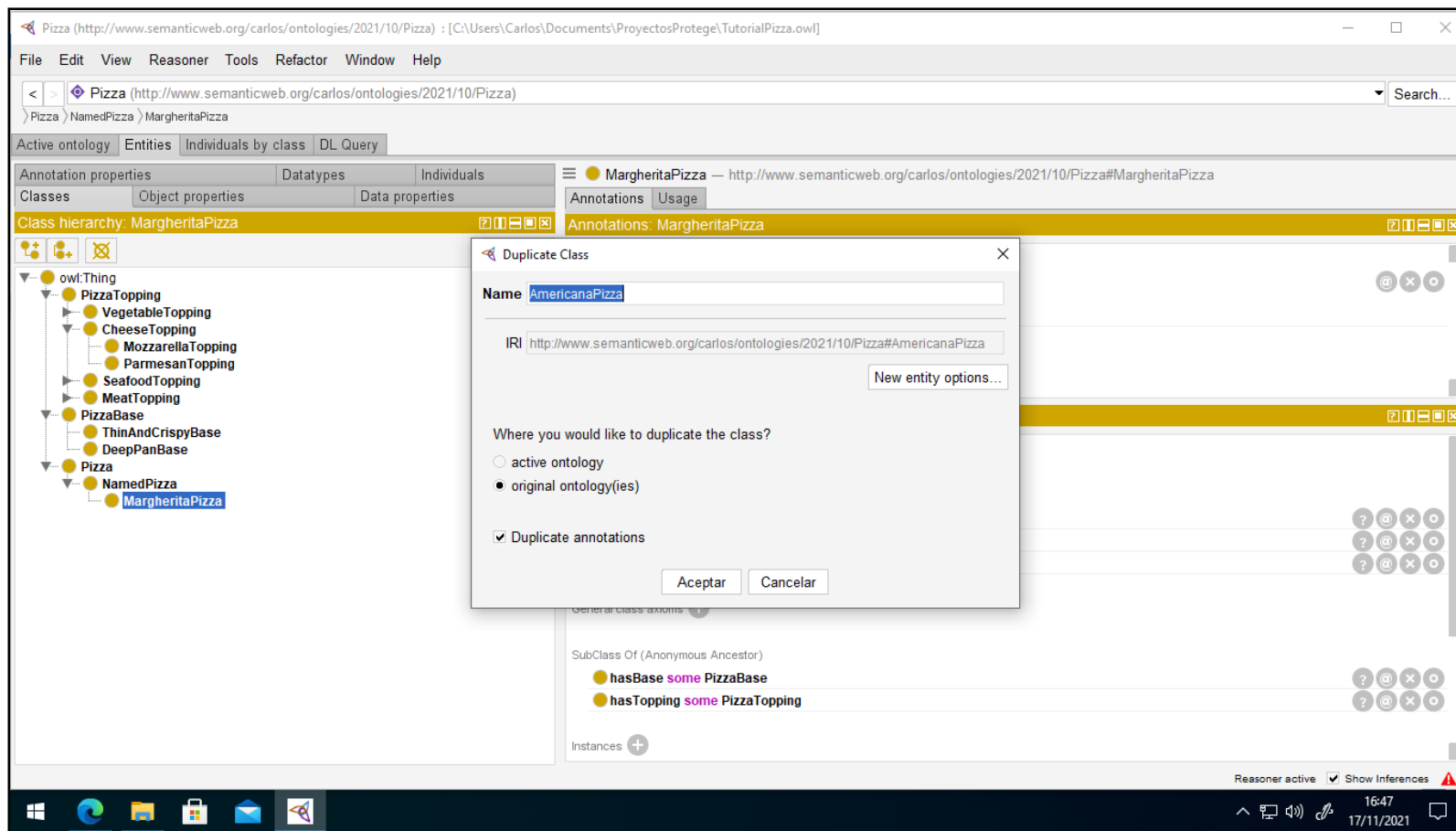
Instances +

Target for Key +

Clonar clases

- Creamos la clase AmericanaPizza clonando la clase Margherita y añadiendo las restricciones adicionales Seleccionando la clase MargheritaPizza. *Right Click > Duplicate class (ver página siguiente)*
 - Name: AmericanaPizza
- Seleccionando la clase AmericanaPizza y añadimos una restricción adicional
 - SubClass Of +
 - Class expression
 - hasTopping some PepperoniTopping
- Añadir una anotación a la clase con la descripción de esta pizza (*ver dos páginas más adelante*)

Grabar y
Sincronizar
Razonador



pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-5.owl]

File Edit View Reasoner Tools Refactor Window Help

< > pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) Search...

Pizza > NamedPizza > AmericanaPizza

Active ontology x Entities x Classes x Individuals by class x OWLViz x DL Query x OntoGraf x SWRLTab x SQWRLTab x

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Annotations Usage

Class hierarchy: AmericanaPizza Annotations: AmericanaPizza

Asserted

- owl:Thing
 - PizzaTopping
 - VegetableTopping
 - CheeseTopping
 - MozzarellaTopping
 - ParmesanTopping
 - SeafoodTopping
 - AnchovyTopping
 - PrawnTopping
 - TunaTopping
 - MeatTopping
 - HamTopping
 - PepperoniTopping
 - SalamiTopping
 - SpicyBeefTopping
 - PizzaBase
 - ThinAndCrispyBase
 - DeepPanBase
 - Pizza
 - NamedPizza
 - AmericanaPizza
 - MargheritaPizza

AmericanaPizza

Literal Entity IRI IRI Editor Property values

AmericanaPizza is defined by Mozzarella, Tomato and Pepperoni toppings

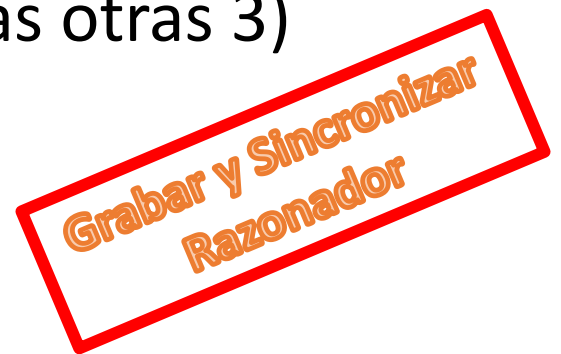
Type xsd:string Lang

Aceptar Cancelar

Reasoner state out of sync with active ontology ☒ Show Inferences

Crear más clases

- Creamos la clase *AmericanaHotPizza* duplicando *AmericanaPizza* y añadiendo *JalapenoPepperTopping*
- Creamos *SohoPizza* clonando *MargheritaPizza* y añadiendo *OliveTopping* y *ParmesanTopping*
- A continuación hacer que todas las subclases de *NamedPizza* sean Disjuntas (a partir de cualquier de ellas, seleccionando las otras 3)



Funcionamiento del Razonador

Detectando clases que no pueden tener miembros

- Cuando el razonador detecta que una clase se ha definido de manera que no es posible crear ninguna instancia, el razonador, definirá una equivalencia a owl:Nothing. Seguramente se debe a un error a la hora de definir la ontología.

Detección de Inconsistencias

- Creamos una clase inconsistente para demostrar el funcionamiento del razonador:
- Desde CheeseTopping crear subclase ProbelInconsistentTopping
- Seleccionar la clase ProbelInconsistentTopping y añadir una restricción, dando a Subclass of +.
 - Desde Class hierarchy tab seleccionar: VegetableTopping
- Resincronizar el razonador
- Mostrar la explicación de por qué se deduce que no puede tener objetos owl:Nothing



pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-6.owl]

File Edit View Reasoner Tools Refactor Window Help

< > pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza)

PizzaTopping > CheeseTopping > ProbelInconsistenceTopping

Active ontology x Entities x Classes x Individuals b

Annotation properties Datatypes

Classes Object properties Data

Class hierarchy: ProbelInconsistenceTopping

owl:Thing

- PizzaTopping
 - VegetableTopping
 - CheeseTopping
 - ProbelInconsistenceTopping
 - MozzarellaTopping
 - ParmesanTopping
 - SeafoodTopping
 - AnchovyTopping
 - PrawnTopping
 - TunaTopping
 - MeatTopping
 - HamTopping
 - PepperoniTopping
 - SalamiTopping
 - SpicyBeefTopping
- PizzaBase
 - ThinAndCrispyBase
 - DeepPanBase
- Pizza
 - NamedPizza
 - SohoPizza
 - AmericanaHotPizza
 - AmericanaPizza
 - MargheritaPizza

ProbelInconsistenceTopping

Data restriction creator Class expression editor Object restriction creator Class hierarchy

owl:Thing

- Pizza
- PizzaBase
- PizzaTopping
 - CheeseTopping
 - MeatTopping
 - SeafoodTopping
 - VegetableTopping

Aceptar Cancelar

Description: ProbelInconsistenceTopping

owl:Nothing

SubClass Of +

CheeseTopping

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

Disjoint Union Of +

Reasoner state out of sync

The screenshot shows the Protege OWL editor interface. The left pane displays a class hierarchy starting from `owl:Thing`, with `PizzaTopping` as a parent class. Under `PizzaTopping`, there are subclasses like `VegetableTopping`, `CheeseTopping`, and `ProbInconsistenceTopping`. The `ProbInconsistenceTopping` class is highlighted, indicating it is the focus of the current view. The right pane shows the 'Annotations' and 'Description' for `ProbInconsistenceTopping`. The 'Description' pane shows that `ProbInconsistenceTopping` is a subclass of `owl:Nothing`, which is represented by a red circle with a question mark. The 'Annotations' pane shows that `ProbInconsistenceTopping` is a subclass of `CheeseTopping` and `VegetableTopping`.

