



**CORSO I.F.T.S.
"TECNICHE PER LA PROGETTAZIONE
E LA GESTIONE DI DATABASE"**

Matricola 2014LA0033

DISPENSE DIDATTICHE

MODULO di "DATABASE SEMANTICI"

Ing. Simone Menabeni

Lezione del 26/09/2014



Esercitazione Protegé

Simone Menabeni

Dipartimento di Ingegneria dell'informazione

Università di Firenze

simone.menabeni@unifi.it

DISIT Lab

<http://www.disit.dinfo.unifi.it/>



Creare una nuova ontologia OWL

1. Avviare Protégé
2. Sovrascrivere l'URI di default con <http://www.pizza.com/ontologies/pizza.owl>
3. Si può anche salvare l'ontologia in un file all'interno del PC. Naviga nel tuo hard disk e salva l'ontologia in un file chiamato 'pizza.owl'.



untitled-ontology-20 (<http://www.semanticweb.org/menabeni/ontologies/2014/8/untitled-ontology-20>) : [<http://www.semanticweb.org/menabeni/ontologies/2014/8/untitled-onto...>]

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-20 (<http://www.semanticweb.org/menabeni/ontologies/2014/8/untitled-ontology-20>) Search for entity

Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OWLViz DL Query OntoGraf Ontology Differences SPARQL Query

Ontology header: ⌵ ⌵ ⌵

Ontology IRI <http://www.semanticweb.org/menabeni/ontologies/2014/8/untitled-ontology-20>

Ontology Version IRI e.g. <http://www.semanticweb.org/menabeni/ontologies/2014/8/untitled-ontology-20/1.0.0>

Annotations +

Ontology imports Ontology Prefixes General class axioms

Imported ontologies: ⌵ ⌵ ⌵

Direct Imports +

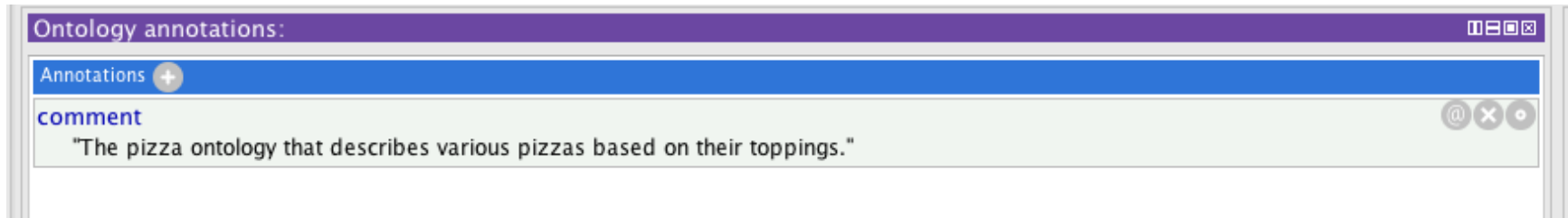
Indirect Imports

Reasoner active Show Inferences

Aggiungere un commento all'ontologia

1. Assicurarsi che il tab `Active Ontology Tab` sia selezionato.
2. Nella finestra `Ontology Annotations`, click sull'icona `Add` vicina ad `Annotations`. Una finestra di modifica apparirà nella tabella. Seleziona 'comment' dalla lista di URI di annotazioni integrate e digita il tuo commento nella casella di testo nella parte destra.
3. Scrivi un commento del tipo `A pizza ontology that describes various pizzas based on their toppings.` e premi OK per memorizzare il commento.

Aggiungere un commento all'ontologia



Ontology annotations: ⏏ ⏏ ⏏ ⏏

Annotations +

comment @ × ⊙

"The pizza ontology that describes various pizzas based on their toppings."


Classi

Add Subclass

Add Sibling Class

Show Usage

Delete Class



Asserted Class Hierarchy: Pizza

- ▼ ● Thing
 - Pizza

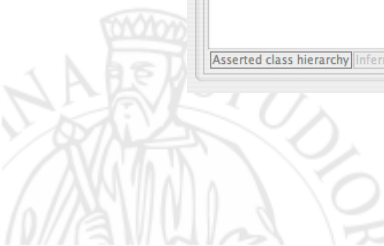
Classi

1. Assicurarsi che sia selezionato il tab 'Classes'.
2. Premere l'icona 'Add'. Questo bottone crea una nuova classe come sottoclasse della classe selezionata (in questo caso vogliamo creare una sottoclasse di Thing).
3. Apparirà una finestra per inserire il nome della classe, scrivere Pizza e premere OK.
4. Ripetere I passi precedenti per aggiungere le classi PizzaTopping e PizzaBase , assicurarsi che la classe Thing sia selezionata prima di premere l'icona Add in modo che le classi siano create come sottoclassi di Thing.





The screenshot shows a web browser window with the URL `http://www.myexample.com/pizza.owl`. The browser's address bar and navigation buttons are visible. Below the browser window is a software interface for editing an ontology. The interface has a menu bar with `File`, `Edit`, `Reasoner`, `Tools`, `Refactor`, `Tabs`, `View`, and `Window`. Below the menu bar is a toolbar with icons for back, forward, and search. The main area of the interface is divided into several panels. At the top, there are tabs for `Active Ontology`, `Entities`, `Classes`, `Object Properties`, `Data Properties`, and `Individuals`. The `Classes` tab is active. On the left, the `Asserted Class Hierarchy` panel shows a tree structure with `Thing` as the root and `Pizza` as a child. Below this panel are tabs for `Asserted class hierarchy` and `Inferred class hierarchy`. On the right, the `Class Annotations` panel is active, showing a list of annotations with a plus sign to add more. Below this panel are tabs for `Class Annotations` and `Class Usage`. At the bottom right, the `Class Description` panel is active, showing sections for `Equivalent classes`, `Superclasses`, `Inherited anonymous classes`, `Instances`, and `Disjoint classes`, each with a plus sign to expand or add content.

