

TERCERA SESION

Sesion 3 - Agenda

- Suposición de nombres únicos (Uniques Names Assumption)
- Cláusulas de cierre
- Cardinalidad exactly
- Visualización de Ontologías
- Aplicación de reglas sobre ontologías con SWRL
- Consultas semánticas con SQWRL

Suposición de nombres únicos

Unique Names Assumption

- Las ontologías y el motor de inferencias, funcionan bajo el criterio de que dos (o más) individuos podrían ser el mismo, salvo que se especifique lo contrario.
 - Permite identificar coincidencias conceptuales
 - Requiere que a la hora de poblar la base de datos, se aporte información específica que separe a individuos que son diferentes

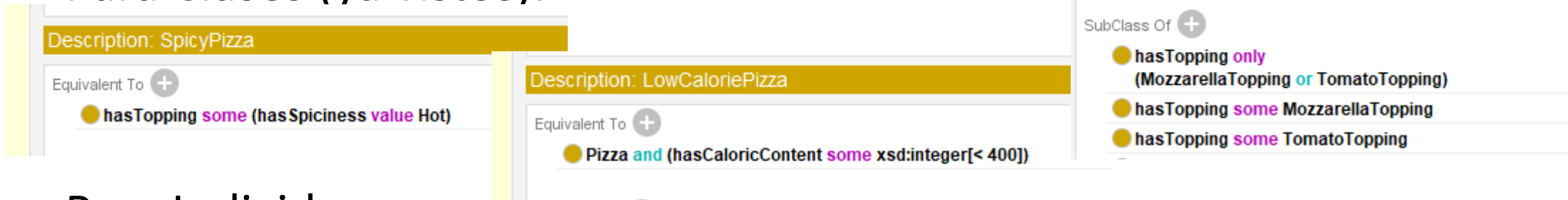
The screenshot displays a software interface with three main panels for an individual named 'Eugenia':

- Individuals: Eugenia**: A list of names and genders, each preceded by a purple diamond icon. The list includes: Eugenia, femenino, Juan, Lucia, Luis, Maria, masculino, Pedro, and Sandra. 'Eugenia' is highlighted with a blue background.
- Annotations: Eugenia**: A panel with a header 'Annotations' and a plus icon, currently empty.
- Description: Eugenia**: A panel with three sections:
 - Types**: A plus icon followed by a yellow circle and the text 'Persona'.
 - Same Individual As**: A plus icon, currently empty.
 - Different Individuals**: A plus icon followed by a purple diamond icon and the text 'Juan, Lucia, Luis, Maria, Pedro, Sandra'.

Cláusulas de cierre

Cláusulas de Cierre

- Conocer técnicas de cierre de cláusula es fundamental para el desarrollo de Ontologías completas.
- Para Clases (ya vistos):



The image shows three screenshots of OWL class definitions for pizzas. Each screenshot displays a class name in a yellow header bar, followed by an 'Equivalent To' section with a plus icon, and then a list of properties and restrictions.

- SpicyPizza:** hasTopping some (hasSpiciness value Hot)
- LowCaloriePizza:** Pizza and (hasCaloricContent some xsd:integer[< 400])
- MargheritaPizza:** hasTopping only (MozzarellaTopping or TomatoTopping), hasTopping some MozzarellaTopping, hasTopping some TomatoTopping

- Para Individuos:
 - Es necesario definir tanto las propiedades de objeto con cada otro individuo, como indicar que el individuo pertenece a una clase que únicamente tiene esa propiedad con esos individuos



The image shows two screenshots of OWL property assertions for the 'Rack' class. The left screenshot shows the 'Types' section with a plus icon, and the right screenshot shows the 'Object property assertions' section with a plus icon.

- Types:** tieneComponentes only {Juan, Luis, Pedro}
- Object property assertions:** tieneComponentes Juan, tieneComponentes Luis, tieneComponentes Pedro

Cardinalidad exactly

Definir una clase que permita únicamente pizzas con 2 ingredientes

- Desde la vista de clases, crear una subclase de Pizza llamada 2IngredientsPizza
- Seleccionar la nueva clase, y hacer click en SubClass Of +
- En el Class expression editor, introducir: hasTopping exactly 2 PizzaTopping
- Convertir la clase en definida (ver siguiente diapositiva)
- Tras sincronizar el razonador y posicionándonos en la vista de jerarquía de clases inferida, observamos que no hay ninguna clase que herede de la que acabamos de crear.
- El motivo es que la definición de PizzaMargherita permite añadir multiples toppings (por ejemplo una pizza con triple de queso mozzarella)

Definir una clase que permita únicamente pizzas con 2 ingredientes

The screenshot shows the OntoGraf web interface with the following components:

- Top Bar:** Active ontology x Entities x Individuals by class x DL Query x SWRLTab x OntoGraf x
- Navigation Tabs:** Annotation properties, Datatypes, Individuals, Classes, Object properties, Data properties.
- Class Hierarchy:** A tree view on the left showing the hierarchy of classes. The **2IngredientsPizza** class is highlighted under the **Pizza** class.
- Class Definition Panel:** On the right, the definition for **2IngredientsPizza** is shown.
 - Description:** 2IngredientsPizza
 - Equivalent To:** Pizza and (hasTopping exactly 2 PizzaTopping)
 - SubClass Of:** Pizza
 - General class axioms:**
 - SubClass Of (Anonymous Ancestor):
 - hasCaloricContent some xsd:integer
 - hasTopping some PizzaTopping
 - hasBase some PizzaBase
 - Instances:** (Empty list)
 - Target for Key:** (Empty list)
 - Disjoint With:** PizzaBase, PizzaTopping, InterestingPizza