The screenshot shows a dialog box titled 'Explanation for ProbInconsistenceTopping EquivalentTo owl:Nothing'. It contains two radio buttons: 'Show regular justifications' (selected) and 'Show laconic justifications'. There is also a 'Limit justifications to' field set to 2. Below the radio buttons, there is a section for 'Explanation 1' with a checkbox for 'Display laconic explanation'. The explanation text is as follows:

- ProbInconsistenceTopping SubClassOf CheeseTopping
- ProbInconsistenceTopping SubClassOf VegetableTopping
- DisjointClasses: CheeseTopping, MeatTopping, SeafoodTopping, VegetableTopping

- Cuando el razonador detecta que una clase se ha definido de manera que no es posible crear ninguna instancia, el razonador, definirá una equivalencia a `owl:Nothing`

Clases Primitivas y Definidas

Describir y Definir Clases

Las Propiedades, nos permiten completar la definición de las clases:

- Clases **Primitivas**: Definen condiciones necesarias para formar parte de ella (Pizza)
- Clases **Definidas**: Definen condiciones necesarias y suficientes para confirmar que se forma parte de ella.
- Clases **Anónimas**: Son clases auxiliares que genera el razonador en casos en que las necesite para hacer inferencias.

Clases Primitivas y Definidas

- Necesario significa: Si algo es miembro de una clase, entonces es requisito que cumpla las condiciones indicadas.
- Necesario NO significa: si algo cumple estas condiciones, entonces es miembro de esa clase
- Suficiente significa: si se cumplen estas condiciones, entonces es miembro de esa clase.

Creación de clases primitivas y definidas

Creamos CheesyPizza (Primitiva):

- Creamos clase CheesyPizza como subclase de Pizza
- Seleccionando CheesyPizza, añadimos la restricción SubClass Of +
 - hasTopping some CheeseTopping

Resumen: La descripción de CheesyPizza indica que para formar parte de esta clase, tiene que ser Pizza y tiene que tener al menos un topping del tipo CheeseTopping (condiciones necesarias, pero no suficientes, porque la definición de la clase no está cerrada: no es una clase definida)



Creación de clases primitivas y definidas

Convertimos CheesyPizza en clase Definida: (ver página siguiente)

Grabar y
Sincronizar
Razonador

- Seleccionamos CheesyPizza -> Edit -> Convert to defined class
 - A partir de ahora, veremos la clase con 3 líneas en la vista de clases
 - Aparece “**Equivalent To**”, que define las condiciones necesarias y suficientes para formar parte de la clase CheesyPizza
 - Gracias al razonador, en la vista Class Hierarchy (Inferred) veremos como se han categorizado elementos bajo CheesyPizza que no estaban antes (ver dos páginas más adelante)
 - AmericanaHotPizza, AmericanaPizza, MargheritaPizza, SohoPizza (porque tienen topping MozzarellaTopping o ParmesanTopping)

Resumen: La descripción de CheesyPizza indica que para formar parte de esta clase, tiene que ser Pizza y tiene que tener al menos un topping del tipo CheeseTopping, como condiciones necesarias y suficientes. Esto queda fijado en la sección Equivalent To.

Interface of the Semantic Web Editor (SWE) showing the ontology 'pizza' (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) loaded. The left pane displays the class hierarchy, including 'CheesyPizza' and its subclasses: 'NamedPizza', 'AmericanaHotPizza', 'AmericanaPizza', 'MargheritaPizza', 'SohoPizza', 'PizzaBase', and 'PizzaTopping'. The right pane shows the 'Description: CheesyPizza' section, which lists the following axioms:

- Equivalent To: **Pizza**
- SubClass Of: **hasTopping some CheeseTopping**
- SubClass Of (Anonymous Ancestor): **hasTopping some PizzaTopping**, **hasBase some PizzaBase**

The 'Instances' section is empty. The 'Reasoner active' checkbox is checked, and 'Show Inferences' is also checked.

Interface of the Semantic Web Editor (SWE) showing the ontology 'pizza' (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) loaded. The left pane displays the class hierarchy, including 'CheesyPizza' and its subclasses: 'NamedPizza', 'AmericanaHotPizza', 'AmericanaPizza', 'MargheritaPizza', 'SohoPizza', 'PizzaBase', and 'PizzaTopping'. The right pane shows the 'Description: CheesyPizza' section, which lists the following axioms:

- Equivalent To: **Pizza and (hasTopping some CheeseTopping)**
- SubClass Of: **Pizza**
- SubClass Of (Anonymous Ancestor): **hasTopping some PizzaTopping**, **hasBase some PizzaBase**

The 'Instances' section is empty. The 'Reasoner active' checkbox is checked, and 'Show Inferences' is also checked. A blue arrow points from the 'Pizza' axiom in the left pane to the 'Pizza' axiom in the right pane.

Conversión de clase Primitiva a Clase Definida

Cambiar “Asserted” (afirmado) por “Inferred” (inferido)

The screenshot displays the Protege OWL editor interface for the 'pizza' ontology. The main window shows the 'Class hierarchy: CheesyPizza' panel on the left, which lists the hierarchy of classes. The 'CheesyPizza' class is highlighted, and its 'Inferred' status is shown in a dropdown menu, which is circled in red. The 'Description: CheesyPizza' panel on the right shows the class's axioms, including 'Equivalent To', 'SubClass Of', 'General class axioms', 'SubClass Of (Anonymous Ancestor)', 'Instances', 'Target for Key', and 'Disjoint With'. The 'Usage: CheesyPizza' panel on the right shows the class's usage in the ontology, including its role in the 'CheesyPizza' class and its relationship to the 'Pizza' class.

Active ontology: pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-8.owl]

File Edit View Reasoner Tools Refactor Window Help

Search...

Active ontology: pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza)

Entities Individuals by class DL Query

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Class hierarchy: CheesyPizza

Inferred

Annotations: CheesyPizza

Usage: CheesyPizza

Show: this disjoints named sub/superclasses

Found 4 uses of CheesyPizza

CheesyPizza

Class: CheesyPizza

CheesyPizza EquivalentTo Pizza and (hasTopping some CheeseTopping)

Equivalent To

Pizza and (hasTopping some CheeseTopping)

SubClass Of

Pizza

General class axioms

SubClass Of (Anonymous Ancestor)

hasTopping some PizzaTopping

hasBase some PizzaBase

Instances

Target for Key

Disjoint With

Reasoner active Show Inferences

Restricción Universal (para todo...)

- Hasta ahora, hemos usado restricciones existenciales (*CheesyPizza*, si es *Pizza* y tiene un *CheesyTopping*)
- Ahora usaremos restricciones universales, para crear una clase que incluya únicamente aquellas pizzas que tienen ingredientes vegetarianos, y solo vegetarianos.

Creamos una Clase usando restricción Universal

- Seleccionamos Pizza y creamos una subclase: VegetarianPizza
- Seleccionamos VegetarianPizza y restringimos con Subclass Of +
- Class expression editor.... ¿Qué axioma *de lógica descriptiva* usaremos?

- *PIENSA UNA SOLUCION ANTES DE IR A LA SIGUIENTE PÁGINA*

- Dato: Las clases que contienen ingredientes no vegetarianos son las que tienen topping de queso o vegetal.
- Pista: Debe usarse “only”

Creamos una Clase usando restricción Universal

- Seleccionamos la clase Pizza y creamos una subclase: VegetarianPizza
- Seleccionamos VegetarianPizza y restringimos con Subclass Of +
- Class expression editor.... ¿Qué axioma de lógica descriptiva usaremos?
 - hasTopping only (VegetableTopping or CheeseTopping)
- Convertimos la clase VegetarianPizza en clase Definida.