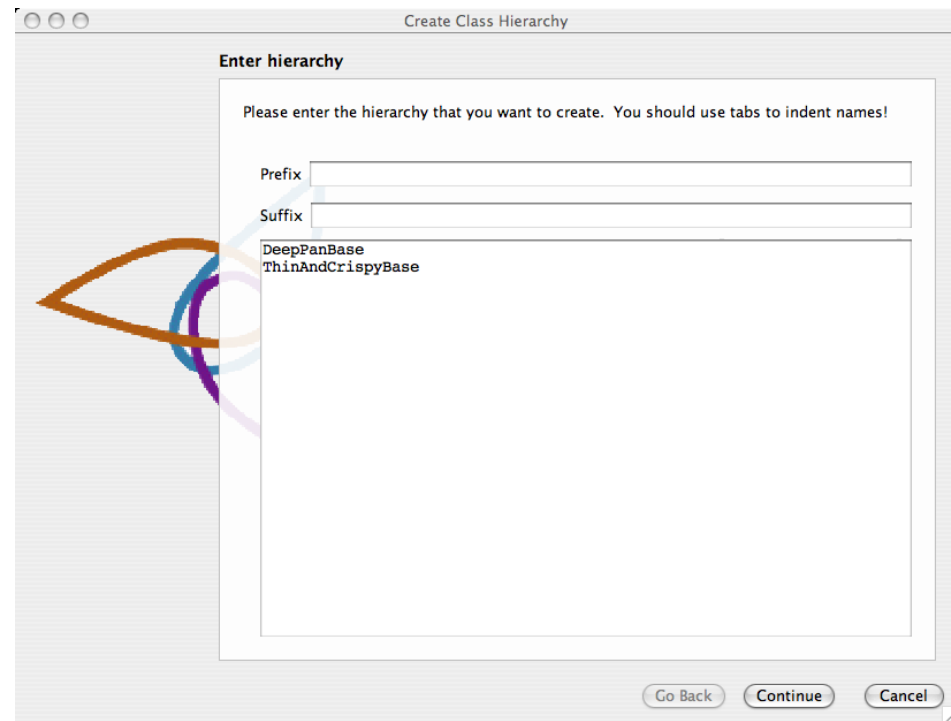
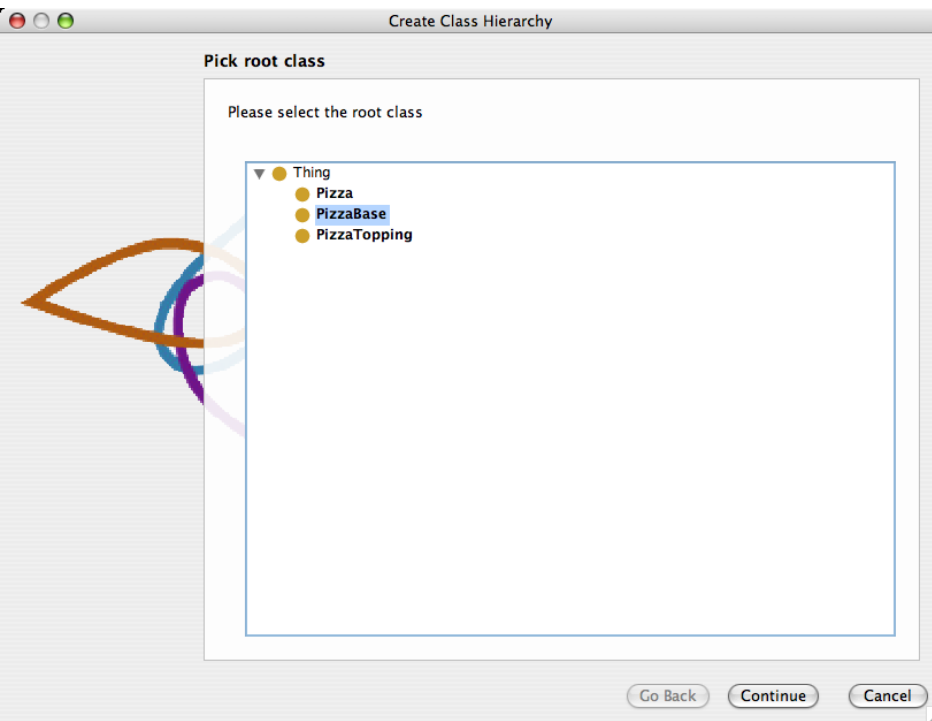


Classi Disgiunte

- Adesso dobbiamo fare in modo che le classi create siano disgiunte, in modo tale che gli individui (o oggetti) non possano essere istanze di più di una di queste classi
 1. Selezionare la classe Pizza nella gerarchia delle classi.
 2. Premere il bottone `Disjoint classes` nella finestra `class description`, questo aprirà una finestra dove possono essere selezionate più classi che devono essere disgiunte (PizzaBase e PizzaTopping). Questo farà sì che queste due classi siano disgiunte da Pizza.

Usare 'Create Class Hierarchy' per creare le classi

1. Selezionare la classe PizzaBase nella gerarchia delle classi.
2. Dal menu Tools sulla barra dei menu di Protegé selezionare **'Create Class Hierarchy...'**.
3. Dal momento che è stata preselezionata la classe PizzaBase , la prima schermata del tool ci mostra che le classi saranno create sotto la classe PizzaBase . Se non avessimo preselezionato PizzaBase prima di avviare il tool, avremmo potuto utilizzare l'albero per selezionare la classe.
4. Premere il bottone 'Next'. Adesso dobbiamo dire al tool che sottoclassi che vogliamo creare. Nella casella di testo grande scrivere il nome della classe (*ThinAndCrispyBase*). Fare la stessa cosa per DeepPanBase.
5. Premere il bottone 'Next'. Il tool controlla che I nomi inseriti aderiscano agli stili dei nomi delle ontologie. Controlla anche che l'unicità dei nomi { se ci sono errori nei nomi delle classi verranno presentati in questa pagina insieme ai suggerimenti per la correzione}.
6. Premere il bottone 'Next'. Assicurarsi che la casella 'Make all new classes disjoint' sia selezionata.



Creare sottoclassi di PizzaTopping

1. Selezionare la classe PizzaTopping.
2. Dal menu Tools sulla barra dei menu di Protegé selezionare **`Create Class Hierarchy...`**.
3. Assicurarsi che PizzaTopping sia selezionata e premere **`Next`**.
4. Quello che vogliamo fare è che tutte le classi dei condimenti abbiano I nomi che finiscono con **‘Topping’**: nel campo **‘Suffix’** scrivere **‘Topping’**.
5. Il tool permette di immettere una gerarchia di classi usando la notazione indentata. Notare che I nomi delle classi devono essere indentati usando I tab.
6. Una volta inserita la lista di classi da aggiungere, premere **‘Next’** e assicurarsi che la casella **‘Make siblings classes disjoint’** sia selezionata.
7. Premere **‘Finish’** per creare le classi. Premere **‘Finish’** un’altra volta per chiudere il tool.

Create Class Hierarchy

Enter hierarchy

Please enter the hierarchy that you want to create. You should use tabs to indent names!