Definir una clase que represente a la Pizza Margharita con el mínimo posible de ingredientes

- Clonar la clase PizzaMargharita y renombrarla (Rename IRI) al nombre MinimalPizzaMargharita
- Seleccionar la nueva clase, y hacer click en SubClass Of +
- En el Class expression editor, añadir a la expresión actual: (hasTopping exactly 1 MozzarellaTopping) and (hasTopping exactly 1 TomatoTopping)
- Tras sincronizar el razonador y posicionándonos en la vista de jerarquía de clases inferida, observamos que ahora la clase MinimalPizza Margharita hereda de 2IngredientsPizza (ver siguientes diapositivas)

Definir una clase que represente a la Pizza Margharita con el mínimo posible de ingredientes

The screenshot displays a Semantic Web editor interface. On the left, a class hierarchy is shown under 'owl:Thing'. The hierarchy includes 'Pizza', which contains subclasses like '2IngredientsPizza', 'CheesyPizza', 'HighCaloriePizza', 'InterestingPizza', 'LowCaloriePizza', 'NamedPizza', 'PizzaBase', 'PizzaTopping', and 'SpicyPizza'. 'NamedPizza' is expanded, showing 'MinimalMargheritaPizza' as a subclass. The 'MinimalMargheritaPizza' class is highlighted. On the right, the 'Annotations' tab is active, showing an 'rdfs:comment' for 'MinimalMargheritaPizza' stating 'Margherita Pizza is defined by Mozzarella and Tomato toppings'. Below this, the 'Description' tab is active, showing the class description for 'MinimalMargheritaPizza'. The description includes 'Equivalent To' (none), 'SubClass Of' (listing 'NamedPizza', '2IngredientsPizza', and 'MargheritaPizza'), and 'General class axioms' (listing 'hasTopping exactly 1 MozzarellaTopping' and 'hasTopping exactly 1 TomatoTopping').

Annotation properties | Datatypes | Individuals
Classes | Object properties | Data properties

Class hierarchy: MinimalMargheritaPizza

owl:Thing

- Pizza
 - 2IngredientsPizza
 - CheesyPizza
 - HighCaloriePizza
 - InterestingPizza
 - LowCaloriePizza
 - NamedPizza
 - MinimalMargheritaPizza**
 - AmericanaHotPizza
 - AmericanaPizza
 - MargheritaPizza
 - SohoPizza
 - VegetarianPizza
- PizzaBase
 - DeepPanBase
 - ThinAndCrispyBase
- PizzaTopping
 - Spiciness
 - SpicyPizza

Annotations: MinimalMargheritaPizza

Annotations +

rdfs:comment [type: xsd:string]
Margherita Pizza is defined by Mozzarella and Tomato toppings

Description: MinimalMargheritaPizza

Equivalent To +

SubClass Of +

- hasTopping exactly 1 MozzarellaTopping
- hasTopping exactly 1 TomatoTopping
- hasTopping only (MozzarellaTopping or TomatoTopping)
- NamedPizza
- 2IngredientsPizza
- MargheritaPizza

General class axioms +

SubClass Of (Anonymous Ancestor)

[pizza](http://www.semanticweb.org/sergg/ontologies/2021/10/pizza) (http://www.semanticweb.org/sergg/ontologies/2021/10/pizza)

[Pizza](#) > [2IngredientsPizza](#) > [MinimalMargheritaPizza](#)

[Active ontology](#) x [Entities](#) x [Individuals by class](#) x [DL Query](#) x [SWRLTab](#) x [OntoGraf](#) x

[Annotation properties](#) [Datatypes](#) [Individuals](#)

[Classes](#) [Object properties](#) [Data properties](#)

[Class hierarchy: MinimalMargheritaPizza](#)

[Inferred](#)

owl:Thing

- Pizza**
 - 2IngredientsPizza**
 - MinimalMargheritaPizza**
 - CheesyPizza**
 - HighCaloriePizza**
 - InterestingPizza**
 - LowCaloriePizza**
 - NamedPizza**
 - AmericanaHotPizza**
 - AmericanaPizza**
 - MargheritaPizza**
 - MinimalMargheritaPizza**
 - SohoPizza**
 - SpicyPizza**
 - VegetarianPizza**
- PizzaBase**
- PizzaTopping**
- Spiciness**

Annotations

rdfs:comment [type: xsd:string]
 Margherita Pizza is defined by Mozzarella and Tomato toppings

Description: MinimalMargheritaPizza

Equivalent To

SubClass Of

- hasTopping exactly 1 MozzarellaTopping**
- hasTopping exactly 1 TomatoTopping**
- hasTopping only (MozzarellaTopping or TomatoTopping)**
- NamedPizza**
- 2IngredientsPizza**
- MargheritaPizza**

General class axioms

SubClass Of (Anonymous Ancestor)

- hasCaloricContent some xsd:integer**
- hasTopping some PizzaTopping**
- hasBase some PizzaBase**
- NamedPizza and (hasTopping some MozzarellaTopping) and (hasTopping some TomatoTopping) and (MozzarellaTopping or TomatoTopping))**
- Pizza and (hasTopping exactly 2 PizzaTopping)**

Show: this
 Found 11 uses

MinimalM

- Minin**
- Minin**
- (Moz**

Definir un individuo que tenga 2 ingredientes

- Desde la vista de individuos, crear:
 - Un individuo llamado CrispyAndThinBase1 del tipo CrispyAndThinBase
 - Un individuo llamado MushroomTopping1 del tipo MushroomTopping
 - Un individuo llamado HamTopping1 del tipo HamTopping
 - Un individuo llamado 2IngredientsPizzaIndividual1 del tipo Pizza
- Seleccionar 2IngredientsPizzaIndividual1 y añadir las siguientes afirmaciones de propiedades de objetos :
 - hasBase CrispyAndThinBase1
 - hasTopping MushroomTopping1
 - hasTopping HamTopping1
- Aún falta un último detalle, ¿cuál podría ser?