Comprobamos el resultado...

- Encontramos algo extraño (para los humanos) en las clases inferidas
 - ¿Dónde deberían estar las clases MargheritaPizza y SohoPizza)?
 - ¿Por qué no están clasificadas como VegetarianPizza?

OWA vs CWA

- OWA (Open World Assumption) – Suposición de mundo abierto
 - Si no se ha definido alguna información, ésta puede existir, es decir que la definición proporcionada permanece abierta.
 - OWL considera que la información puede no estar integrada aún en nuestra ontología.
- CWA (Closed World Assumption)- Suposición de mundo cerrado
 - Si no se ha definido alguna información, ésta no existe, es decir, que la definición proporcionada es completa.
 - ¡¡Más común en los lenguajes de programación y en lenguajes de representación del conocimiento!!

El razonador supone que las pizzas Margherita y Soho podrían tener otros toppings no vegetarianos, dado que no se ha definido que únicamente pertenezcan a esas clases los individuos que tengan esos toppings, sino que son necesarios para formar parte de la clase, aunque podrían tener más, de manera que quedarían fuera de la clase Vegetariana y seguirían siendo Margherita o Soho.

Axioma de Cierre de clase

- Para completar la información de una clase en una ontología que necesitamos, es necesario añadir el axioma de cierre, que evita que se añadan ingredientes a *MargheritaPizza* y a *SohoPizza*:
 - Seleccionar *MargheritaPizza* => SubClass of +
 - Class expression editor: *hasTopping* only (*MozzarellaTopping* or *TomatoTopping*)
- Seleccionar *SohoPizza* => SubClass of +
- Class expression editor: *hasTopping* only (*MozzarellaTopping* or *TomatoTopping* or *ParmesanTopping* or *OliveTopping*)
- Comprobar la jerarquía de clases inferidas.
- Comprobar las deducciones haciendo click en la ?



The screenshot shows the Protege OWL editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The address bar shows the URL: <http://www.semanticweb.org/sergg/ontologies/2021/10/pizza>. The main workspace is divided into several panes. On the left, the 'Class hierarchy: VegetarianPizza' pane shows a tree structure starting from 'owl:Thing' and branching into 'Pizza', 'CheesyPizza', 'NamedPizza', and 'VegetarianPizza'. 'VegetarianPizza' is highlighted. The 'Annotations: VegetarianPizza' pane shows a list of annotations, including 'SubClassOf hasTopping (CheeseTopping or VegetableTopping)'. The 'Usage: VegetarianPizza' pane shows a list of uses, including 'VegetarianPizza SubClassOf Pizza'. The 'Description: VegetarianPizza' pane shows the class description: 'Equivalent To: Pizza and (hasTopping only (CheeseTopping or VegetableTopping))'. The 'SubClass Of' pane shows 'Pizza' as a subclass. The 'General class axioms' pane shows 'hasBase some PizzaBase' and 'hasTopping some PizzaTopping'. The 'Instances' and 'Target for Key' panes are empty. The bottom status bar indicates 'Reasoner active' and 'Show Inferences'.

Esta vez sí que se deduce que MargheritaPizza y SohoPizza son subclase de VegetarianPizza, porque tras definir la cláusula de cierre, se cumplen las condiciones necesarias y suficientes para pertenecer a la clase.

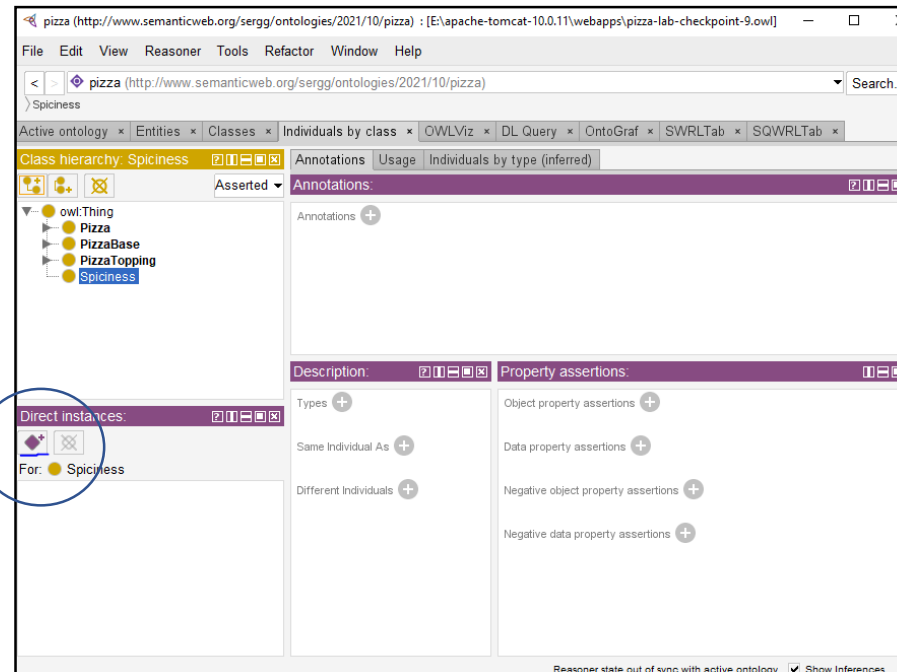
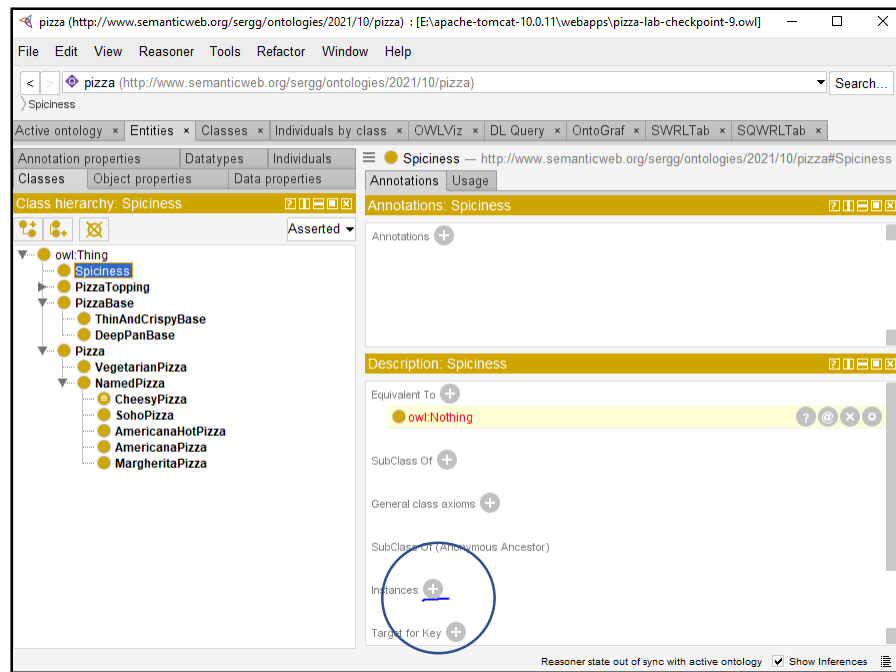
Clases Enumeradas

Clases Enumeradas

- En ocasiones, se necesita incluir en una Ontología información específica sobre clases, como por ejemplo, una propiedad que pueda tener un conjunto de valores finito.
- Para crearlas, crearemos una clase nueva, y crearemos individuos para cada uno de los valores posibles.