Prefix

Suffix

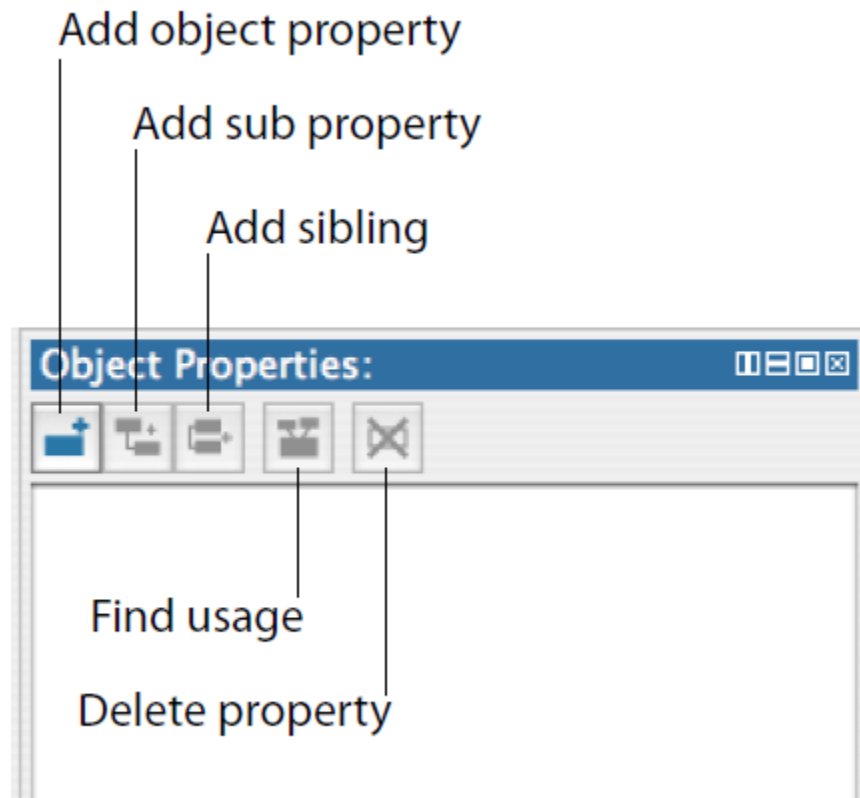
```
Cheese
  Mozzarella
  Parmezan
Meat
  Ham
  Pepperoni
  Salami
  SpicyBeef
Seafood
  Anchovy
  Prawn
  Tuna
Vegetable
  Caper
  Mushroom
  Olive
  Onion
  Pepper
    RedPepper
    GreenPepper
    JalapenoPepper
Tomato
```



Proprietà OWL

- Le Proprietà OWL rappresentano le relazioni.
 1. Andare nel tab **`Object Properties`**. Usare il bottone **`Add Object Property`** per creare una nuova Object.
 2. Attraverso la finestra **`Property Name Dialog`** nominare la proprietà *hasIngredient*.

Proprietà OWL



Creare sotto-proprietà

1. Per creare la proprietà `hasTopping` come una sotto-proprietà della proprietà `hasIngredient`, selezionare la proprietà *hasIngredient*.
2. Premere '**Add subproperty**'. Verrà creata una nuova sotto-proprietà della proprietà *hasIngredient*.
3. Nominare la nuova proprietà *hasTopping*.
4. Ripetere i passi precedenti per la proprietà *hasBase*.



Creare proprietà inverse

1. Usare il bottone **`Add object property`** nel tab **`Object Properties`** per creare una nuova Object Property chiamata *isIngredientOf* (diventerà la proprietà inversa di *hasIngredient*).
2. Premere **`Add`** vicino a **`Inverse properties`** nella finestra **`Property Description`**. Questo farà apparire una finestra nella quale possono essere selezionate le proprietà. Selezionare la proprietà *hasIngredient* e premere **`OK`**. La proprietà *hasIngredient* dovrebbe apparire nella finestra **`Inverse Property`**.
3. Selezionare la proprietà *hasBase*.
4. Premere **`Add`** vicino a **`Inverse properties`** nella finestra **`Property Description`**. Creare una nuova proprietà chiamata *isBaseOf*. Selezionare questa proprietà e premere **`OK`**. Notare che *hasBase* adesso ha una proprietà inversa chiamata *isBaseOf*. Opzionalmente, è possibile posizionare la nuova proprietà *isBaseOf* come sotto-proprietà di *isIngredientOf* (N.B Questo verrà inferito in ogni caso dal reasoner, come vedremo più avanti).
5. Selezionare la proprietà *hasTopping*.
6. Premere **`Add`** icon () vicino a **`Inverse properties`** nella finestra **`Property Description`**. Usare la finestra che appare per creare la proprietà *isToppingOf* e premere **`OK`**.

Description: isToppingOf ☰ ☱ ☲ ☳

Domains (intersection) +

Ranges (intersection) +

Equivalent object properties +

Super properties +

isIngredientOf ✕ ○

Inverse properties +

hasTopping ✕ ○

Disjoint properties +

Property chains +



OWL Object Property Characteristics

- **Proprietà Funzionali** (Se una proprietà è dichiarata funzionale, per un dato individuo, allora ci può essere un solo individuo che può essere collegato all'individuo per mezzo di questa proprietà)
- **Proprietà Funzionali Inverse** (Se una proprietà è dichiarata funzionale inversa significa che la proprietà inversa è funzionale)
- **Proprietà Transitive** (Se una proprietà P è dichiarata transitiva, e la proprietà mette in relazione un individuo A con un individuo B, e anche l'individuo B con l'individuo C, allora possiamo inferire che l'individuo A è in relazione con l'individuo C per mezzo della Proprietà P)
- **Proprietà Simmetriche** (Se una proprietà P è dichiarata simmetrica, e la proprietà mette in relazione l'individuo A con l'individuo B allora l'individuo B è anche in relazione con l'individuo A per mezzo della proprietà P)
- **Proprietà Asimmetriche** (Se una proprietà P è dichiarata simmetrica, e la proprietà mette in relazione l'individuo A con l'individuo B allora l'individuo B non può essere in relazione con l'individuo A per mezzo della proprietà P)
- **Proprietà riflessive** (Una proprietà P è detta riflessiva quando la proprietà mette in relazione l'Individuo A con se stesso)
- **Proprietà Non Riflessive** (Se una proprietà P è dichiarata non riflessiva, può essere descritta come una proprietà che mette in relazione l'individuo A con l'individuo B, dove l'individuo A e l'individuo B non sono uguali)

OWL Object Property Characteristics

Characteristics:

- Functional
- Inverse functional
- Transitive
- Symmetric
- Antisymmetric
- Reflexive
- Irreflexive

Rendere transitiva la proprietà *hasIngredient*

1. Selezionare la proprietà *hasIngredient* nel tab **'Object Properties'**.
2. Selezionare la casella **'Transitive'** nella finestra **'Property Characteristics View'**.
3. Selezionare la proprietà *isIngredientOf*, che è l'inversa di *hasIngredient*. Assicurarsi che la casella **transitive** sia selezionata.

Rendere funzionale la proprietà `hasBase`

1. Selezionare la proprietà *hasBase*.
2. Selezionare la casella **'Functional'** nella finestra **'Property Characteristics View'**.