Definir un individuo que tenga exactamente 2 ingredientes

- Desde la vista de individuos, crear:
 - Un individuo llamado CrispyAndThinBase1 del tipo CrispyAndThinBase
 - Un individuo llamado MushroomTopping1 del tipo MushroomTopping
 - Un individuo llamado HamTopping1 del tipo HamTopping
 - Un individuo llamado 2IngredientsPizzaIndividual1 del tipo Pizza
- Seleccionar 2IngredientsPizzaIndividual1 y añadir las siguientes afirmaciones de propiedades de objetos:
 - hasBase CrispyAndThinBase1
 - hasTopping MushroomTopping1
 - hasTopping HamTopping1
- Seleccionar el individuo como clase (Type)
 - hasTopping only {HamTopping1, MushroomTopping1}
- Sincroniza el razonador y observar el resultado

Definir un individuo que tenga exactamente 2 ingredientes

The screenshot shows the OntoGraf web interface for an ontology. The top navigation bar includes tabs for 'Active ontology', 'Entities', 'Individuals by class', 'DL Query', 'SWRLTab', and 'OntoGraf'. Below this, there are tabs for 'Annotation properties', 'Datatypes', 'Individuals', 'Classes', 'Object properties', and 'Data properties'. The 'Individuals' tab is active, showing a list of individuals on the left and a detailed view of '2IngredientPizzaIndividual1' on the right.

Individuals by class:

- 2IngredientPizzaIndividual1 (selected)
- AmericanaHotPizza1
- AmericanaPizza1
- CustomPizza1
- CustomPizza2
- CustomPizza3
- CustomPizza4
- HamTopping1
- Hot
- MargheritaPizza1
- MargherittaPizza2
- Medium
- Mild
- MushroomTopping1
- PepperoniTopping1
- SohoPizza1
- SohoPizza2
- ThinAndChrispyBase1

Annotations: 2IngredientPizzaIndividual1

Annotations: +

Description: 2IngredientPizzaIndividual1

Types: +

- hasTopping only {HamTopping1, MushroomTopping1}
- Pizza
- 2IngredientsPizza

Same Individual As: +

Different Individuals: +

Property assertions: 2IngredientPizzaIndividual1

Object property assertions: +

- hasBase ThinAndChrispyBase1
- hasTopping HamTopping1
- hasTopping MushroomTopping1
- hasIngredient HamTopping1
- hasIngredient MushroomTopping1
- hasIngredient ThinAndChrispyBase1

Data property assertions: +


Negative object property assertions: +

Visualización de Ontologías

Nos familiarizamos con la Ontología ampliada

- Cargar la ontología PizzaTutorialWithDataV2.owl
 - <https://campusvirtual.uva.es/mod/resource/view.php?id=1658318>
- Revisamos el contenido de la ontología:
 - Localizamos a las personas (*Person*) que pueden ser clientes (*Customer*), empleados (*Employee*)
 - Vemos la definición, el uso de las clases y sus instancias en la vista Usage.
 - Buscamos cómo se almacenan en esta ontología las compras (*purchasedPizza*, *purchasedByCustomer*)
 - Buscamos cómo se almacenan las preferencias de un cliente (*hasSpicinessPreference*)
 - Buscamos cómo se almacena el número de pizzas

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

[illegible]

Asserted ▼

- hasIngredient
- hasSpiciness
- hasSpicinessPreference
- isIngredientOf
 - isBaseOf
 - isToppingOf
- isMilderThan
- isSpicierThan
- purchasedByCustomer
- purchasedPizza

Annotations	Usage
-------------	-------

Annotations 

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

Equivalent To

+

SubProperty Of Inverse Of

Domains (intersection)

Customer

③ ④ ⑤ ⑥

Ranges (intersection)

