Introduction

Food Recommendation Ontology

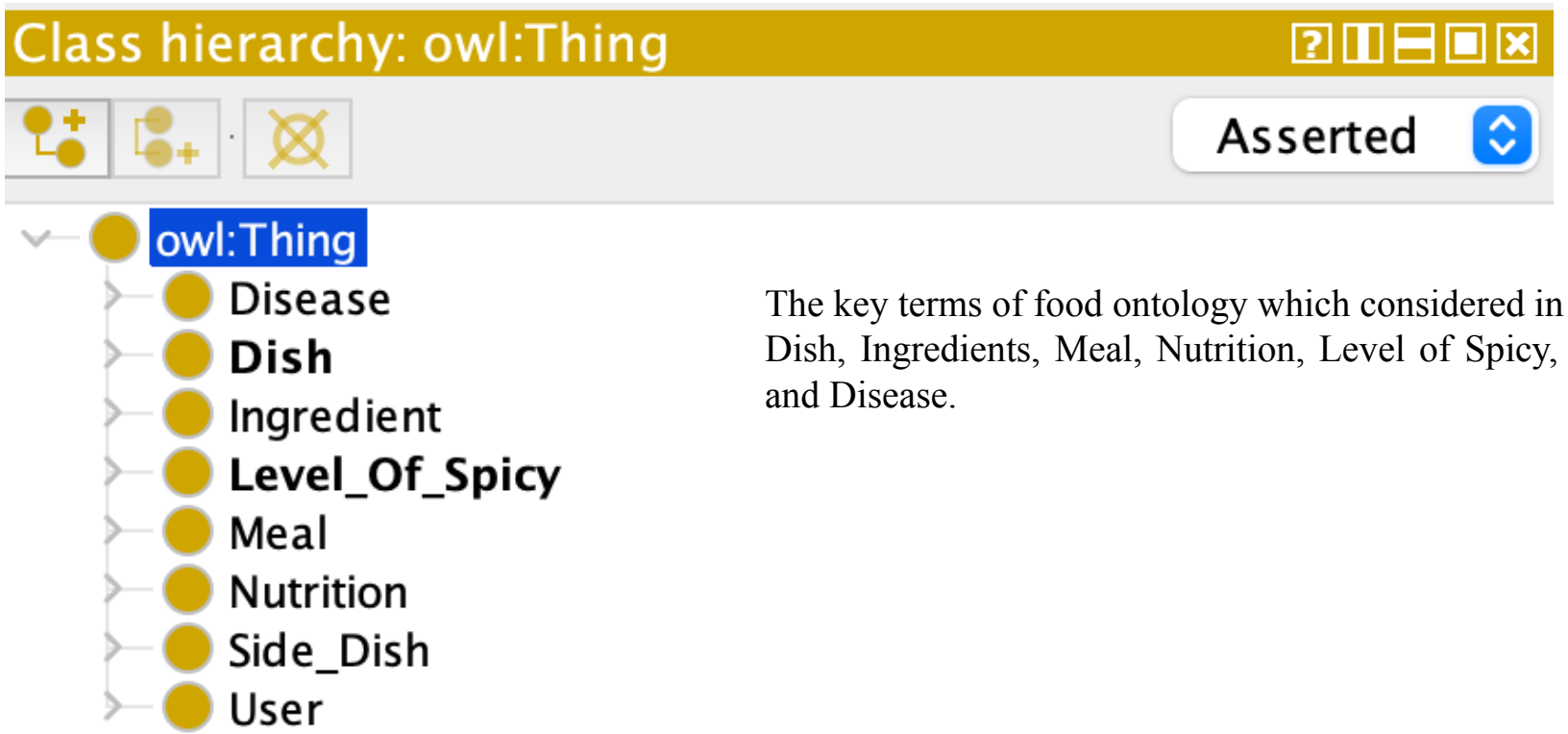
- Goal:

The project designed to personalized nutritional recommendations which effectively support healthy eating, through a food ontology.

Expected queries to be answered:

- Recommend the user dishes based on the ingredients
- Recommend the user dishes based on the nutrition.
- Recommend the user dishes based on amount of calorie.
- Recommend the user dishes based on different categories such as, high protein dishes, vegan dishes, complex dishes, and different meals.
- Recommend the user dishes based on their allergy.
- Recommend the user dishes based on their disease.

Top Classes



The key terms of food ontology which considered in this work: Dish, Ingredients, Meal, Nutrition, Level of Spicy, Side Dish, User and Disease.

Figure 1

Dish Class

- A dish is made by mixing different ingredients.
- Named_Dish is a subclass of Dish with 27 individuals.
- There are 15 defined class based on different features like: high calorie dish, vegan dish or dishes with balanced amount of salt.