Specificare il range di hasTopping

1. Selezionare la proprietà hasTopping nel tab `Object Properties`.
2. Premere l'icona `Add` vicino a `Ranges` nella finestra `Property Description`. Apparirà una finestra che permette di selezionare una classe dalla gerarchia dell'ontologia.
3. Selezionare PizzaTopping e premere `OK`. PizzaTopping dovrebbe apparire nella lista Range.

Specificare il range di hasTopping

Description: hasTopping ⏏ ⏏ ⏏ ⏏

Domains (intersection) + _____

Ranges (intersection) + _____

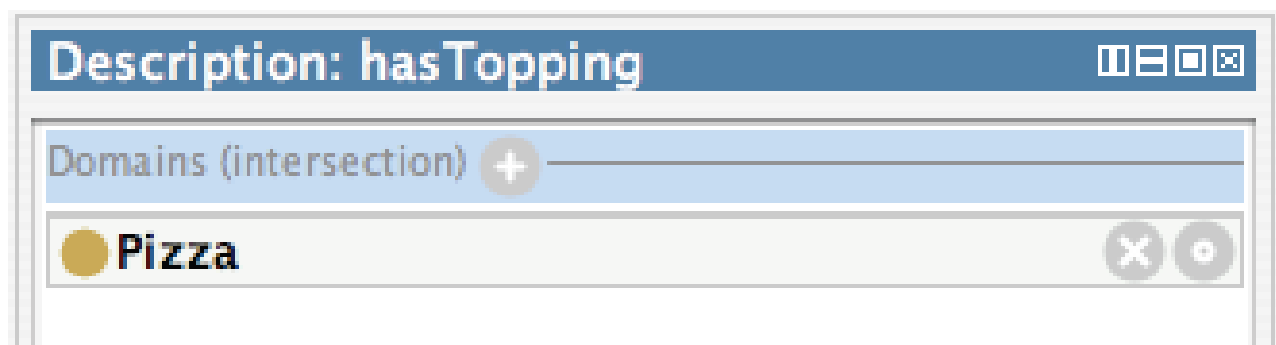
● **PizzaTopping** ✕ ○

Specificare il dominio di hasTopping

1. Selezionare la proprietà hasTopping nel tab `Object Properties`.
2. Premere l'icona `Add` vicino a `Domain` nella finestra `Property Description`. Apparirà una finestra che permette di selezionare una classe dalla gerarchia dell'ontologia.
3. Selezionare Pizza e premere `OK`. Pizza dovrebbe apparire nella lista Domain.



Specificare il dominio di hasTopping



Dominio e Range per la proprietà hasBase

1. Selezionare la proprietà hasBase.
2. Specificare il dominio della proprietà hasBase come Pizza.
3. Specificare il range della proprietà hasBase come PizzaBase.
4. Selezionare la proprietà isBaseOf. Notare che il dominio di isBaseOf è il range della proprietà inversa hasBase e che il range di isBaseOf è il dominio della proprietà inversa hasBase.
5. Impostare il dominio di isBaseOf come PizzaBase.
6. Impostare il range di isBaseOf come Pizza.

Aggiungere una restrizione a Pizza

1. Selezionare Pizza dal tab `Classes`.
2. Selezionare l'icona `Add` vicina a `SubClass of` nella finestra `Class Description View` in modo da creare una condizione necessaria.
3. Ci sono due possibilità: usare il tab `Object Restriction Creator` o scrivere a mano la restrizione nel tab `Class Expression Editor`. Nel primo caso selezionare la proprietà `hasBase`.

Aggiungere una restrizione a Pizza

4. Aggiungere il tipo della restrizione, useremo una restrizione esistenziale quindi selezionare 'some'.
5. Specificare che l'oggetto della restrizione è PizzaBase
6. Premere 'Enter' per creare la restrizione e chiudere la text box. Se tutta l'informazione è stata inserita correttamente la finestra si chiuderà e la restrizione sarà mostrata nella finestra 'Class Description View'.

Creare una sottoclasse di Pizza

1. Selezionare la classe Pizza dal tab `Classes`.
2. Premere l'icona `Add` per creare una nuova sottoclasse di Pizza e chiamarla NamedPizza.
3. Creare una nuova sottoclasse di NamedPizza e chiamarla MargheritaPizza.
4. Aggiungere un commento alla classe MargheritaPizza usando la finestra `Annotations`: A pizza that only has Mozzarella and Tomato toppings

Creare restrizione esistenziale per MargheritaPizza

1. Selezionare la classe MargheritaPizza
2. Usare l'icona 'Add' nella sezione 'SubClass of' della finestra 'Class Description view'.
3. Selezionare hasTopping come la proprietà a cui deve essere applicata la restrizione.
4. Selezionare 'some' per creare una restrizione esistenziale.
5. Selezionare la classe MozzarellaTopping come oggetto della restrizione.
6. Premere 'Enter' per creare la restrizione – se ci sono errori, la restrizione non sarà creata, e l'errore sarà evidenziato in rosso

Creare restrizione esistenziale per MargheritaPizza

1. Selezionare la classe MargheritaPizza
2. Usare l'icona 'Add' nella sezione 'SubClass of' della finestra 'Class Description view'.
3. Selezionare hasTopping come la proprietà a cui deve essere applicata la restrizione.
4. Selezionare 'some' per creare una restrizione esistenziale.
5. Selezionare la classe TomatoTopping come oggetto della restrizione.
6. Premere 'OK' per creare la restrizione

"Equivalent class header" A list of equivalent classes appear here. The classes are sometimes referred to as a Necessary & sufficient criteria.

"Subclass of header" A list of subclasses appear here. These classes are sometimes referred to as Necessary criteria.

"Inferred/Inherited header" A list of conditions that have been inherited from superclasses will be displayed here.



Creare la classe AmericanaPizza

1. Selezionare la classe MargheritaPizza nel tab Classes
2. Selezionare "Duplicate selected class" dal menu `Edit`. Apparirà una finestra per nominare la nuova classe, questa sarà creata esattamente con le stesse condizioni di MarheritaPizza.
3. Assicurarsi che AmericanaPizza sia ancora selezionata, selezionare l'icona 'Add' vicino a `SubClass of` nella finestra 'class description'.
4. Selezionare hasTopping come la proprietà a cui deve essere applicata la restrizione.
5. Selezionare `some` per creare una restrizione esistenziale.
6. Selezionare la classe PepperoniTopping come oggetto della restrizione.
7. Premere `OK` per creare la restrizione

Create an AmericanHotPizza and a SohoPizza

1. Una AmericanHotPizza è la stessa cosa di una AmericanaPizza, ma ha in più il pepe Jalapeno – crearla clonando la classe AmericanaPizza e aggiungere una restrizione esistenziale sulla proprietà hasTopping con oggetto JalapenoPepperTopping.
2. Una SohoPizza è la stessa cosa di una MargheritaPizza, ma ha in più le olive e il parmigiano – crearla clonando la classe MargheritaPizza e aggiungere 2 restrizioni esistenziali sulla proprietà hasTopping con oggetto OliveTopping e ParmezanTopping.