 Spiciness

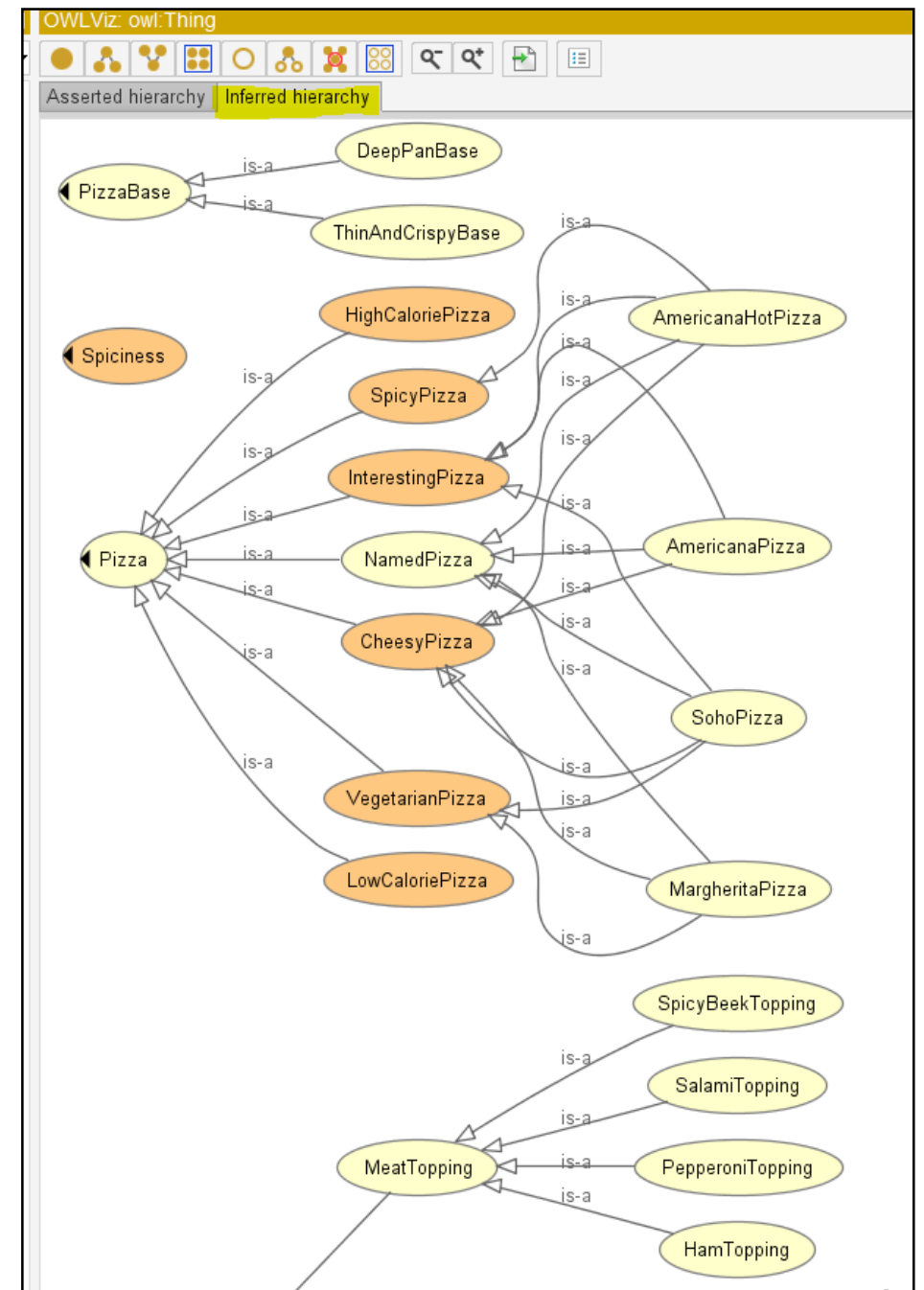
③ ④ ⑤ ⑥

Disjoint With 

SuperProperty Of (Chain)

Representación de Ontologías por medio de Grafos

- Protégé permite la generación automática de grafos de jerarquía de clases con OWLViz
- Puede visualizarse la parte introducida por el usuario (asserted) y la generada por el razonador (inferred)



OWLViz: opciones



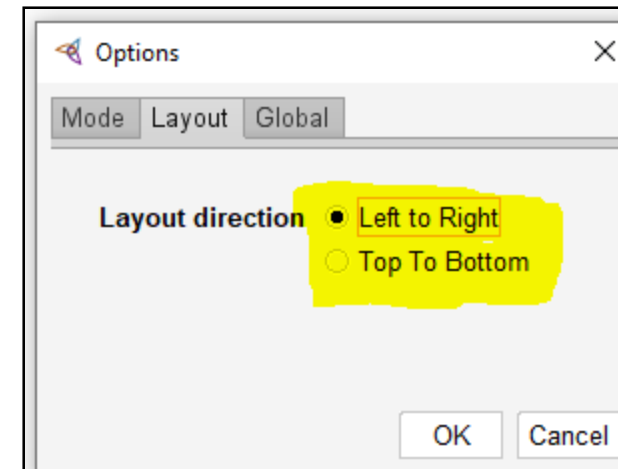
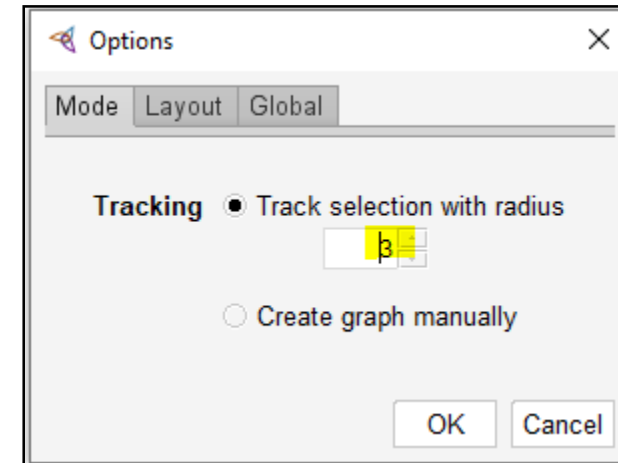
- Visualizar más niveles

Options – Mode

(no se refresca automáticamente)