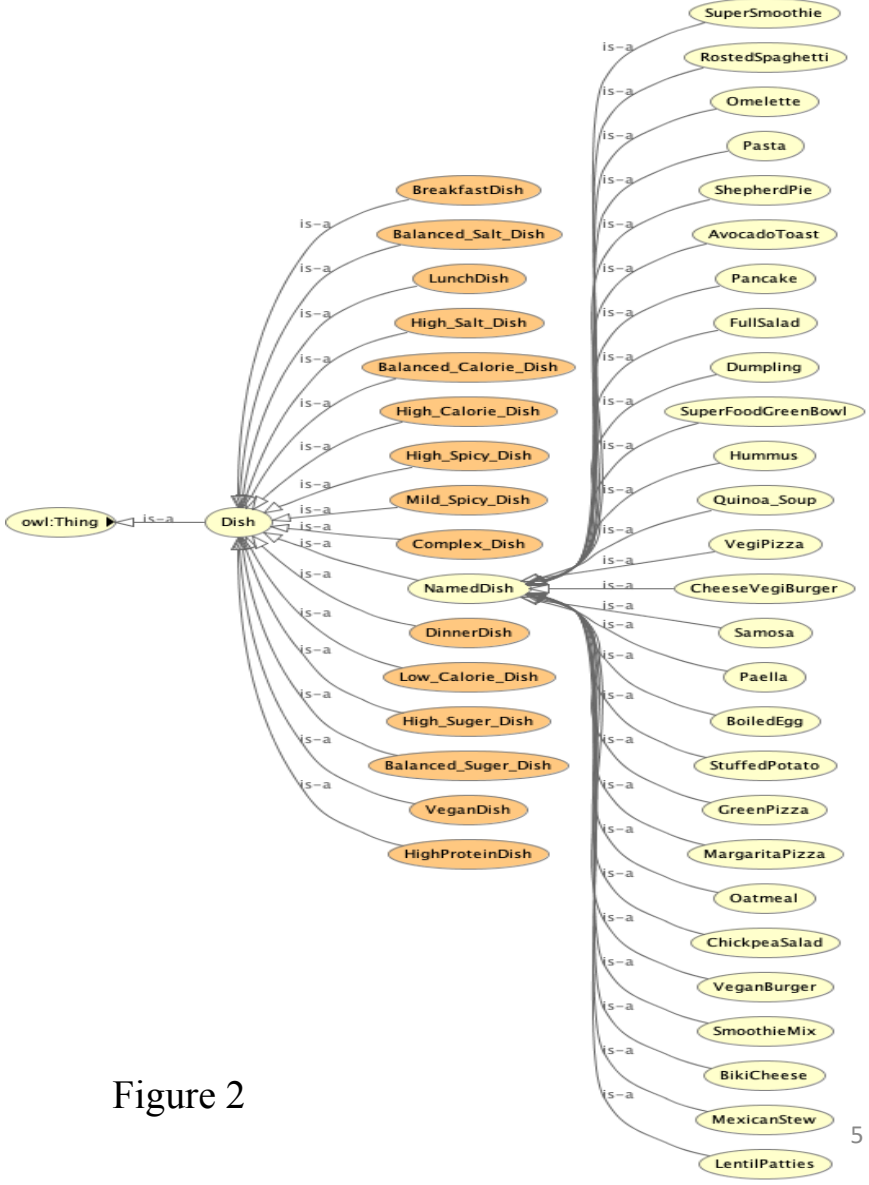


Figure 2

Nutrition Class

- Nutrition is superclass of:
  - carbohydrates,
  - fat,
  - fiber,
  - protein,
  - minerals
  - and vitamins (vitamin A, B2, B5, B6, B12, C, D, E, K).

Figure 3

Ingredient class

- Ingredient class is described as a substance used to prepare a dish
- Is divided in two main categories including animal-based ingredient and plant-based ingredient and they have their own subclasses.
- For instance, in plane-based ingredient, there are class of fruit, grain, herbs and spices, nuts, oil and vinegar and vegetables.

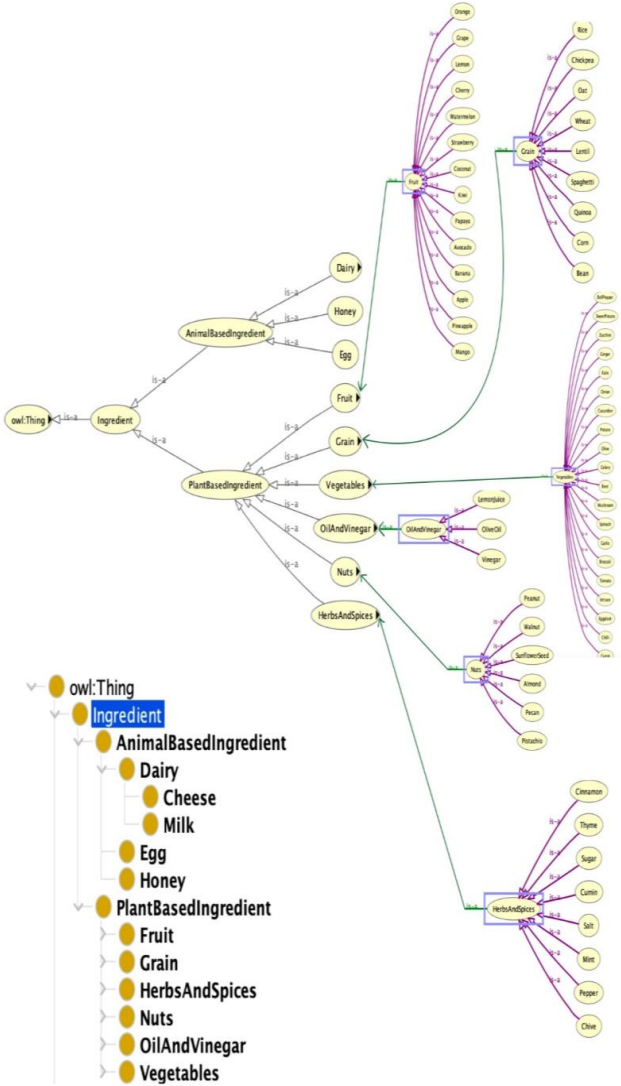


Figure 4

Other classes

a) Side Dish class
b) Level of Spicy
c) Disease class
d) User class
e) Meal class