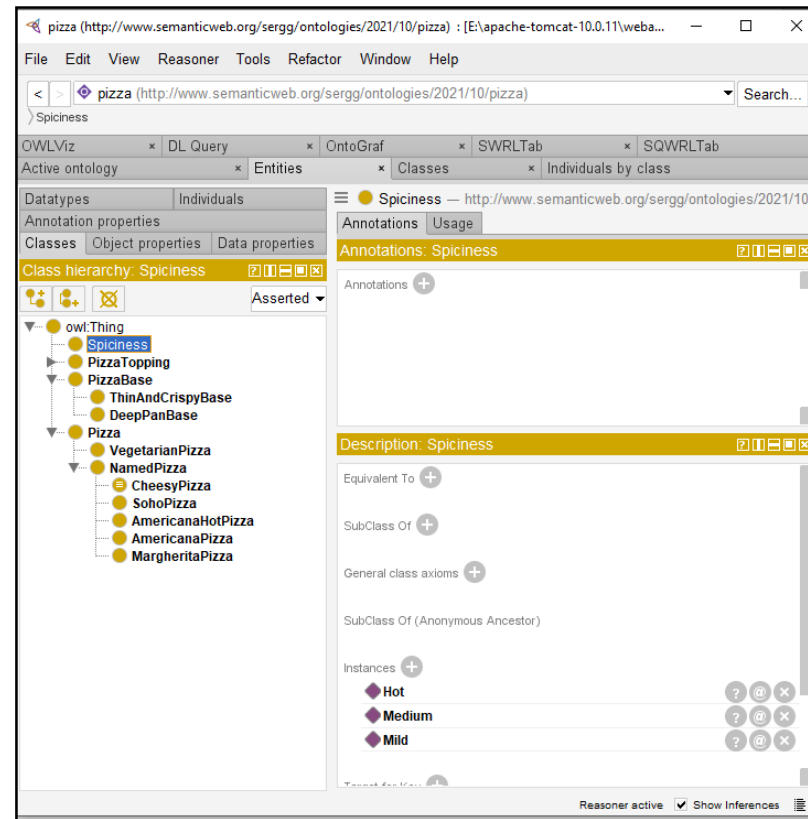
Creamos una clase enumerada para el nivel de Picante

- Seleccionando owl:Thing desde la vista de clases (Asserted), creamos la clase Spiciness.
- A continuación, seleccionamos la clase Spiciness y creamos 3 instancias (hay varias maneras de hacerlo)



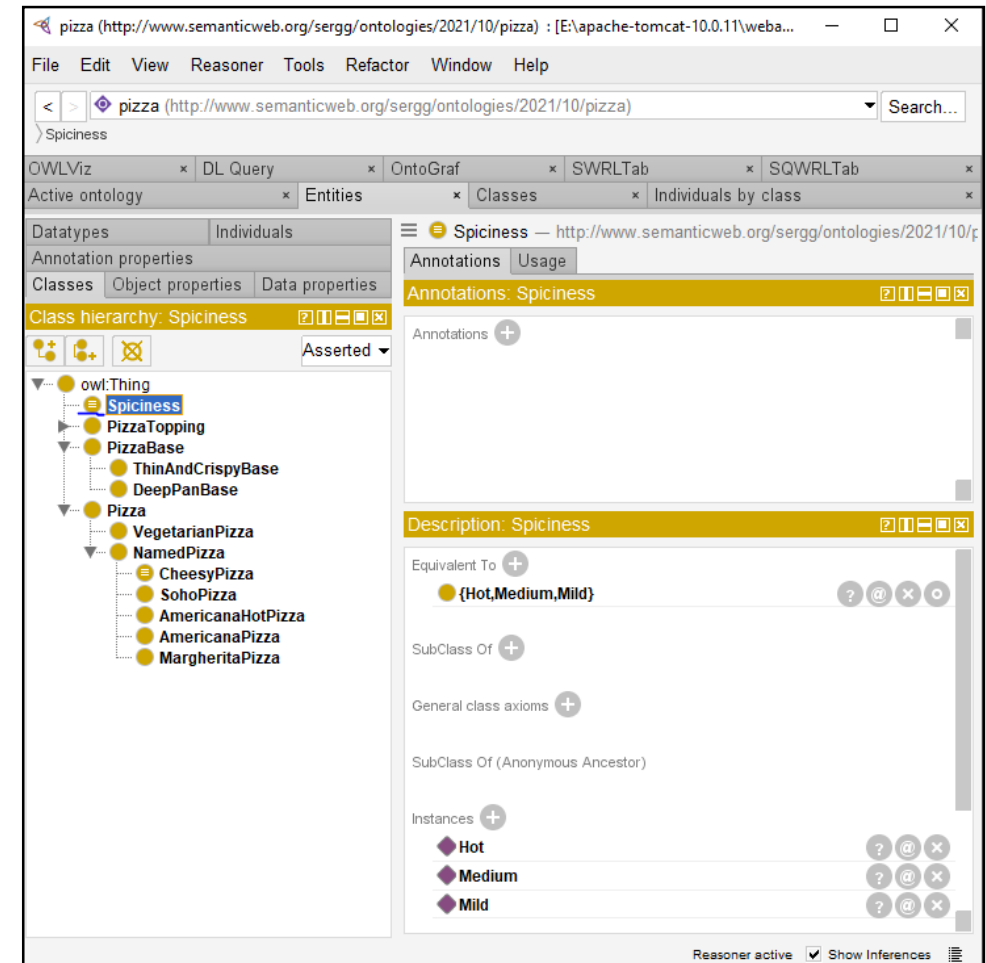
Creación de Instancias

- Creamos las instancias (Individuos): Hot, Medium, Mild de clase Spiciness



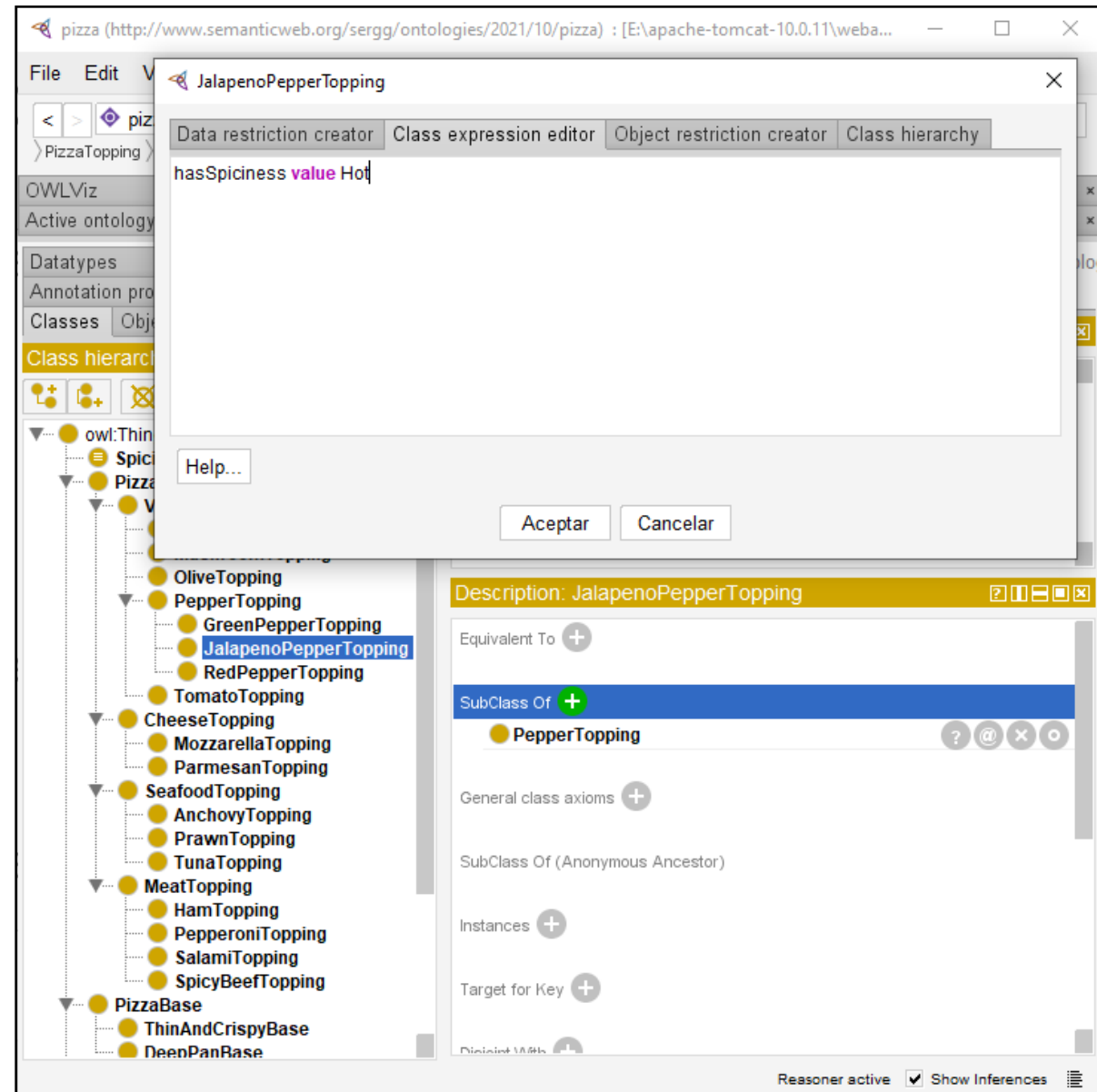
Limitamos los casos posibles de Spiciness

- Una vez creados, seleccionar la clase Spiciness y hacer click en Equivalent To + en la vista de Descripción.
- Seleccionar, Class Expression Editor y meter el axioma de lógica descriptiva {Hot, Medium, Mild}
- Ejecutar el razonador => Ahora, la clase es Definida (debido a que tiene la sección *Equivalent To* definida), lo sabemos gracias a las 3 líneas horizontales



Asociar un nivel de Picante a las clases

- Seleccionar Object Properties y crear una nueva propiedad llamada: hasSpiciness que sea subpropiedad, como las anteriores propiedades de TopObjectProperty.
 - Dominio: PizzaTopping
 - Rango: Spiciness
 - Sincronizar el razonador.
-
- Desde la vista de clases, seleccionar JalapenoPepperTopping y seleccionar SubClass Of +
 - Indicar hasSpiciness **value** Hot
 - Atención: La restricción se refiere a un individuo en concreto (Hot)



Crear una clase que represente a las clases picantes (I)

- Seleccionamos la clase Pizza y creamos la subclase SpicyPizza
- Seleccionamos SpicyPizza y hacemos click en SubClass Of +
 - ¿Qué axioma habrá que definir para identificar a las clases que son picantes?