Rendere le sottoclassi di NamedPizza disgiunte

1. Selezionare la classe MargheritaPizza nel tab 'Classes'.
2. Selezionare l'opzione 'Make primitive siblings disjoint' nel menu 'Edit' per rendere le pizze disgiunte fra loro.

Condizioni Necessarie e Sufficienti

1. Selezionare Pizza dal tab 'Classes'
2. Premere l'icona 'Add' per creare una sottoclasse di Pizza. Chiamarla CheesyPizza.
3. Assicurarsi che CheesyPizza sia selezionata. Selezionare l'icona 'Add' vicina a 'SubClass Of'.
4. Selezionare hasTopping come la proprietà a cui si deve applicare la restrizione
5. Selezionare 'some' per creare la restrizione esistenziale
6. Inserire CheeseTopping come oggetto della restrizione e premere OK per chiudere la finestra e creare la restrizione
7. Assicurarsi che CheesyPizza sia selezionata
8. Nel menu 'Edit' selezionare 'Convert to defined class'.

Condizioni Necessarie e Sufficienti

Description: CheesyPizza

Equivalent classes +

Superclasses +

- **Pizza**
- **hasTopping some CheeseTopping**

Inherited anonymous classes

- **hasBase some PizzaBase**

Description: CheesyPizza

Equivalent classes +

- **Pizza**
and hasTopping some CheeseTopping

Superclasses +

Inherited anonymous classes

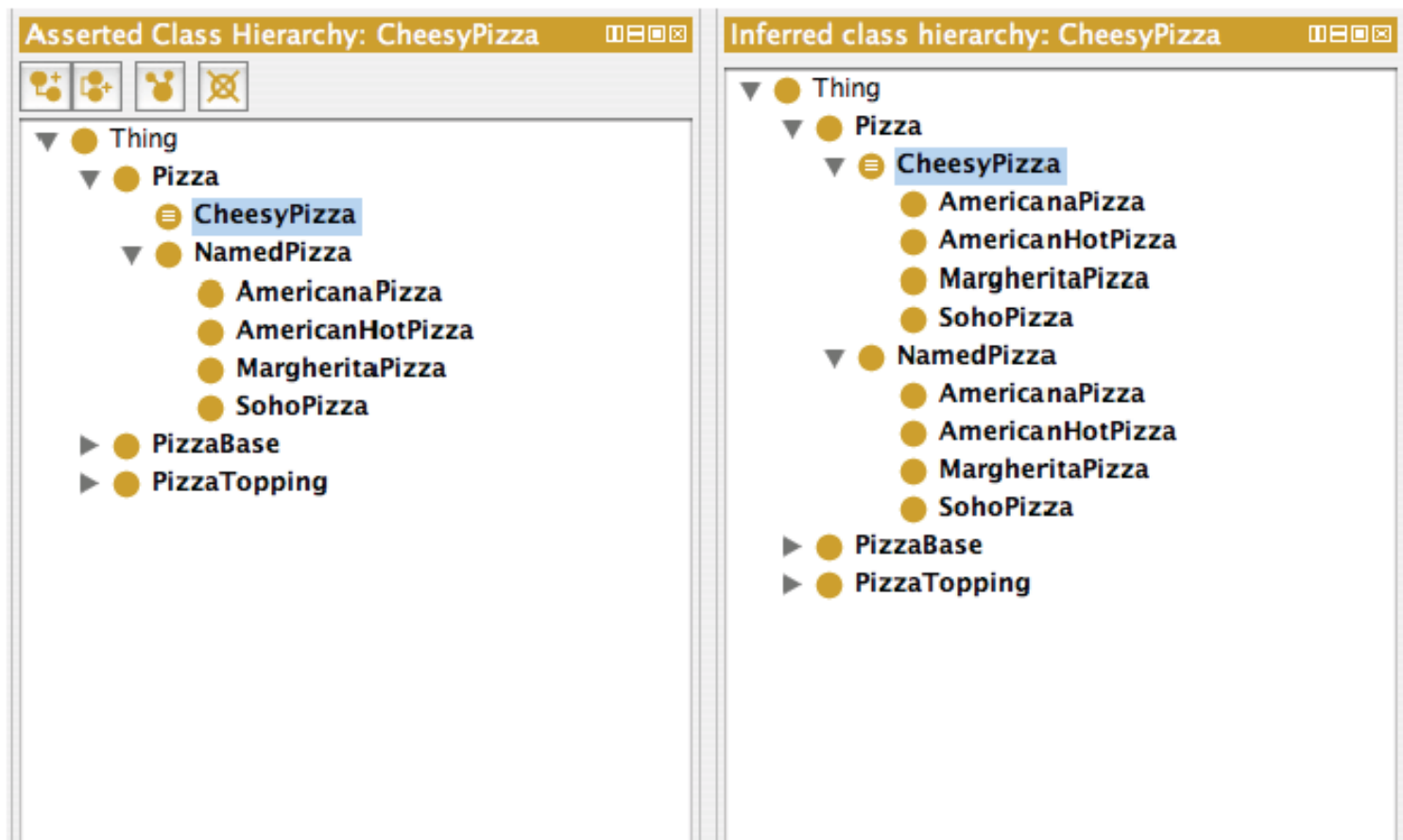
- **hasBase some PizzaBase**



Invocare il Reasoner

- Protege 4 permette di utilizzare differenti reasoner OWL, quello incluso in Protege si chiama Fact++. L'ontologia può essere inviata al reasoner per elaborare la gerarchia e per controllare la consistenza logica dell'ontologia.