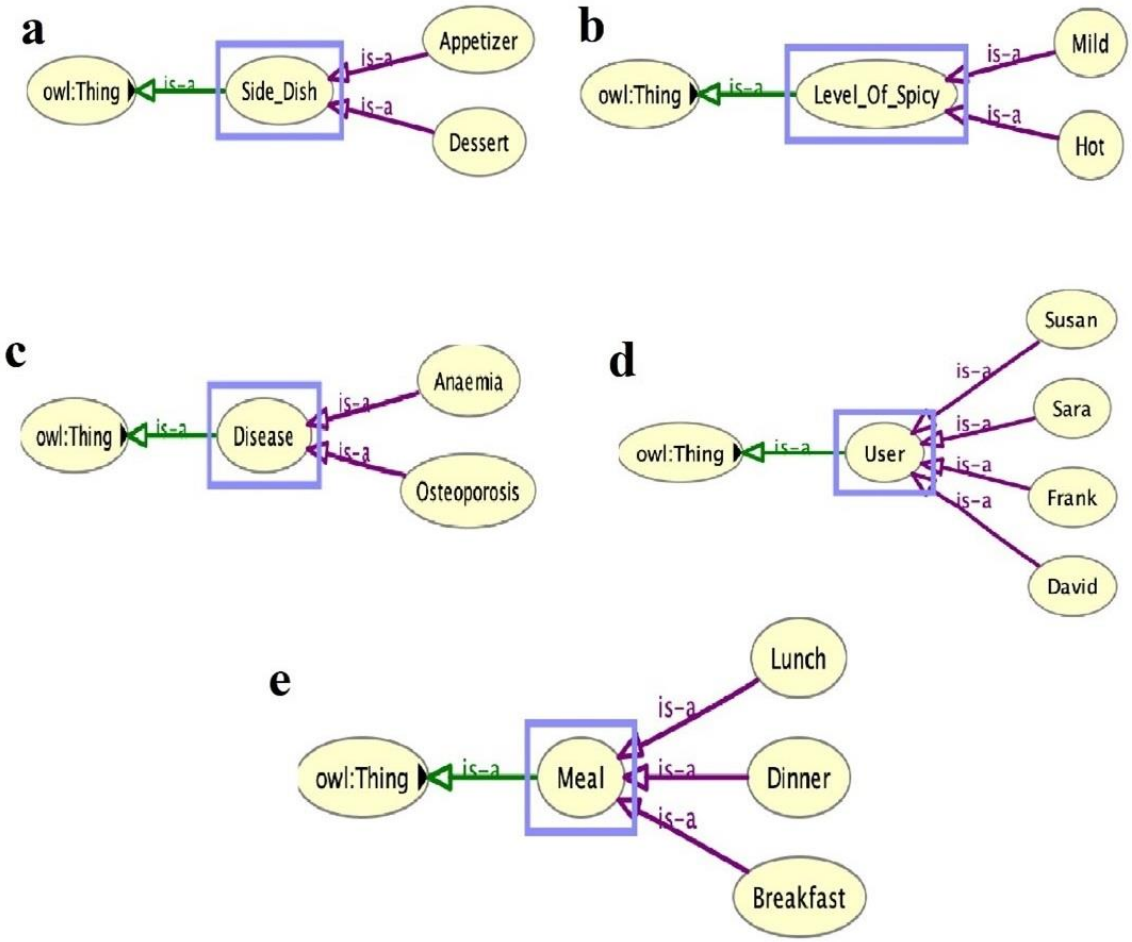


Figure 5

Applying Disjoint

- These classes are disjoint, so that an individual (or object) can not be an instance of more than one of these
- For example: Vitamin and carbohydrate are disjoint. It simply means an instance can't be at the same time vitamin and carbohydrate