Crear una clase que represente a las clases picantes (II)

- Seleccionamos la clase Pizza y creamos la subclase SpicyPizza
- Seleccionamos SpicyPizza y hacemos click en SubClass Of +
 - ¿Qué axioma habrá que definir para identificar a las clases que son picantes?
 - hasTopping some (hasSpiciness value Hot)
- Ejecutar el razonador.
- Observar la vista de Class hierarchy (inferred)
- ¿Se ha añadido la pizza con ingredientes picantes? ¿Por qué?

Crear una clase que represente a las clases picantes (III)

- Seleccionamos la clase Pizza y creamos la subclase SpicyPizza
- Seleccionamos SpicyPizza y hacemos click en SubClass Of +
 - ¿Qué axioma habrá que definir para identificar a las clases que son picantes?
 - `hasTopping some (hasSpiciness value Hot)`
- Ejecutar el razonador.
- Observar la vista de Class hierarchy (inferred)
 - ¿Se ha añadido la pizza con ingredientes picante? ¿Por qué?
 - Falta convertirla en Clase Definida para aplicar el cierre de clase
- Seleccionamos SpicyPizza y la convertimos en clase definida
- Ejecutar el razonador.
- Observar la vista de Class hierarchy (inferred)

pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-11.owl]

File Edit View Reasoner Tools Refactor Window Help

< > pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-11.owl]

Pizza > SpicyPizza > AmericanaHotPizza

Active ontology x Entities x Classes x Individuals by class

Annotation properties Datatypes Individuals
 Classes Object properties Data properties

Class hierarchy: AmericanaHotPizza

owl:Thing
 Pizza
 NamedPizza
 SpicyPizza
 AmericanaHotPizza
 VegetarianPizza
 PizzaBase
 PizzaTopping
 Spiciness

Explanation for AmericanaHotPizza SubClassOf SpicyPizza

☒ Show regular justifications ☒ All justifications
☐ Show laconic justifications ☐ Limit justifications to 2

☐ Display laconic explanation

Explanation 1

Explanation for: AmericanaHotPizza SubClassOf SpicyPizza

AmericanaHotPizza SubClassOf hasTopping some JalapenoPepperTopping
 JalapenoPepperTopping SubClassOf hasSpiciness value Hot
 SpicyPizza EquivalentTo hasTopping some (hasSpiciness value Hot)

Aceptar

Description: AmericanaHotPizza

Equivalent To +

SubClass Of +

hasTopping some JalapenoPepperTopping
 hasTopping some MozzarellaTopping
 hasTopping some PepperoniTopping
 hasTopping some TomatoTopping
 NamedPizza
 CheesyPizza
 SpicyPizza

General class axioms +

SubClass Of (Anonymous Ancestor)

Reasoner active ☒ Show Inferences

Restricciones sobre propiedades y datos

Restricciones de Cardinalidad

- Sirven para definir una clase en función del número de relaciones con otros individuos o tipos de datos.
- Creamos una clase que represente a las pizzas con al menos 3 ingredientes.
- Seleccionamos la clase Pizza y creamos una subclase InterestingPizza
- Restringimos la clase, haciendo click en SubClass Of +, y en el Class expression editor, introducimos:
 - hasTopping min 3 PizzaTopping
- Convertimos la clase InterestingPizza en clase Definida
- Ejecutamos el razonador
- Comprobamos las clases inferidas en la vista de Class hierarchy (inferred)

pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-12.owl]

File Edit View Reasoner Tools Refactor Window Help

< > pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza)

Pizza InterestingPizza

Active ontology x Entities x Classes x Individuals by class x OWLViz x DL Query x OntoGraf x SWRLTab x SQWRLTab

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Annotations Usage

Class hierarchy: InterestingPizza

Annotations: InterestingPizza

Annotations +

Description: InterestingPizza

Equivalent To +

Pizza and (hasTopping min 3 PizzaTopping)

SubClass Of +

Pizza

General class axioms +

SubClass Of (Anonymous Ancestor)

hasBase some PizzaBase

hasTopping some PizzaTopping

Instances +

Class hierarchy: InterestingPizza

owl:Thing

Pizza

InterestingPizza

AmericanaHotPizza

AmericanaPizza

SohoPizza

NamedPizza

SpicyPizza

VegetarianPizza

PizzaBase

PizzaTopping

Spiciness

Propiedades de tipos de datos

- OWL se beneficia de una gran librería de tipos de datos predefinidos, generalmente de XML, tales como `xsd:string` y `xsd:integer`.
- Ejercicio: Definiremos una propiedad de datos, que nos permita disponer de la información calórica de los individuos, y de identificar, en una clase, las pizzas con alto nivel calórico.

Creamos una propiedad de datos

- Abrimos la vista Window > Tabs > Data properties (ver siguiente transparencia)
- Seleccionamos owl:topDataProperty y creamos una subpropiedad llamada hasCaloricContent (dejar por defecto la opción de no hacer las propiedades disjuntas)
- Dominio: Pizza
- Rango: Built-in ➔ xsd:integer
- Seleccionamos Functional, porque una pizza únicamente puede tener un valor calórico.

Browser window: pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-13.o...

File Edit View Reasoner Tools Refactor Window Help

Navigation: < > pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) Search...

hasCaloricContent

Active ontology x Entities x Classes x Individuals by class x OWLViz x DL Query x OntoGraf x SWRLTab x SQWRLTab x

Annotation properties Datatypes Individuals Classes Object properties Data properties

Data property hierarchy: hasCaloricContent ? [] [] [] []

Annotations Usage

Annotations: hasCaloricContent ? [] [] [] []

Annotations +

Chara [] [] [] [] Description: hasCaloricContent [] [] [] []