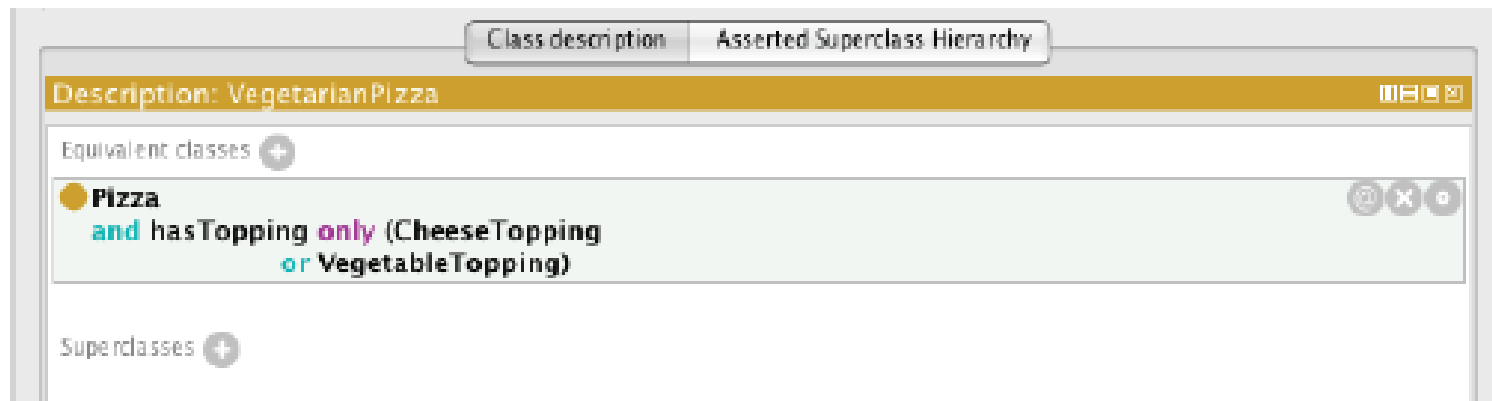
Invocare il Reasoner



Restrizioni Universali

1. Creare una sottoclasse di Pizza e chiamarla VegetarianPizza.
2. Assicurarsi che VegetarianPizza sia selezionata, premere sull'icona 'Add' vicina a 'SubClass Of'.
3. Inserire hasTopping come la proprietà a cui applicare la restrizione.
4. Inserire 'only' per creare una restrizione universale quantificata
5. Per l'oggetto selezionare 'CheeseTopping o VegetableTopping'. Per fare ciò dobbiamo inserire le classi tra parentesi. Aprire una parentesi seguita dalla classe CheeseTopping. Adesso dobbiamo utilizzare l'operatore unionOf tra le classi. Inserire la classe VegetableTopping. Dovreste aver "hasTopping only (CheeseTopping or VegetableTopping)" nella text box.
6. Premere OK per chiudere la finestra e creare la restrizione
7. Assicurarsi che VegetarianPizza sia selezionata
8. Dal Menu 'Edit' selezionare 'Convert to defined class'.

Restrizioni Universali



The screenshot shows a software interface with two tabs: "Class description" (selected) and "Asserted Superclass Hierarchy". The "Class description" tab displays the following information:

- Description:** VegetarianPizza
- Equivalent classes:** +
 - Pizza** (indicated by a yellow dot) and hasTopping **only** (CheeseTopping or VegetableTopping)
- Superclasses:** +

Restrizioni Universali

1. Selezionare la classe MargheritaPizza.
2. Premere l'icona 'Add' vicina alla sezione 'Subclass Of' della finestra 'Class Description'.
3. Selezionare hasTopping come proprietà su cui applicare la restrizione.
4. Selezionare 'only' per creare la restrizione universale.
5. Aprire la parentesi e digitare 'MozzarellaTopping or TomatoTopping', chiudere la parentesi.
6. Premere 'OK' per creare la restrizione e aggiungerla alla classe MargheritaPizza.

Restrizioni Universali

Description: MargheritaPizza ⏏

Equivalent classes +

Superclasses +

● NamedPizza	@ X ○
● hasTopping some MozzarellaTopping	@ X ○
● hasTopping some TomatoTopping	@ X ○
● hasTopping only (MozzarellaTopping or TomatoTopping)	@ X ○

Restrizioni Universali

1. Selezionare la classe SohoPizza.
2. Premere l'icona 'Add' vicina alla sezione 'Subclass Of' della finestra 'Class Description'.
3. Selezionare hasTopping come proprietà su cui applicare la restrizione.
4. Selezionare 'only' per creare la restrizione universale.
5. Aprire la parentesi e digitare 'MozzarellaTopping or TomatoTopping or ParmezanTopping or OliveTopping', chiudere la parentesi.
6. Premere 'OK' per creare la restrizione e aggiungerla alla classe SohoPizza.

Restrizioni Universali

1. Selezionare la classe AmericanaPizza.
2. Nella finestra 'class description', selezionare una delle restrizioni sulla proprietà hasTopping, per esempio 'hasTopping some TomatoTopping'.
3. Cliccare con il tasto destro sulla restrizione e selezionare 'Create closure axiom'.

Restrizioni di cardinalità

- In OWL è possibile descrivere la classe degli individui che hanno almeno, al massimo o esattamente un numero specifico di relazioni con altri individui o valori di datatype.
- Per una data proprietà P , una restrizione di cardinalità minima specifica il numero minimo di relazioni P che un individuo deve avere.
- Una restrizione di cardinalità massima specifica il numero massimo di relazioni P che un individuo deve avere.
- Una restrizione di cardinalità specifica il numero esatto di relazioni P che un individuo deve avere.

Restrizioni di cardinalità

1. Andare nel tab Classes e selezionare la classe Pizza.
2. Creare una sottoclasse di Pizza chiamata InterestingPizza.
3. Premere l'icona 'Add' nella sezione `SubClass Of' della finestra 'class description'.
4. Selezionare hasTopping come la proprietà su cui applicare la restrizione.
5. Selezionare `min' per creare una restrizione di cardinalità minima.
6. Specificare una cardinalità minima di tre scrivendo 3 nella text box.
7. Premere `Enter' per chiudere la finestra e creare la restrizione.
8. Selezionare l'opzione `Convert to defined class' nel menu `Edit'. La sezione `SubClass Of' dovrebbe essere vuota mentre la sezione `Equivalent classes' dovrebbe mostrare 'Pizza' e 'hasTopping min 3'.

Restrizioni di cardinalità

Description: InterestingPizza

Equivalent classes +

● Pizza
and hasTopping min 3

Superclasses +

Restrizioni di cardinalità

1. Creare una sottoclasse di Pizza chiamata FourCheesePizza
2. Premere l'icona Add nella sezione `SubClass Of`.
3. Selezionare hasTopping come proprietà
4. Selezionare exactly per creare una restrizione di cardinalità esatta
5. Specificare una cardinalità di quattro scrivendo 4 nella text box
6. Scrivere CheeseTopping per specificare il tipo di condimento
7. Premere OK per creare la restrizione