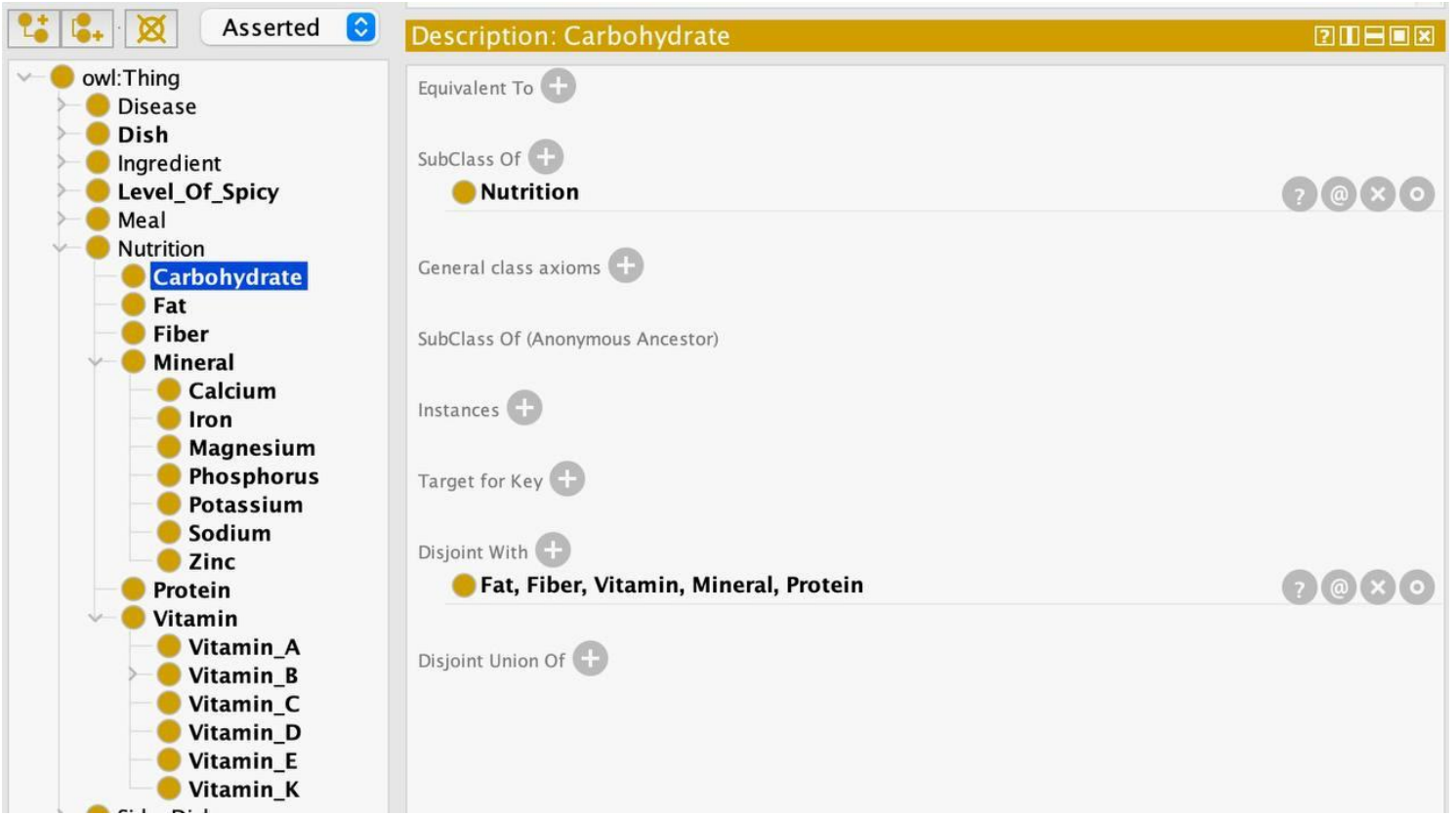


Figure 6

Relationship between different classes

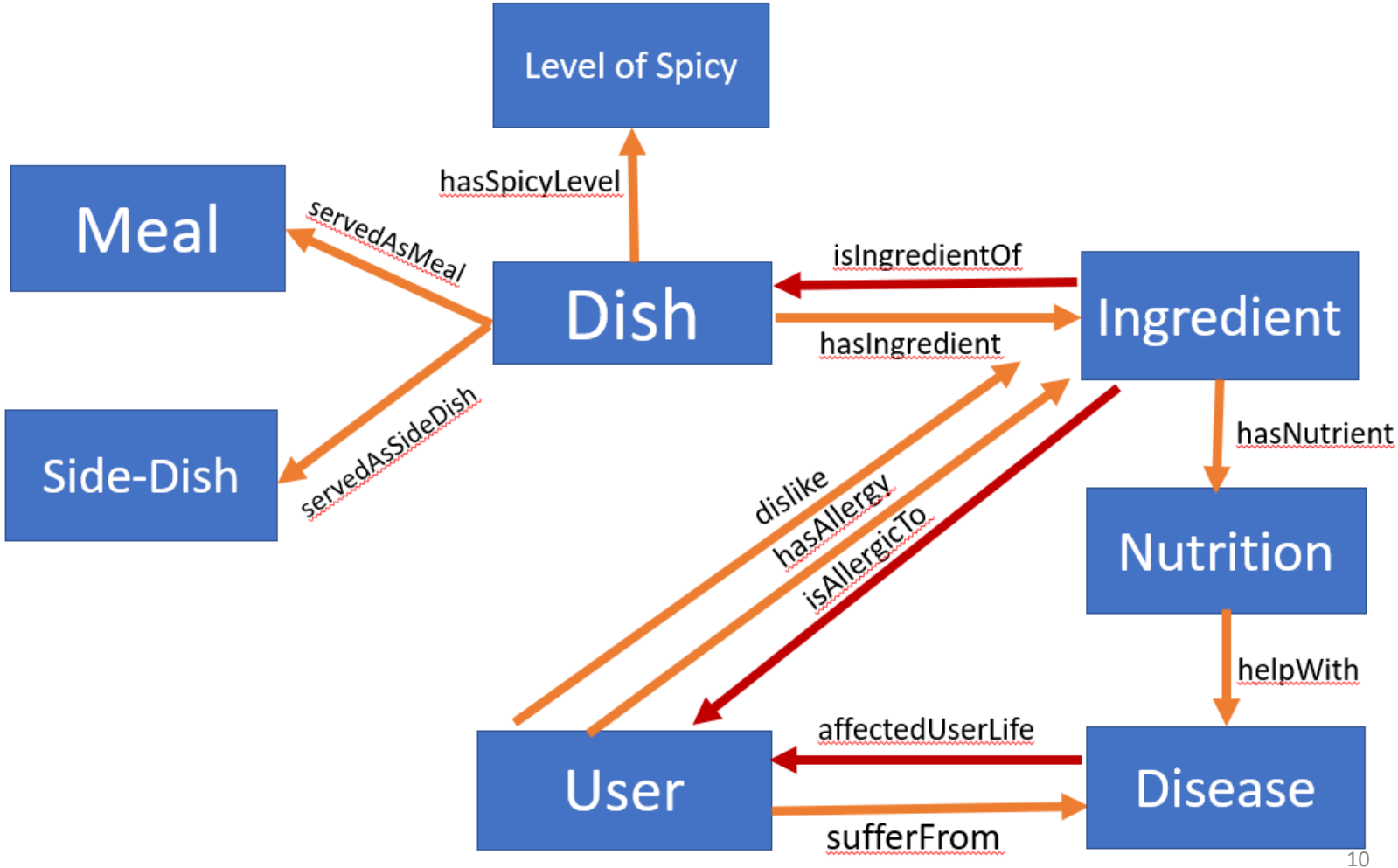


Figure 7

Assigning domain and range to object properties

	Object property	Domains	Ranges	Inverse of	Characteristics
1	**hasIngredient**	**Dish**	**Ingredient**	**isIngredientOf**	-
2	**isIngredientOf**	**Ingredients**	**Dish**	**hasIngredient**	-
To make dishes we need ingredient, so each dish has ingredient and ingredients are used to make dishes (Inverse Relationship)					
3	**hasNutrient**	**Ingredients**	**Nutrition**	**-**	**Transitive**
Ingredients have nutrition and dishes have ingredient, therefore dishes have nutrition (Transitive Property)					
4	**hasSpicyLevel**	**Dish**	**Level_Of_Spicy**	**-**	**Functional**
In this ontology each dish has a level of spicy as hot or mild, this property is functional due to a dish can't be hot and mild at the same time(Functional Property)					
5	**servedAsMeal**	**Dish**	**Meal**	-	
Each main dish can be served as Breakfast or Lunch or dinner, furthermore this property isn't functional due to same dish could be served as lunch or dinner or breakfast.					
6	**servedAsSideDish**	**Dish**	**Side_Dish**	**-**	**Functional**
Dishes could serve as side dish and a side dish could be appetiser or either dessert, so it's functional property.					
7	**dislikeIngredient**	**User**	**Ingredient**	**-**	**Transitive**
8	**likeIngredient**	**User**	**Ingredient**	**isLikedBy**	**Transitive**
User may like or dislike ingredient and dishes have ingredient, so users may like or dislike dishes (Transitive Property)					
9	**hasAllergyTo**	**User**	**Ingredient**	**isAllergicTo**	**-**
10	**isAllergicTo**	**Ingredient**	**User**	**hasAllergyTo**	**-**
Users may have allergy to some ingredients and ingredients may are allergic to users (Inverse Relationship).					
11	**affectUserLife**	**Disease**	**User**	**sufferFromDisease**	-
12	**sufferFromDisease**	**User**	**Disease**	**affectUserLife**	-
13	**helpWithDisease**	**Nutrition**	**Disease**	**-**	-
User would suffer from some diseases and diseases may affect user's life (Inverse Relationship). Users need special nutrition to improve their health condition against specified disease**.**					