☒ Functional

Equivalent To +

SubProperty Of +

Domains (intersection) +

Pizza ? @ x o

Ranges +

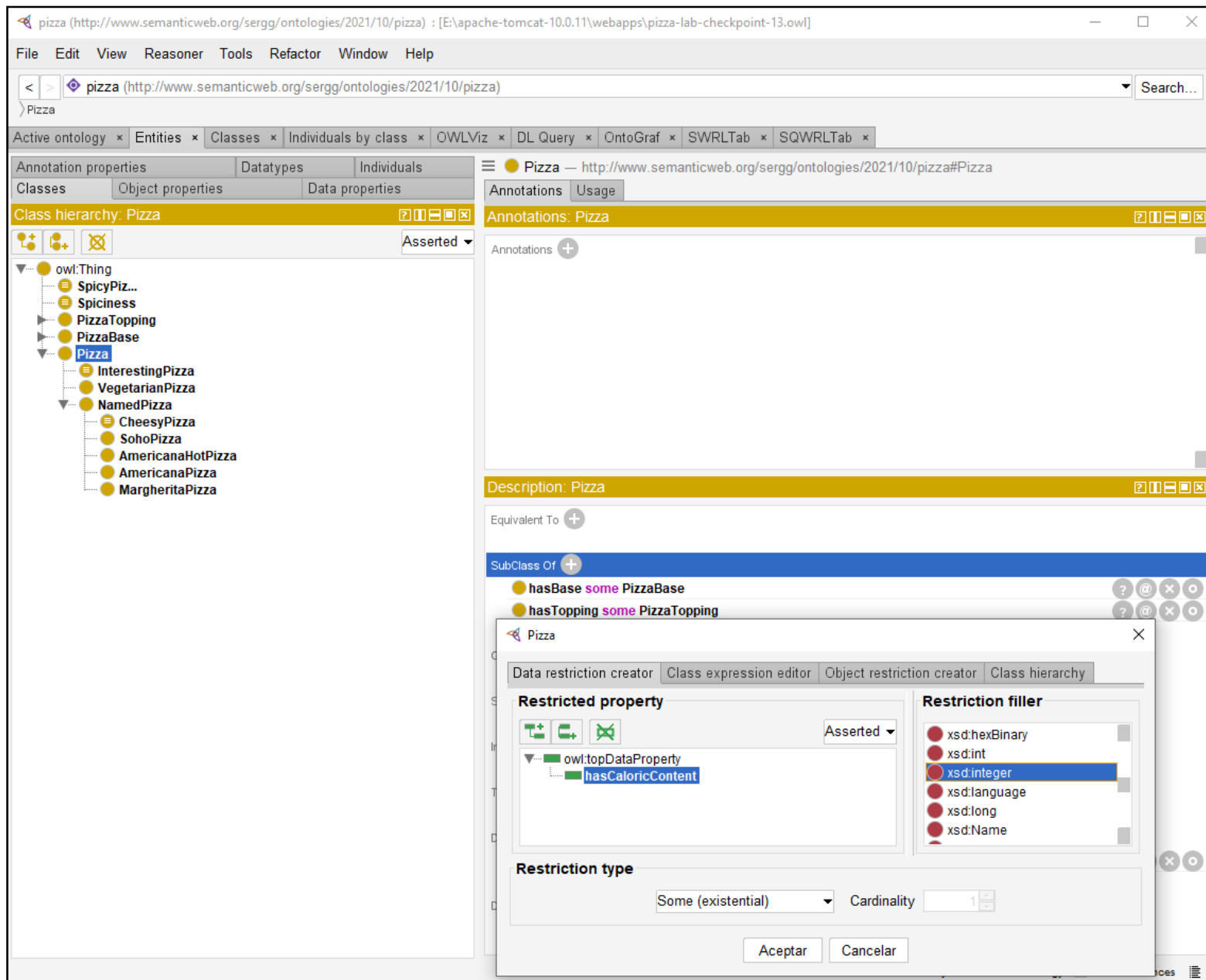
xsd:integer ? @ x o

Disjoint With +

Reasoner state out of sync with active ontology ☒ Show Inferences []

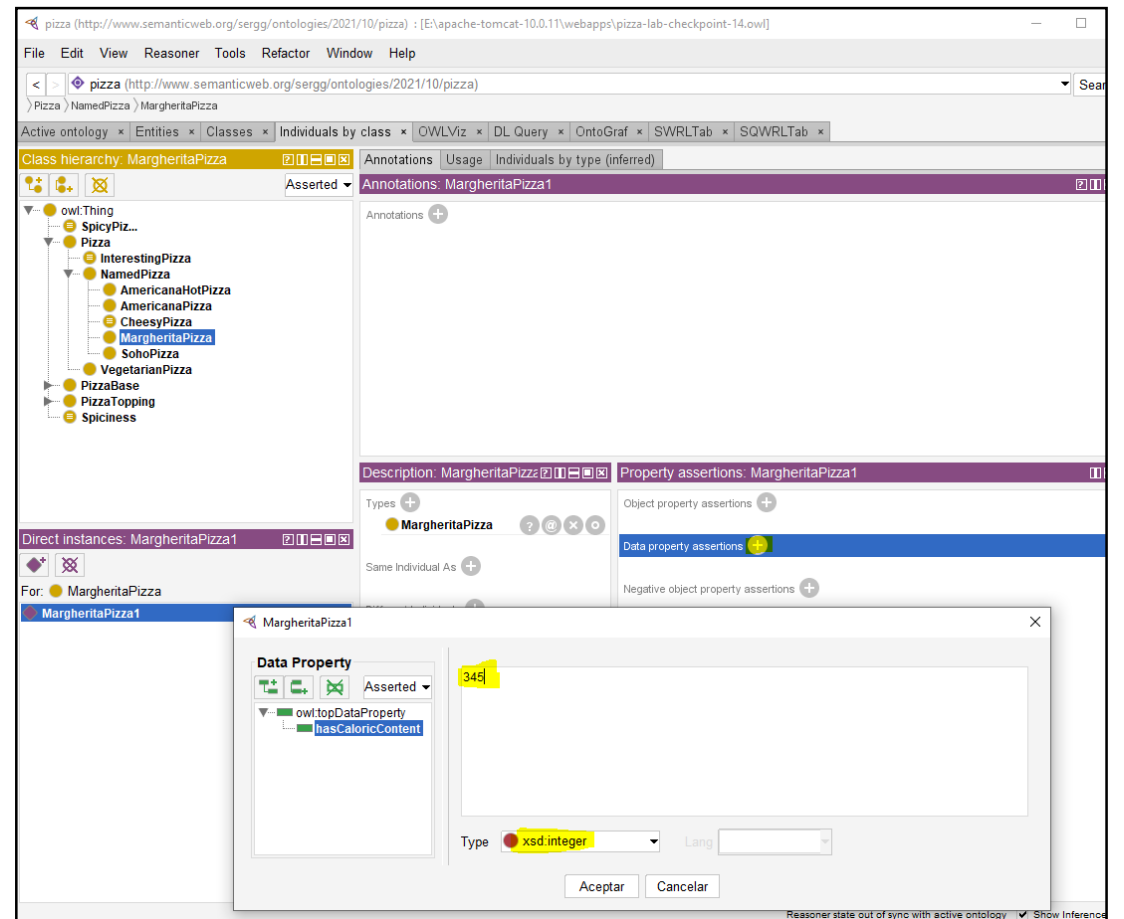
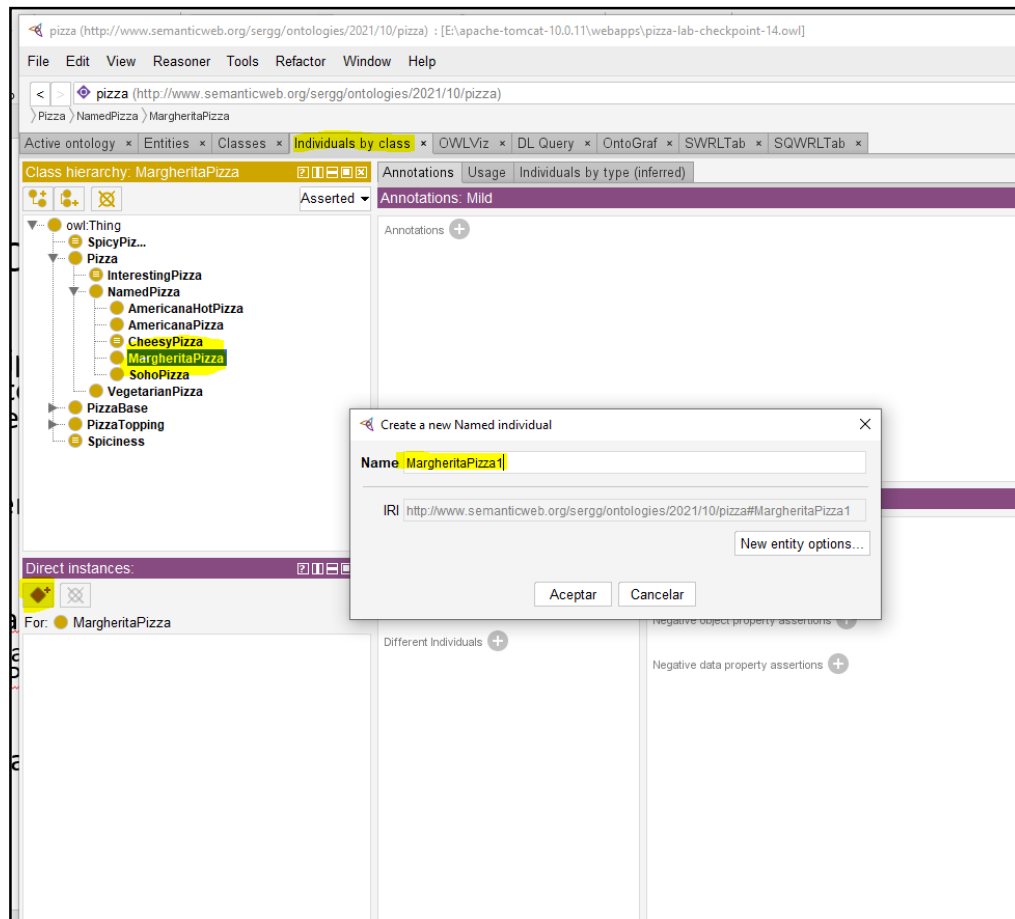
Creamos una restricción que obligue a todas las pizzas a proporcionar su valor calórico