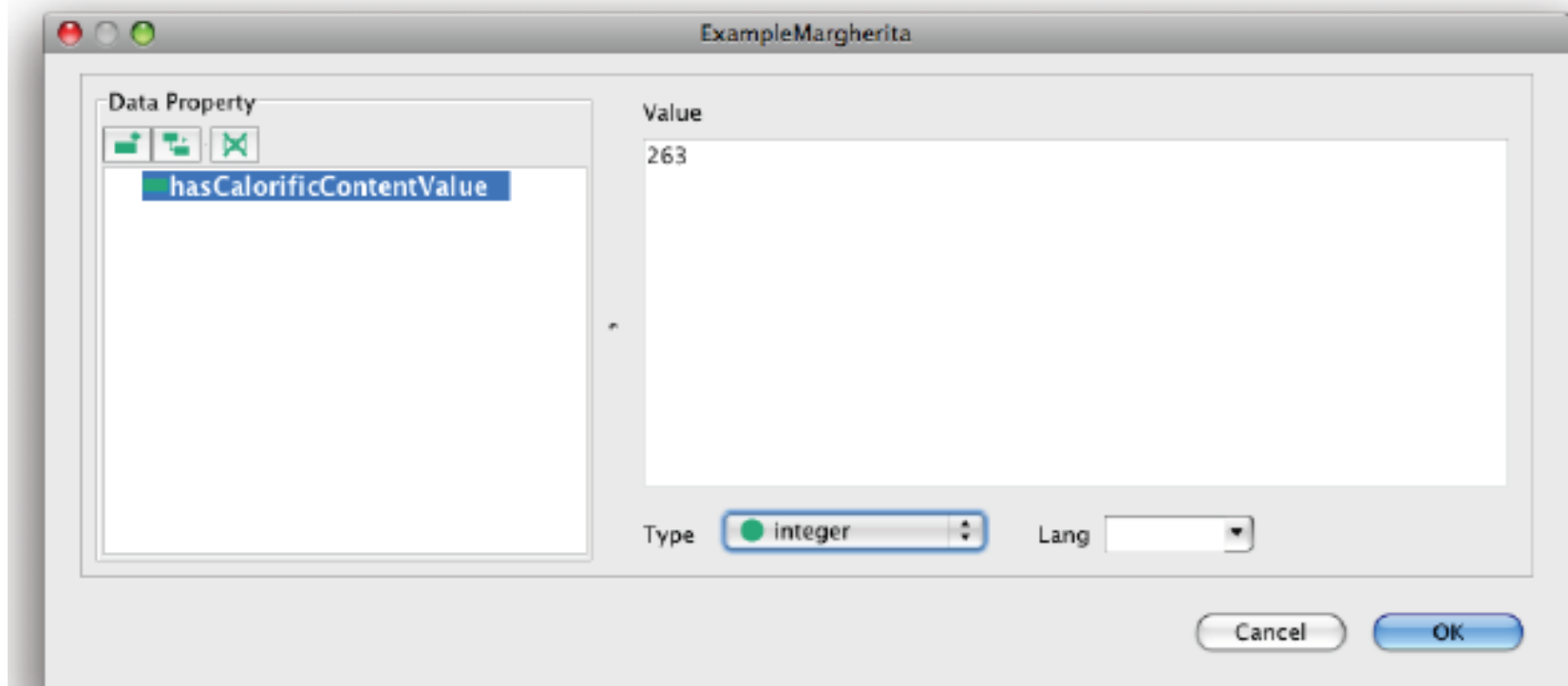
Datatype Properties

- Le Datatype Properties collegano un individuo ad un valore di un datatype definito in uno XML Schema o a un literal RDF.
- In altre parole, descrivono le relazioni tra un individuo e i valori dei dati.

Datatype Properties

1. Andare nel tab `Datatype Properties`. Usare il bottone `Add Datatype Property` per creare una nuova Datatype property chiamata `hasCalorificContentValue`.
2. Assicurarsi che sia selezionato il tab `Entities Tab` o `Individuals Tab` e che sia visibile la finestra `Individuals view`.
3. Premere il bottone `Add individual` e creare un individuo chiamato `ExampleMargherita`
4. Nella finestra `Individual Description` aggiungere il tipo `MargheritaPizza`. Si può fare sia nella `Class hierarchy` o nella `Class expression editor`.
5. Nella finestra `Property assertions` aggiungere una `Data property assertion` e nella finestra di dialog assicurarsi che la proprietà `hasCalorificContentValue` sia selezionata, `integer` sia selezionato come tipo e il valore `263` sia inserito.
6. Creare molti altri esempi di individui di pizza con differenti calorie inclusa un'istanza di `QuattroFormaggio` con `723` calorie.