Data properties and relation

	Top Data Properties	Characteristic	Type
1	hasCalorieValue	Functional	Integer
2	hasSaltAmountGram	Functional	Integer
3	hasSugarAmountGram	Functional	Integer

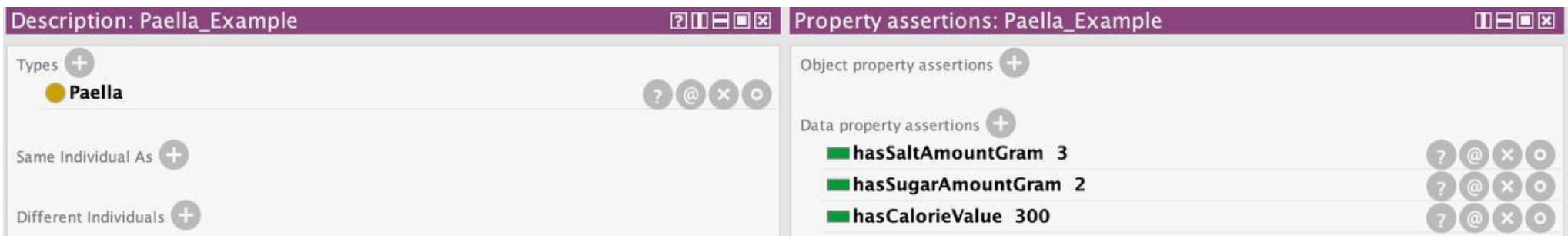


Figure 8. An example of using data property to one instance of class Dish

Property Restriction

These classes are made by using property restriction such as, quantifier restriction (Existential and Universal), cardinality restriction and hasValue restriction.

- **Complex_Dish:** Dish and (hasIngredient min 10 owl:Thing)
- **VeganDish:** Dish and (hasIngredient only PlantBasedIngredient)
- **HighProteineDish:** Dish and ((hasIngredient some Bean) or (hasIngredient some Chickpea) or (hasIngredient some Lentil) or (hasIngredient some Quinoa))
- **Mild_Spicy_Dish:** Dish and (hasIngredient some (hasSpicyLevel some Mild)
- **High_Spicy_Dish:** Dish and (hasIngredient some (hasSpicyLevel some Hot))
- **Balanced_Calorie_Dish:** Dish and ((hasCalorieValue some xsd:integer[>= 250]) and (hasCalorieValue some xsd:integer[< 400]))
- **High_Calorie_Dish:** Dish and (hasCalorieValue some xsd:integer[>= 400])
- **Low_Calorie_Dish:** Dish and (hasCalorieValue some xsd:integer[< 250])
- **High_Salt_Dish:** Dish and (hasSaltAmountGram some xsd:integer[> 2])
- **Balanced_Salt_Dish:** Dish and (hasSaltAmountGram some xsd:integer[<= 2])
- **High_Sugar_Dish:** Dish and (hasSugarAmountGram some xsd:integer[>= 4])
- **Balanced_Sugar_Dish:** Dish and (hasSugarAmountGram some xsd:integer[< 4])

Figure 9. Using cardinality restriction to make Complex_Dish class

Figure 10. Using universal restriction to make VeganDish class

Figure 11. Using hasValue restriction to make Mild_Spicy_Dish class

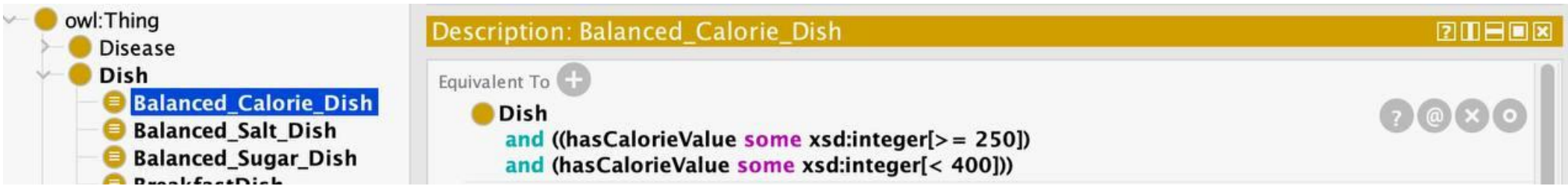


Figure 12. Using data property to make Balanced_Calorie_Dish class

Applying Closure Axiom

In the figure below, the closure axiom consists of universal restriction that works with other properties to identify for example Hummus can only be made with Chickpea, Olive oil, Pepper and Salt.