- Seleccionamos Pizza en la vista de Classes
- Subclass Of +
- Usaremos esta vez: Data Restriction tab
 - seleccionar hasCaloricContent en la vista de Restricted property view
 - Seleccionamos xsd:integer
 - Tipo de restricción: some (también podría ser exactly 1, pero al haber definido hasCaloricContent como funcional, no es necesario)



Poblar la base de conocimiento

- Creamos varios individuos de tipo Pizza, y especificamos su valor calórico mediante esa propiedad de dato. Introducir 5 instancias con calorías por debajo de 400 y otros 5 por encima de 400 (ver las siguientes dos transparencias)
- Convención de nombres: Usaremos CamelBack añadiendo un número. Ejemplo: MargheritaPizza1
- Ir a Window > Tabs > Individuals by class
 - Añadir la vista Direct instances en Window > Views > Individual views > Direct instances
 - Hacer clic en el rombo morado de Direct instances para añadir individuos y, una vez seleccionados, poblar los datos que se deseen en la vista: Property assertions En este caso vamos a poblar para cada uno “Data property assertions +”
- Al asignar un valor a la propiedad de datos hasCaloricContent, especificar el tipo de dato xsd:integer.



pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-14.owl]

File Edit View Reasoner Tools Refactor Window Help

< > pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza)

Pizza

Active ontology x Entities x Classes x Individuals by class x OWLViz x DL Query x OntoGraf x SWRLTab x SQWRLTab x

Class hierarchy: Pizza

Annotations Usage Individuals by type (inferred)

Annotations: CustomPizza1

Annotations +

Description: CustomPizza1

Property assertions: CustomPizza1

Types +

Pizza

Same Individual As +

Different Individuals +

Object property assertions +

Data property assertions +

hasCaloricContent 560

Negative object property assertions +

Negative data property assertions +

Direct instances: CustomPizza1

For: Pizza

CustomPizza1

pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-14.owl]

File Edit View Reasoner Tools Refactor Window Help

< > pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza)

Active ontology x Entities x Classes x Individuals by class x OWLViz x DL Query x OntoGraf x SWRLTab x SQWRLTab x

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Individuals: CustomPizza2

Individuals +

AmericanaHotPizza1

AmericanaPizza1

CustomPizza1

CustomPizza2

CustomPizza3

CustomPizza4

Hot

MargheritaPizza1

MargheritaPizza2

Medium

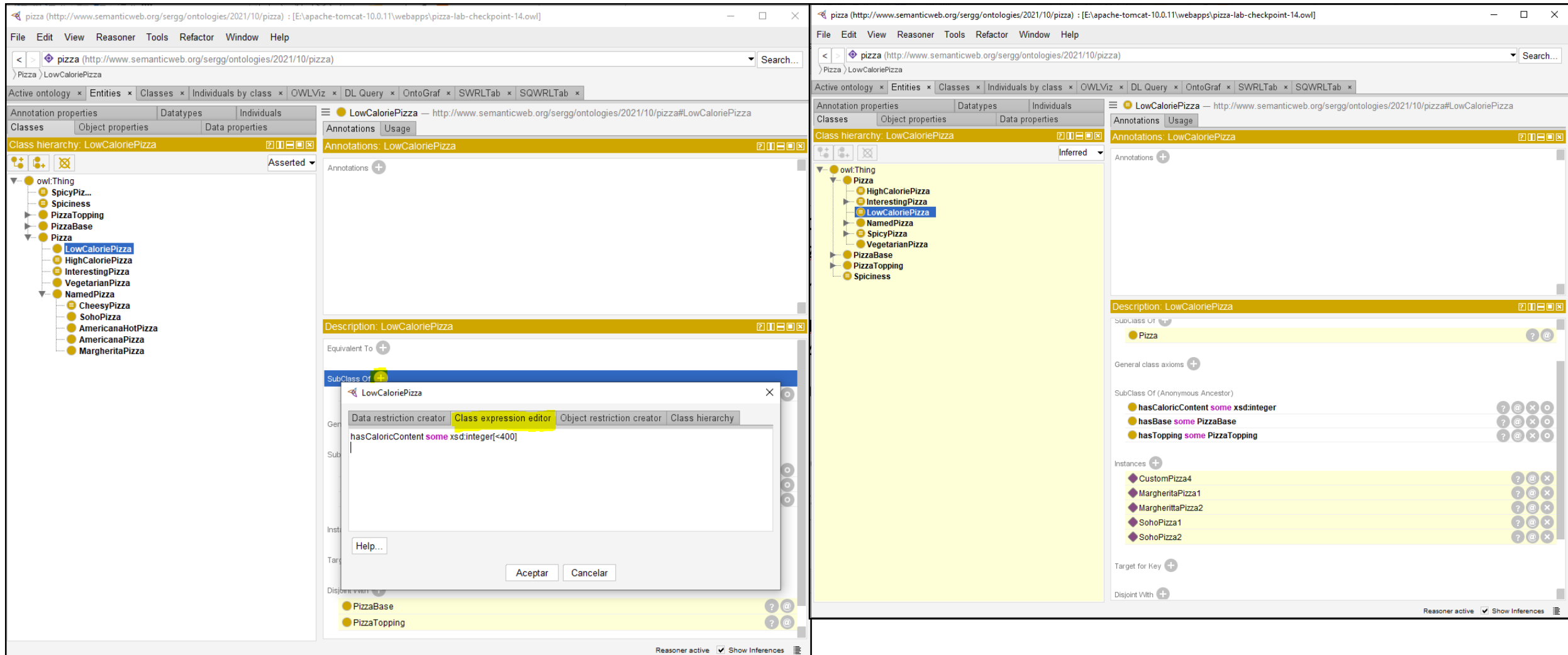
Mild

SohoPizza1

SohoPizza2

Definimos una clase que identifique a las Pizzas con alto nivel calórico

- Desde la vista de clases, crear una subclase de Pizza llamada HighCaloriePizza
- Seleccionar la nueva clase, y hacer click en SubClass Of +
- En el Class expression editor, introducir: `hasCaloricContent some xsd:integer[>=400]`
- Convertir la clase HighCaloriePizza en clase definida.
- Repetir los pasos para LowCaloriePizza, con valor menor de 400.
 - En este caso es conveniente dejar espacios entre los corchetes y la expresión “< 400”, si no el editor puede dar un error.
- Sincronizar el razonador y observar los resultados inferidos



Definición de propiedades transitivas

Añadiendo Orden a una clase Enumerada mediante propiedades transitivas

- Ejercicio para demostrar una propiedad transitiva y transitiva inversa
- Definiremos un orden mediante propiedades de objetos para {Hot, Medium, Mild}
- Crear la propiedad: isSpicierThan bajo owl:topObjectProperty
- Domain: Spiciness
- Características: Transitiva
- Creamos la propiedad inversa: isMilderThan
- Inverse Of + isSpicerThan
- Sincronizar el razonador y comprobar las inferencias que se han generado gracias a las características definidas.

pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-15.owl]

File Edit View Reasoner Tools Refactor Window Help

< > pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) Search...

isSpicierThan

Active ontology x Entities x Classes x Individuals by class x OWLViz x DL Query x OntoGraf x SWRLTab x SQWRLTab x

Annotation properties Datatypes Individuals
Classes Object properties Data properties

Object property hierarchy: isSpicierThan Annotations: isSpicierThan

owl:topObjectProperty
isSpicierThan
hasSpiciness
isIngredientOf
isBaseOf
isToppingOf
hasIngredient
hasBase
hasTopping

Annotations +

Character Description: isSpicierThan

☐ Functional
☐ Inverse functional
☒ Transitive
☐ Symmetric
☐ Asymmetric
☐ Reflexive
☐ Irreflexive

Equivalent To +
SubProperty Of +
owl:topObjectProperty
Inverse Of +
Domains (intersection) +
Spiciness
Ranges (intersection) +
Disjoint With +
SuperProperty Of (Chain) +

Reasoner state out of sync with active ontology ☒ Show Inferences

pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-15.owl]

File Edit View Reasoner Tools Refactor Window Help

< > pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) Search...

isMildierThan

Active ontology * Entities * Classes * Individuals by class * OWLViz * DL Query * OntoGraf * SWRLTab * SQWRLTab *

Datatypes Individuals

Annotation properties

Classes Object properties Data properties

Object property hierarchy: isMildierThan

Asserted

- owl:topObjectProperty
 - isMildierThan
 - isSpicierThan
 - hasSpiciness
 - isIngredientOf
 - isBaseOf
 - isToppingOf
 - hasIngredient
 - hasBase
 - hasTopping

Annotations

Annotations: isMildierThan

Annotations +

Characteristic Description: isMildierThan

☐ Functional

☐ Inverse functional

☐ Transitive

☐ Symmetric

☐ Asymmetric

☐ Reflexive

☐ Irreflexive

Equivalent To +

SubProperty Of +

owl:topObjectProperty

Inverse Of +

Domains (intersection) +

Ranges (intersection) +

Disjoint With +

SuperProperty Of (Chain) +

isMildierThan

Asserted

- owl:topObjectProperty
 - isMildierThan
 - isSpicierThan
 - hasSpiciness
 - hasIngredient
 - hasBase
 - hasTopping
 - isIngredientOf
 - isBaseOf
 - isToppingOf

Acceptar Cancelar

ontology ☒ Show Inferences

Definiendo el orden de la propiedad

- En la vista de Individuals, seleccionar Hot.
- En Object Property assertions, definimos “isSpicierThan”, y a continuación introducimos el valor Medium, que es el siguiente inmediato a Hot.
- Repetimos para definir que Medium es más picante que Mild.
- Sincronizamos el razonador y observamos como se han añadido las nuevas inferencias (ver dos transparencias más adelante)

pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-15.owl]

File Edit View Reasoner Tools Refactor Window Help

< > pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) Search...