Datatype Properties



ExampleMargherita

Data Property

- hasCalorificContentValue

Value

263

Type integer

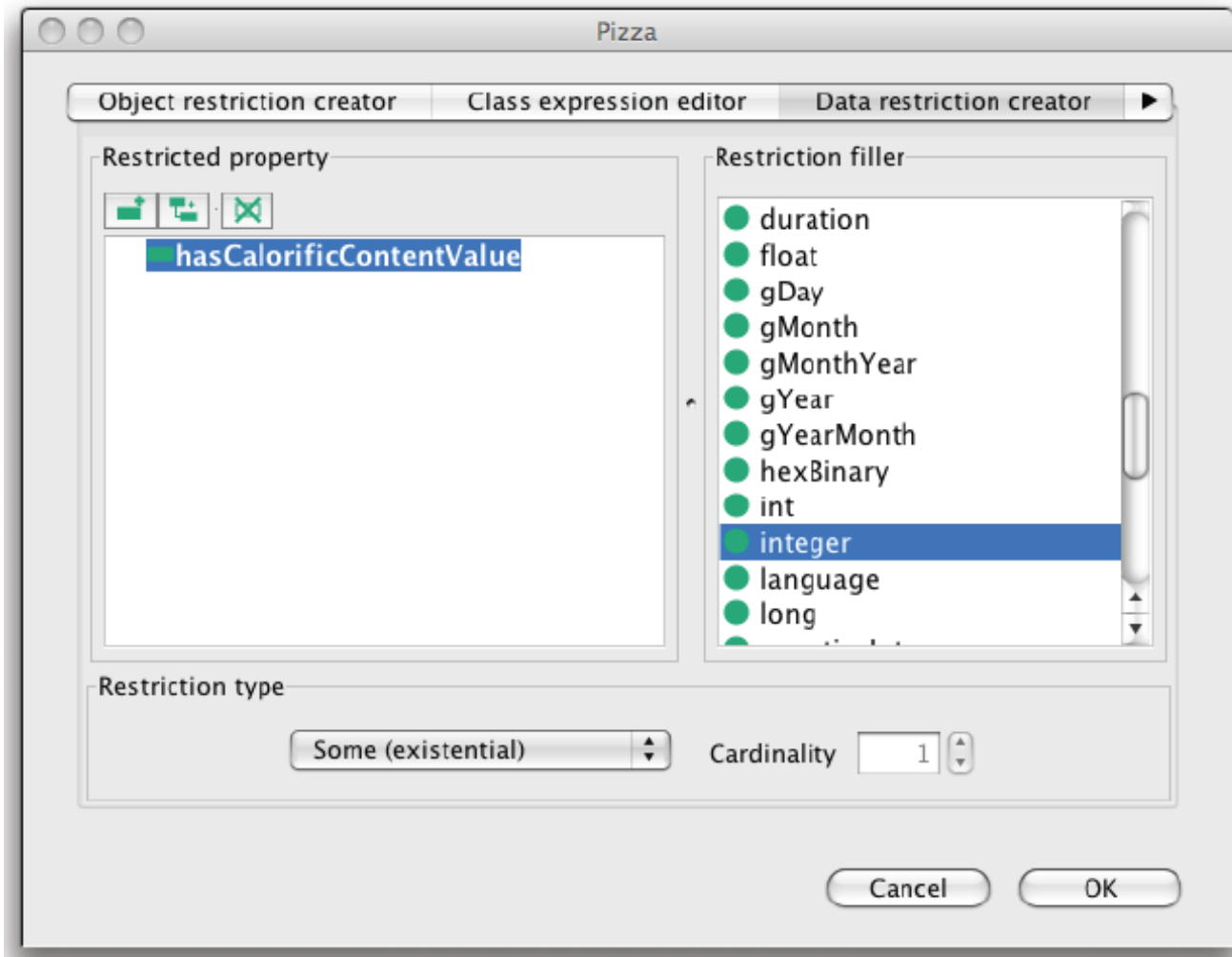
Lang

Cancel OK

Datatype Properties - Restrizioni

1. Assicurarsi che il tab 'Entities' o 'Classes' sia selezionato
2. Selezionare Pizza e nella finestra 'Class Description' aggiungere una sottoclasse.
3. Nella finestra che si apre selezionare 'Data restriction creator'.
4. Assicurarsi che il tipo di restrizione sia posizionato su 'some'.
5. Selezionare hasCalorificContentValue come la proprietà su cui applicare la restrizione.
6. Scegliere il datatype 'integer'
7. Premere 'OK'.

Datatype Properties - Restrizioni





Datatype Properties - Range


1. Assicurarsi che il tab 'Classes' o 'Entities' sia selezionato.
2. Creare una sottoclasse di Pizza chiamata HighCaloriePizza.
3. Nella finestra 'Class Description' premere l'icona 'Add' nella sezione 'SubClass Of' per aggiungere una nuova restrizione
4. Nella finestra 'Class expression editor', scrivere 'hasCalorificContentValue some integer[>= 400]' e premere 'OK'
5. Convertire la classe in una defined class ('Ctrl-D', or 'Command-D' on a Mac).
6. Creare una classe LowCaloriePizza nello stesso modo, ma definirla come equivalente a 'Pizza and hasCalorificContentValue some integer[< 400]'.

Datatype Properties - Range

Description: HighCaloriePizza

Equivalent classes 

-  **Pizza**
and hasCalorificContentValue **some** integer[>= "400"^^integer]

Superclasses 

Datatype Properties

1. Selezionare un reasoner dal 'Reasoner menu' o premere 'Classify' se ne è già stato selezionato uno. Il reasoner dovrebbe classificare e mostrare la gerarchia di classi inferita.
2. Selezionare HighCaloriePizza. Dovrebbe essere possibile vedere le inferenze mostrate nella finestra 'Class Description' in giallo con un bordo tratteggiato.
3. Controllare la sezione 'Members'. Dovrebbe includere l'istanza di 'QuattroFormaggio' e forse altri individui che sono stati definiti avere un valore di calorie uguale o superiore a 400.
4. Selezionare LowCaloriePizza. Controllare la sezione 'Members'. Dovrebbe includere l'istanza di 'ExampleMargherita' e forse altri individui che sono stati definiti avere calorie con un valore minore di 400.

Datatype Properties

Description: HighCaloriePizza

Equivalent classes +

- Pizza and hasCalorificContentValue some integer[>= "400"^^integer]

Superclasses +

- Pizza

Inherited anonymous classes

- hasCalorificContentValue some integer
- hasBase some PizzaBase

Members +

- ◆ ExampleAmericanHot
- ◆ ExampleFruttiDiMare
- ◆ ExampleQuattroFormaggio

Disjoint classes +



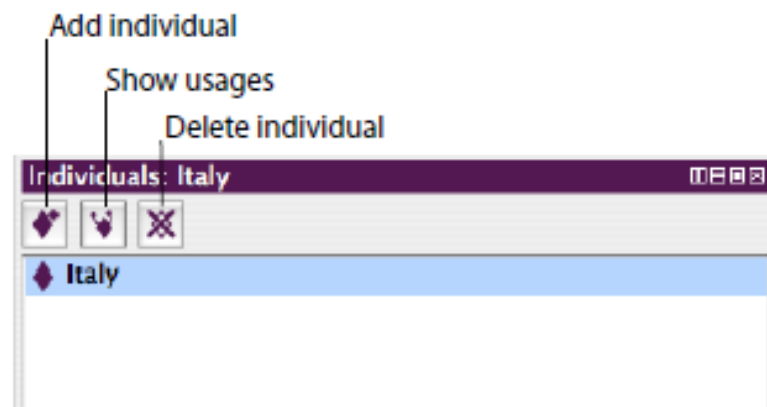
Datatype Properties - Functional

1. Andare al tab `Datatype Properties` e selezionare `hasCalorificContentValue`
2. Nel pannello `Data Type Characteristics`, selezionare la casella `functional`.
3. Testare che funziona creando un individuo di pizza che ha 2 valori di calorie. Questo dovrebbe far diventare l'ontologia inconsistente.

Creare Individui

1. Creare Country come sottoclasse di Thing.
2. Andare al tab `Individuals`.
3. Premere il bottone `Add individual`.
4. Chiamare il nuovo individuo Italy.
5. Selezionare l'icona `Add` vicina a `Types` dalla finestra `Individual Types` posizionata al centro del tab Individual. Scegliere la classe Country, questo farà diventare Italy un individuo della classe Country.
6. Usare I passi precedenti per creare altri individui che siano membri della classe Country chiamati America, England, France, e Germany.

Creare Individui



Restrizione hasValue

1. Andare al tab `Object Properties`. Creare una nuova object property e chiamarla hasCountryOfOrigin.
2. Andare al tab `Classes` e selezionare la classe MozzarellaTopping.
3. Selezionare l'icona `Add` nella sezione `SubClass Of`.
4. Selezionare hasCountryOfOrigin come la proprietà su cui applicare la restrizione.
5. Selezionare `value` come tipo di restrizione.
6. Immettere Italy come l'individuo che completa la restrizione.
7. Premere `Enter` per chiudere la finestra e creare la restrizione.

Restrizione hasValue

Description: MozzarellaTopping

Equivalent classes +

Superclasses +

- CheeseTopping
- hasSpiciness **some** Mild
- hasCountryOfOrigin **value** Italy