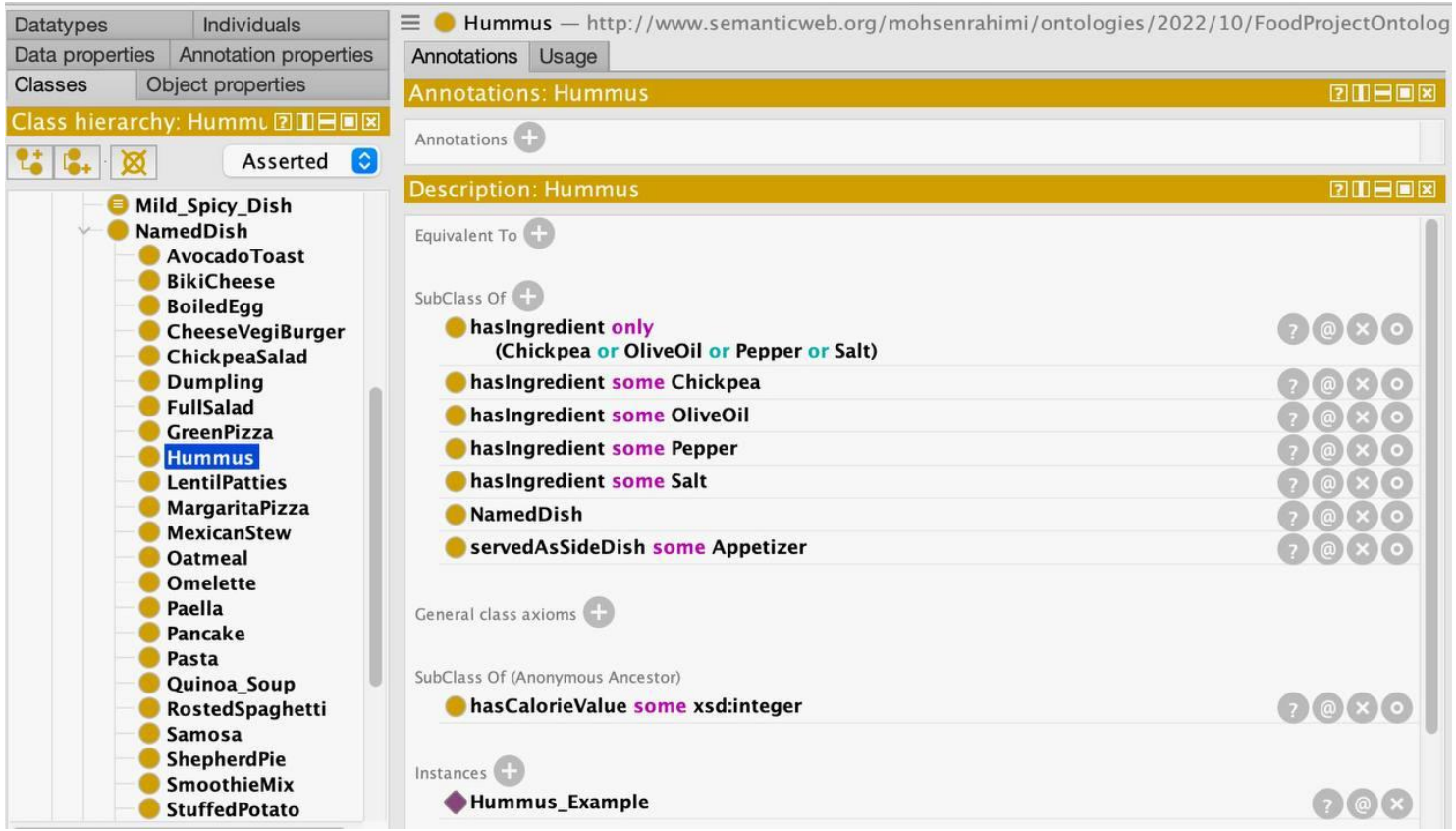


Figure 13

Change a Primitive class to a Defined class

By adding sufficient condition to necessary condition a primitive class (Figure 14) will change to a defined class (Figure 15).
Another example in Figure 16.