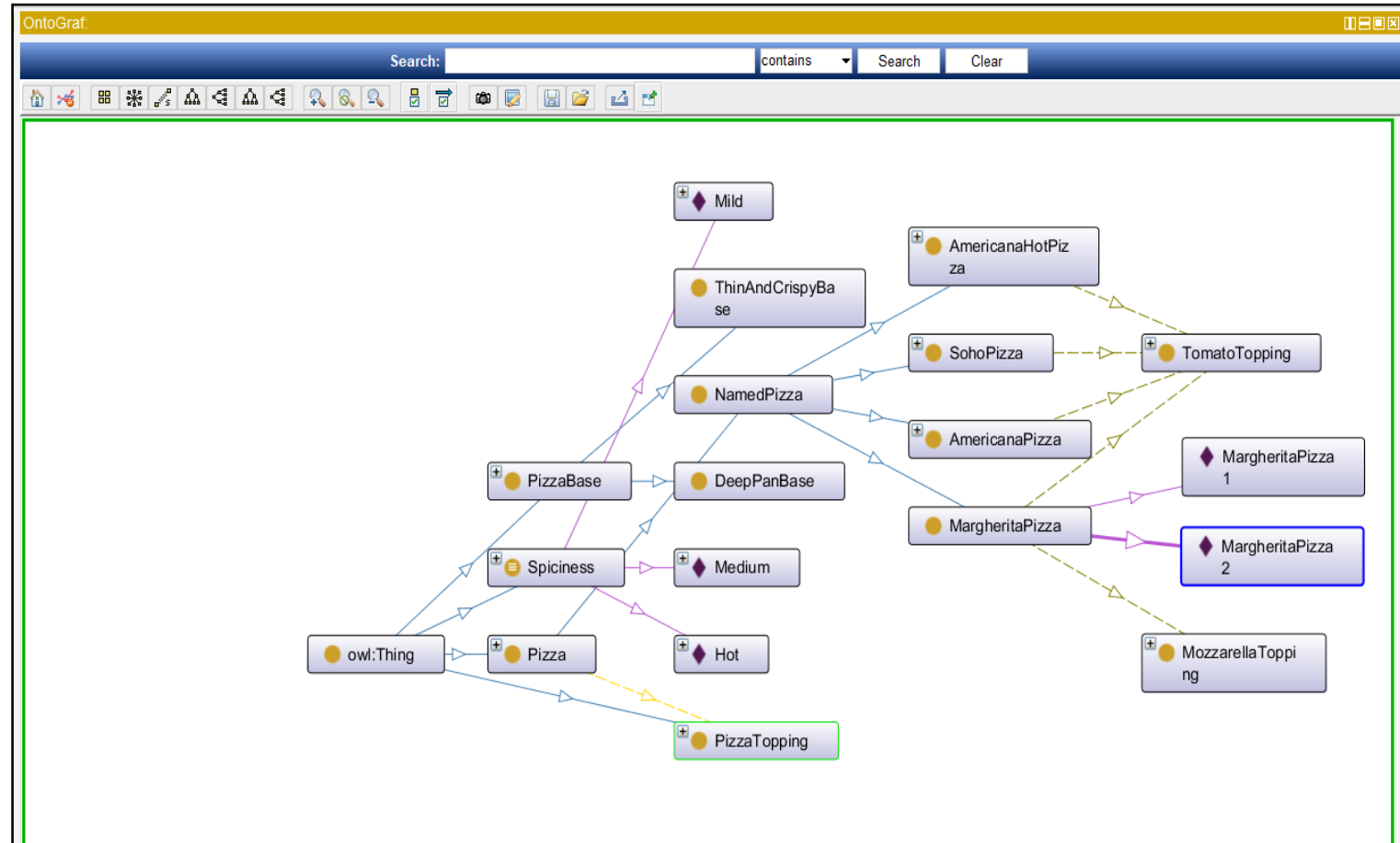
- Cambiar orientación

Options - Layout



Representación de Ontologías y Bases de Conocimiento por medio de Grafos: OntoGraph

- Seleccionar la ventana OntoGraph
- Permite visualizar jerarquías y categorizar las relaciones con códigos de colores en las líneas.
- Más complicado de ver para ontologías grandes



Aplicación de reglas sobre ontologías con SWRL

Definición de Reglas sobre Ontologías: SWRL

- SWRL: Semantic Web Rule Language
- El objetivo es enriquecer la ontología
- Recurso para hacer inferencias que no pueden definirse mediante axiomas de descripción lógica (o resulta demasiado complicado o poco intuitivo)
- Utilizaremos la ventana SWRL tab para definir reglas
- AVISO: NO utilizar los botones de ejecución (RUN), dejaremos que el motor de inferencias resuelva las reglas que definamos

Sistema de reglas de SWRL

- Sintaxis: 'Antecedente' -> 'Consecuente'
- Para representar la expresión “y”, se utiliza ^.
- Para representar “cualquier individuo”, se usan variables precedidas por ?.
- El consecuente únicamente se ejecuta, cuando se satisfacen todas las expresiones del antecedente.
- Dado que puede haber muchas posibilidades para que se cumplan las reglas, el razonador ejecutará todas las posibles, de manera que puede ejecutar iteraciones.

SWRL – Tipos de Expresiones

- Expresiones de clases:
 - Utiliza un único parámetro
 - Resuelve si un objeto es de una determinada clase (Pizza(?var))
- Expresiones de propiedades:
 - Utiliza dos parámetros, el primero es el individuo a comprobar, y el segundo el valor de la propiedad)
 - purchasedPizza(?c,?p) identificaría todos los casos en los que un cliente (?c) ha comprado una pizza (?p)
- Funciones integradas (built-in):
 - Utilizadas para hacer comprobaciones sobre datos básicos (comprobaciones matemáticas, sobre cadenas de texto, ...)
 - Por ejemplo, swrlb:greaterThan(?np, 1) (que el valor de la variable ?np sea mayor que 1)
 - Se pueden consultar en:
https://protegewiki.stanford.edu/images/5/57/SWRL-IQ_manual.pdf

Definición de reglas SWRL (I)

- Asegurarnos de utilizar la ontología `PizzaTutorialWithDataV2.owl`

Caso de uso 1:

- Por motivos de fallo de previsión, sobran ingredientes picantes, por lo que la gerencia de la pizzería, decide ofrecer un descuento a las personas que prefieren pizzas picantes (`hasSpicinessPreference Hot`)
- Pasos:
 - Crear una regla SWRL que dé un descuento de un 20% a aquellos usuarios que hayan comprado 2 o más pizzas y prefieren pizzas picantes.