Active ontology x Entities x Classes x Individuals by class x OWLViz x DL Query x OntoGraf x SWRLTab x SQWRLTab x

Datatypes Individuals
Annotation properties
Classes Object properties Data properties

Individuals: Hot

- AmericanaHotPizza1
- AmericanaPizza1
- CustomPizza1
- CustomPizza2
- CustomPizza3
- CustomPizza4
- Hot
- MargheritaPizza1
- MargherittaPizza2
- Medium
- Mild
- SohoPizza1
- SohoPizza2

Hot — http://www.semanticweb.org/sergg/ontologies/2021/10/pizza#Hot

Annotations Usage

Annotations: Hot

Annotations +

Description: Hot

Types +

- Spiciness

Same Individual As +

Different Individuals +

Property assertions: Hot

Object property assertions +

Data property assertions +

Negation

Negation

Hot

isSpicierThan Medium

(Tip: Use CTRL+Space to auto-complete names)

Aceptar Cancelar

Reasoner active ☒ Show Inferences

pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-15.owl]

File Edit View Reasoner Tools Refactor Window Help

Active ontology x Entities x Classes x Individuals by class x OWLViz x DL Query x OntoGraf x SWRLTab x SQWRLTab x

Datatypes Individuals

Annotation properties

Classes Object properties Data properties

Annotations: Medium

Annotations +

Types +

Spiciness

Same Individual As +

Different Individuals +

Property assertions: Medium

Object property assertions +

isSpicierThan Mild

isMildierThan Hot

Data property assertions +

Negative object property assertions +

Negative data property assertions +

Individuals: Medium

- AmericanaHotPizza1
- AmericanaPizza1
- CustomPizza1
- CustomPizza2
- CustomPizza3
- CustomPizza4
- Hot
- MargheritaPizza1
- MargherittaPizza2
- Medium
- Mild
- SohoPizza1
- SohoPizza2

pizza (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) : [E:\apache-tomcat-10.0.11\webapps\pizza-lab-checkpoint-15.owl]

File Edit View Reasoner Tools Refactor Window Help

Active ontology x Entities x Classes x Individuals by class x OWLViz x DL Query x OntoGraf x SWRLTab x SQWRLTab x

Datatypes Individuals

Annotation properties

Classes Object properties Data properties

Annotations: Mild

Annotations +

Types +

Spiciness

Same Individual As +

Different Individuals +

Property assertions: Mild

Object property assertions +

isMildierThan Medium

isMildierThan Hot

Data property assertions +

Negative object property assertions +

Negative data property assertions +

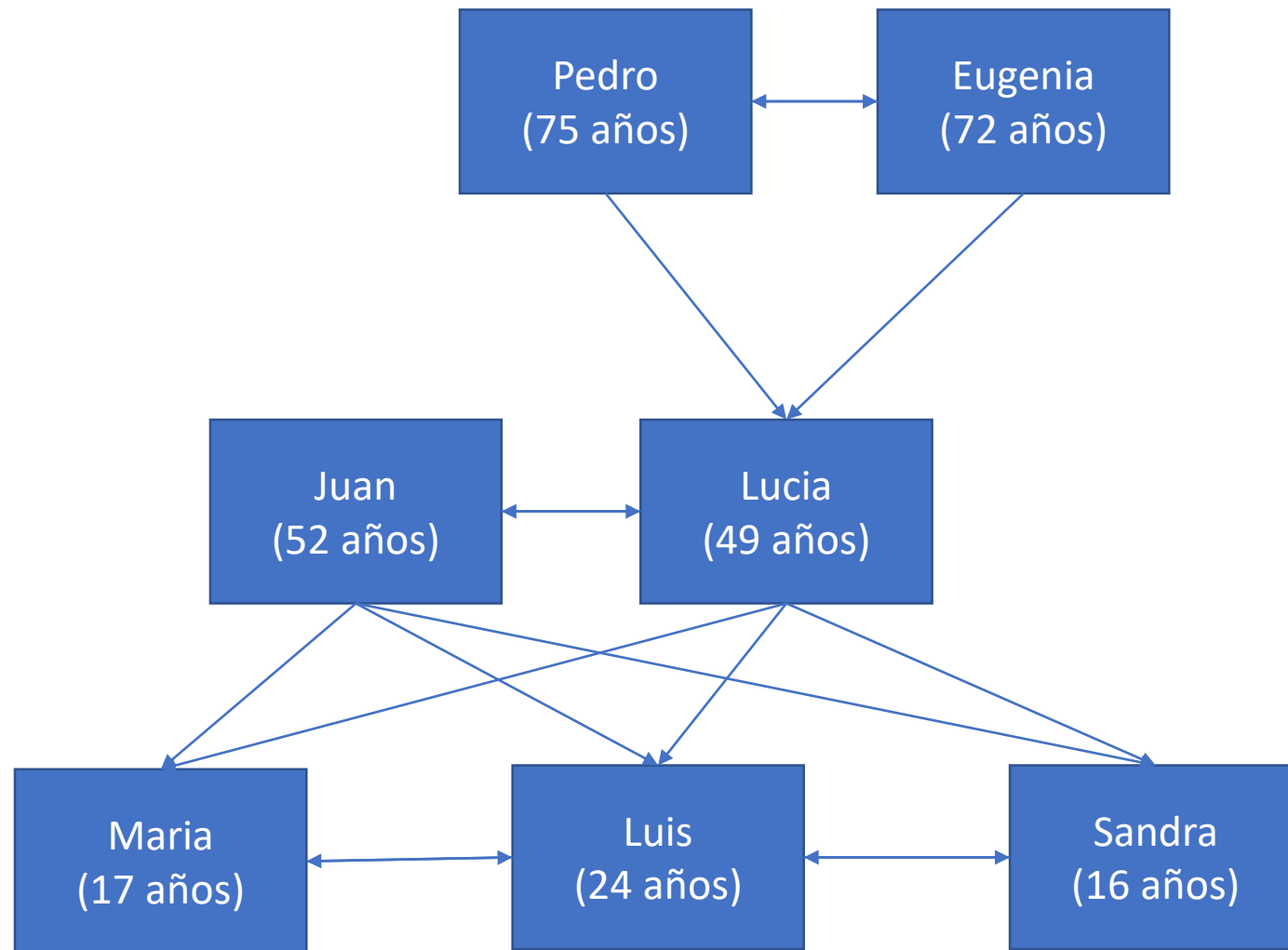
Individuals: Mild

- AmericanaHotPizza1
- AmericanaPizza1
- CustomPizza1
- CustomPizza2
- CustomPizza3
- CustomPizza4
- Hot
- MargheritaPizza1
- MargherittaPizza2
- Medium
- Mild
- SohoPizza1
- SohoPizza2

Reasoner active Show Inferences

PRÁCTICA: PARTE 2

- Completar la ontología sobre las familias, considerando:
 - Definir la edad de una persona
 - Definir el sexo de la persona como una clase enumerada
 - Especificar subpropiedades de esDescendienteDe y esAscendienteDe en función del sexo (esPadreDe, esMadreDe, esHijoDe, esHijaDe)
 - Crear una clase para identificar a las personas menores de edad y otra para mayores de 65 años
 - Una clase que identifique a las personas que estén emparejadas y que tengan descendencia
 - Una clase que identifique a los padres y otra a las madres
 - Lo que consideres que enriquezca la ontología



FIN DE LA SEGUNDA SESION