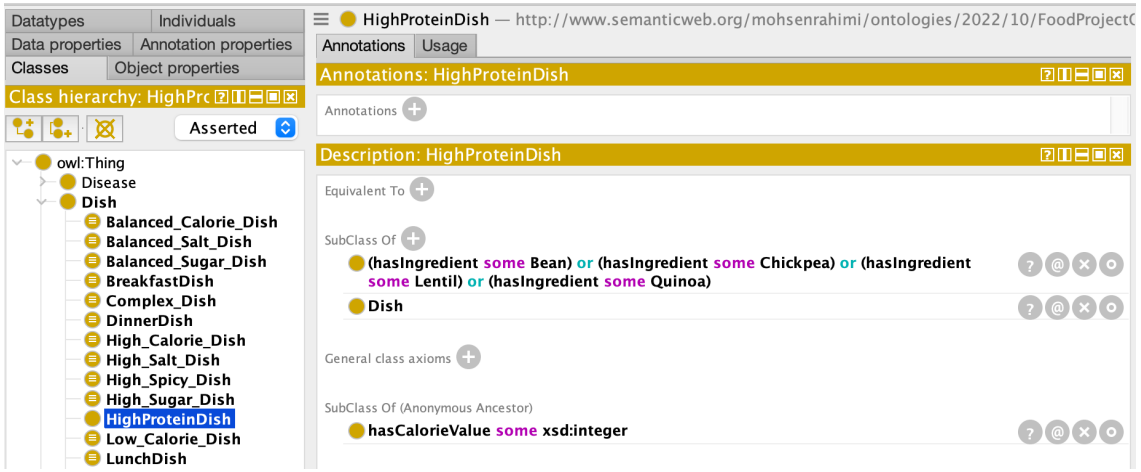


Figure 14

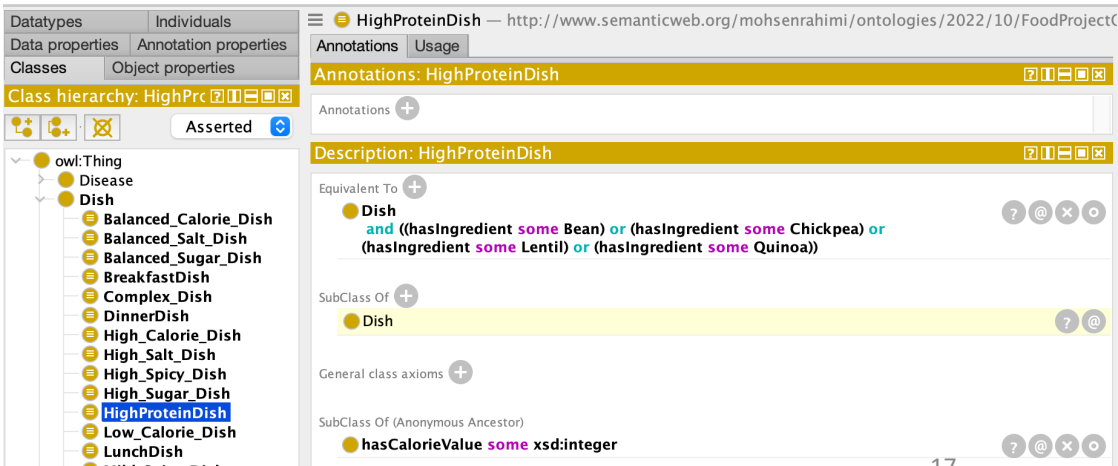


Figure 15

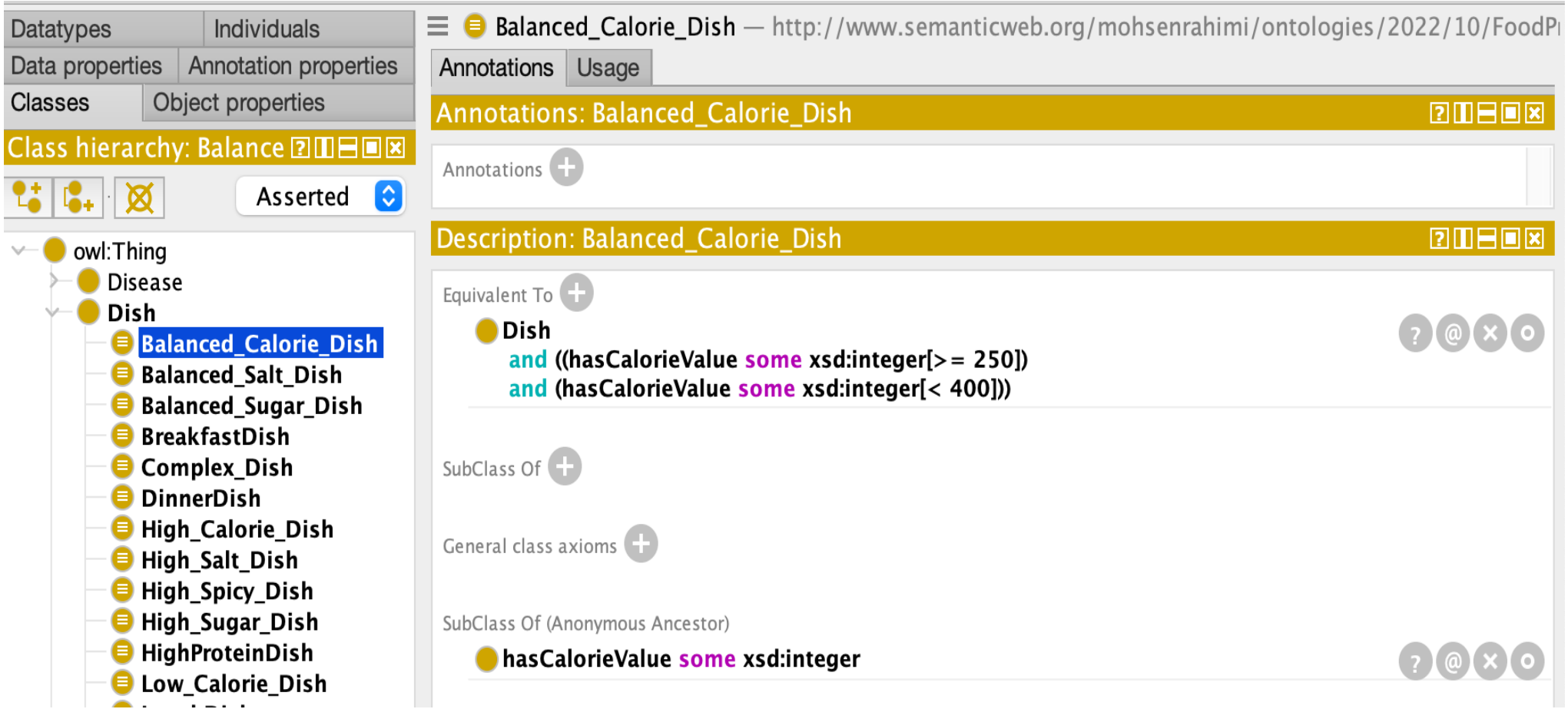

Datatypes
Individuals
Data properties
Annotation properties
Classes
Object properties
Class hierarchy: Balance
Asserted
owl:Thing
Disease
Dish
Balanced_Calorie_Dish
Balanced_Salt_Dish
Balanced_Sugar_Dish
BreakfastDish
Complex_Dish
DinnerDish
High_Calorie_Dish
High_Salt_Dish
High_Spicy_Dish
High_Sugar_Dish
HighProteinDish
Low_Calorie_Dish
Balanced_Calorie_Dish — http://www.semanticweb.org/mohsenrahimi/ontologies/2022/10/FoodP
Annotations
Usage
Annotations: Balanced_Calorie_Dish