Definición de reglas SWRL (II)

- Nombre de la regla: HotDiscountRule

1. Customer(?c) ^
2. numberOfPizzasPurchased(?c, ?np) ^
3. hasSpicinessPreference(?c, Hot) ^
4. swrlb:greaterThan(?np, 1)
5. -> hasDiscount(?c, 0.2)

1. recorremos los clientes
2. obtenemos el número de pizzas que han comprado los clientes (?c) y lo almacenamos en ?np
3. Comprobamos que el cliente (?c) tiene como preferencia de picante Hot
4. Comprobamos que el número de pizzas es mayor que 1 (2 o más)
5. **SE INFIERE** la propiedad hasDiscount para el cliente (?c) con el valor 0.2.

PizzaTutorial (http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial) : [E:\apache-tomcat-10.0.11\webapps\PizzaTutorialWithDataV2.0.owl]

File Edit View Reasoner Tools Refactor Window Help

< > PizzaTutorial (http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial) Search...

Pizza

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

Name	Rule	Comment

Edit

Name

HotDiscountRule

Comment

Provide a special discount for customer who prefer hot pizzas

Status

Ok

`pizzatutorial:Customer(?c) ^ pizzatutorial:numberOfPizzasPurchased(?c, ?np) ^ pizzatutorial:hasSpicinessPreference(?c, pizzatutorial:Hot) ^ swrlb:greaterThan(?np, 1) -> pizzatutorial:hasDiscount(?c, 0.2)`

Cancel Ok

New Edit Clone Delete

s->OWL

Reasoner state out of sync with active ontology ☒ Show Inferences

Grabar y
Sincronizar
Razonador

PizzaTutorial (http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial) : [E:\apache-tomcat-10.0.11\webapps\PizzaTutorialWithDataV2.0.owl]

File Edit View Reasoner Tools Refactor Window Help

< > PizzaTutorial (http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial) Search...

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: Customer1

- AmericanaHotPizza1
- AmericanaHotPizza2
- AmericanaHotPizza3
- AmericanaPizza1
- AmericanaPizza2
- CheesyPizza1
- Chef
- ChicagoAmericanaHotPizza1
- Customer1**
- Customer10
- Customer2
- Customer3
- Customer4
- Customer5
- Customer6
- Customer7
- Customer8
- Customer9
- CustomPizza1
- CustomPizza2
- GreenPepperTopping2
- Hot
- HotVeggiePizza1
- HotVeggiePizza2

Annotations: Customer1

Annotations +

Description: Customer1

Types +

- Customer

Same Individual As +

Different Individuals +

Property assertions: Customer1

Object property assertions +

- hasSpicinessPreference Hot
- purchasedPizza AmericanaHotPizza2
- purchasedPizza AmericanaHotPizza1

Data property assertions +

- numberOfPizzasPurchased 2
- hasPhone "555-111-3332"
- hasDiscount 0.2

Negative object property assertions +

Negative data property assertions +

Reasoner active ☒ Show Inferences

Definición de reglas SWRL (III)

- Asegurarnos de utilizar la ontología `PizzaTutorialWithDataV2.owl`

Caso de uso 2:

- Para el resto de los usuarios que también hayan consumido 2 o más pizzas, aplicamos un descuento del 10%
- Pasos:
 - Crear una regla SWRL que de un descuento de un 10% de aquellos usuarios que hayan comprado 2 o más pizzas y tengan preferencias de nivel de picante “menores” que Hot.

Definición de reglas SWRL (IV)

- Nombre de la regla: LessSpicyDiscountRule

1. Customer(?c) ^
2. numberOfPizzasPurchased(?c, ?np) ^
3. hasSpicinessPreference(?c, ?spr) ^
4. swrlb:greaterThan(?np, 1) ^
5. isMilderThan(?spr, Hot)
5. -> hasDiscount(?c, 0.1)

1. recorremos los clientes
2. obtenemos el número de pizzas que han comprado los clientes (?c) y lo almacenamos en ?np
3. Comprobamos que el cliente (?c) tiene como preferencia de picante (?spr)
4. Comprobamos que el número de pizzas es mayor que 1 (2 o más)
5. Comprobamos que la preferencia de picante (?spr) es más suave que Hot
6. **SE INFIERE** la propiedad hasDiscount para el cliente (?c) con el valor 0.1.

PizzaTutorial (http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial) : [E:\apache-tomcat-10.0.11\webapps\PizzaTutorialWithDataV2.0.owl]

File Edit View Reasoner Tools Refactor Window Help

< > PizzaTutorial (http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial) Search...

Pizza

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

HotDiscountRule

Comment: Provide a special discount for customer who prefer mild pizzas

Status: Ok

Rule: `pizzatutorial:Customer(?c) ^ pizzatutorial:numberOfPizzasPurchased(?c, ?np) ^ pizzatutorial:hasSpicinessPreference(?c, ?spr) ^ pizzatutorial:isMilderThan(?spr, pizzatutorial:Hot) ^ swrlb:greaterThan(?np, 1) -> pizzatutorial:hasDiscount(?c, 0.1)`

Cancel Ok

Reasoner state out of sync with active ontology Show Inferences

Grabar y
Sincronizar
Razonador

PizzaTutorial (http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial) : [E:\apache-tomcat-10.0.11\webapps\PizzaTutorialWithDataV2.0.owl]

File Edit View Reasoner Tools Refactor Window Help

< > PizzaTutorial (http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial) Search...

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: Customer5 Annotations: Customer5

Annotations +

Description: Customer: Property assertions: Customer5

Types +
Customer ? @ x o

Same Individual As +

Different Individuals +

Object property assertions +
purchasedPizza AmericanaPizza1 ? @ x o
hasSpicinessPreference Medium ? @ x o
purchasedPizza MediumVeggiePizza1 ? @ x o

Data property assertions +
numberOfPizzasPurchased 2 ? @ x o
hasPhone "555-111-3339" ? @ x o
hasDiscount 0.1 ? @

Negative object property assertions +

Negative data property assertions +

Reasoner active ☒ Show Inferences

AmericanaHotPizza1
AmericanaHotPizza2
AmericanaHotPizza3
AmericanaPizza1
AmericanaPizza2
CheesyPizza1
Chef
ChicagoAmericanaHotPizza1
Customer1
Customer10
Customer2
Customer3
Customer4
Customer5
Customer6
Customer7
Customer8
Customer9
CustomPizza1
CustomPizza2
GreenPepperTopping2
Hot
HotVeggiePizza1
HotVeggiePizza2

Consultas semánticas con SQWRL

Consultas semánticas: SQWRL



- SQWRL – Semantic Query-Enhanced Web Rule Language (squirrel)
- Esta herramienta, nos permite lanzar consultas sobre la base de conocimiento, para poder evaluar el resultado de las reglas SWRL antes de aplicarlas
- Trabajamos desde la pestaña SQWRL Tab

Obtenemos el resultado de las reglas

- Desde la pestaña SQWRL, clonar una de las reglas anteriores (se muestra en la transparencia siguiente).
- Modificar el consecuente, en lugar de inferir las propiedades, vamos a mostrar los resultados.
 - Nuevo consecuente: **`sqwrl:select(?c, ?np)`**
- Seleccionar la nueva regla, hacer clic en RUN y observar los resultados en la nueva pestaña que se crea en la parte inferior.

PizzaTutorial (http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial) : [E:\apache-tomc...

File Edit View Reasoner Tools Refactor Window Help

< > PizzaTutorial (http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial) Search...

Individuals by class × OWLViz × DL Query × OntoGraf × SWRLTab × SQWRLTab ×

Active ontology × Entities × Classes ×

Name	Query	Comment
HotDiscountRule	pizzatutorial:Customer(?c) ^ pizzatutorial:numberOfPizzasPurchased(?c, ?np)...	Provide a special discount for...
LessSpicyDiscountR...	pizzatutorial:Customer(?c) ^ pizzatutorial:numberOfPizzasPurchased(?c, ?np)...	Provide a special discount for...

New Edit Clone Delete

SQWRL Queries OWL 2 RL

Select a SQWRL query from the list above and press the 'Run' button.
If the selected query generates a result, the result will appear in a new sub tab.

The SWRLAPI supports an OWL profile called OWL 2 RL and uses an OWL 2 RL-based reasoner to perform querying.
See the 'OWL 2 RL' subtab for more information on this reasoner.

Executing queries in this tab does not modify the ontology.

Using Drools for query execution.

Run

Reasoner state out of sync with active ontology ☒ Show Inferences

Edit

Name

QueryHotDiscount

Comment

Querly the customer that would receive a discount because they prefer hot pizzas

Status

Ok

pizzatutorial:Customer(?c) ^ pizzatutorial:numberOfPizzasPurchased(?c, ?np) ^ pizzatutorial:hasSpicinessPreference(?c, pizzatutorial:Hot) ^ swrlb:greaterThan(?np, 1) -> **swrl:select(?c, ?np)**

Cancel

Ok

PizzaTutorial (http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial) : [E:\apache-tomc...

File Edit View Reasoner Tools Refactor Window Help

< >

PizzaTutorial (http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial)

Search...

Individuals by class

OWL Viz

DL Query

OntoGraf

SWRLTab

SQWRLTab

Active ontology

Entities

Classes

Name	Query	Comment
HotDiscountRule	pizzatutorial:Customer(?c) ^ pizzatutorial:numberOfPizzasPurchased(?c, ?np)...	Provide a special discount for...
LessSpicyDiscountR...	pizzatutorial:Customer(?c) ^ pizzatutorial:numberOfPizzasPurchased(?c, ?np)...	Provide a special discount for...
QueryHotDiscount	pizzatutorial:Customer(?c) ^ pizzatutorial:numberOfPizzasPurchased(?c, ?np)...	Querly the customer that woul...

New

Edit

Clone

Delete

SQWRL Queries

OWL 2 RL

Select a SQWRL query from the list above and press the 'Run' button.
If the selected query generates a result, the result will appear in a new sub tab.

The SWRLAPI supports an OWL profile called OWL 2 RL and uses an OWL 2 RL-based reasoner to perform querying.
See the 'OWL 2 RL' subtab for more information on this reasoner.

Executing queries in this tab does not modify the ontology.

Using Drools for query execution.

Run

Reasoner state out of sync with active ontology
 ☒ Show Inferences

PizzaTutorial (http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial) : [E:\apache-tomc...

File Edit View Reasoner Tools Refactor Window Help

< > PizzaTutorial (http://www.semanticweb.org/pizzatutorial/ontologies/2020/PizzaTutorial) Search...

Individuals by class × OWLViz × DL Query × OntoGraf × SWRLTab × SQWRLTab ×

Active ontology × Entities × Classes ×

Name	Query	Comment
HotDiscountRule	pizzatutorial:Customer(?c) ^ pizzatutorial:numberOfPizzasPurchased(?c, ?np)...	Provide a special discount for...
LessSpicyDiscountR...	pizzatutorial:Customer(?c) ^ pizzatutorial:numberOfPizzasPurchased(?c, ?np)...	Provide a special discount for...
QueryHotDiscount	pizzatutorial:Customer(?c) ^ pizzatutorial:numberOfPizzasPurchased(?c, ?np)...	Querly the customer that woul...

New Edit Clone Delete

SQWRL Queries OWL 2 RL QueryHotDiscount

c	np
pizzatutorial:Customer1	2
pizzatutorial:Customer8	2
pizzatutorial:Customer4	3

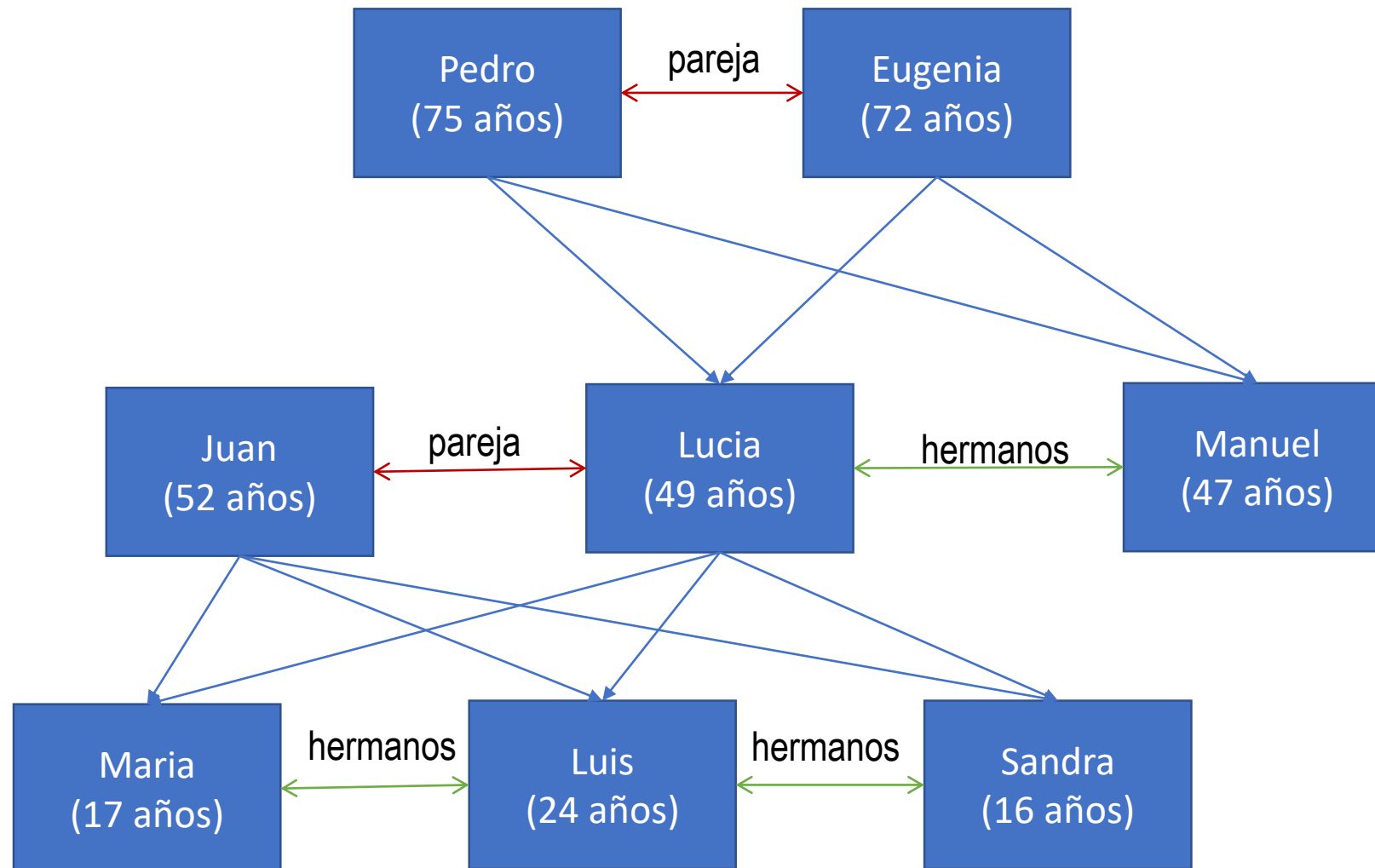
Save as CSV... Rerun Close

Reasoner state out of sync with active ontology ☒ Show Inferences

- Cuando hacemos clic en run, se genera una nueva subpestaña con los resultados. En este caso, hemos lanzado la consulta para averiguar los distintos clientes y el número de pizzas que han consumidos

PRACTICA: PARTE 3 (ENTREGABLE JUNTO CON PARTE 2)

- Elaborar una regla para identificar Abuelos
- Elaborar una regla para identificar Tíos (ver jerarquía en la página siguiente)
- Elaborar una clase para identificar personas que tienen exactamente 2 hijos
- Elaborar una clase para identificar personas que tienen una familia de al menos 3 hijos (FamiliaNumerosa)



FIN DE LA TERCERA SESION