Annotations
Description: Balanced_Calorie_Dish
Equivalent To
Dish
and ((hasCalorieValue some xsd:integer[>= 250])
and (hasCalorieValue some xsd:integer[< 400]))
SubClass Of
General class axioms
SubClass Of (Anonymous Ancestor)
hasCalorieValue some xsd:integer

Figure 16

Using the reasoner

- The reasoner can show us if all the statements and definitions in the ontology are consistent.
- It help to maintain the hierarchy by recognizing which concept fit which definitions.
- More examples in Figure 18 and 19.

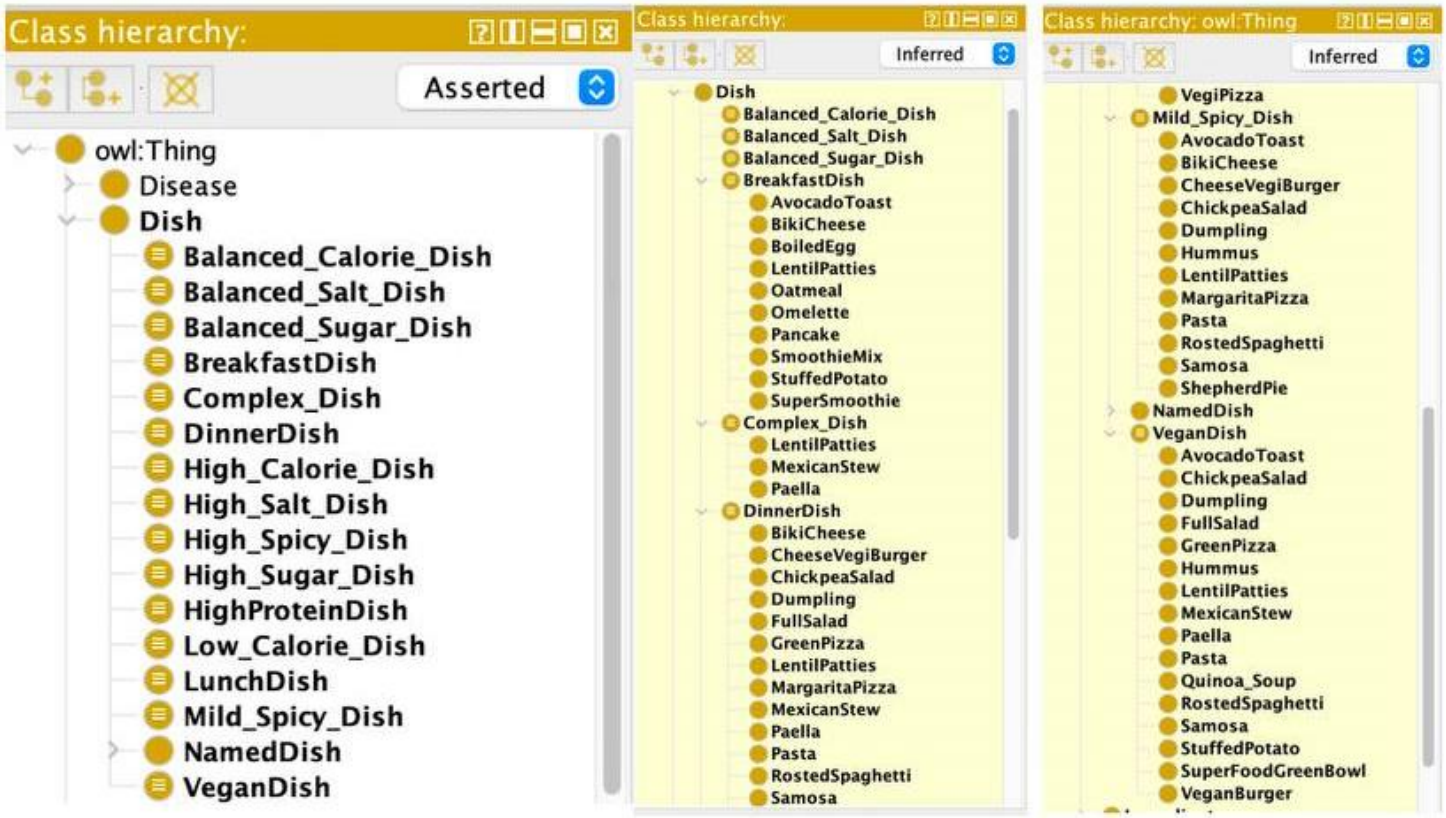


Figure 17. The different between Asserted window and Inferred window

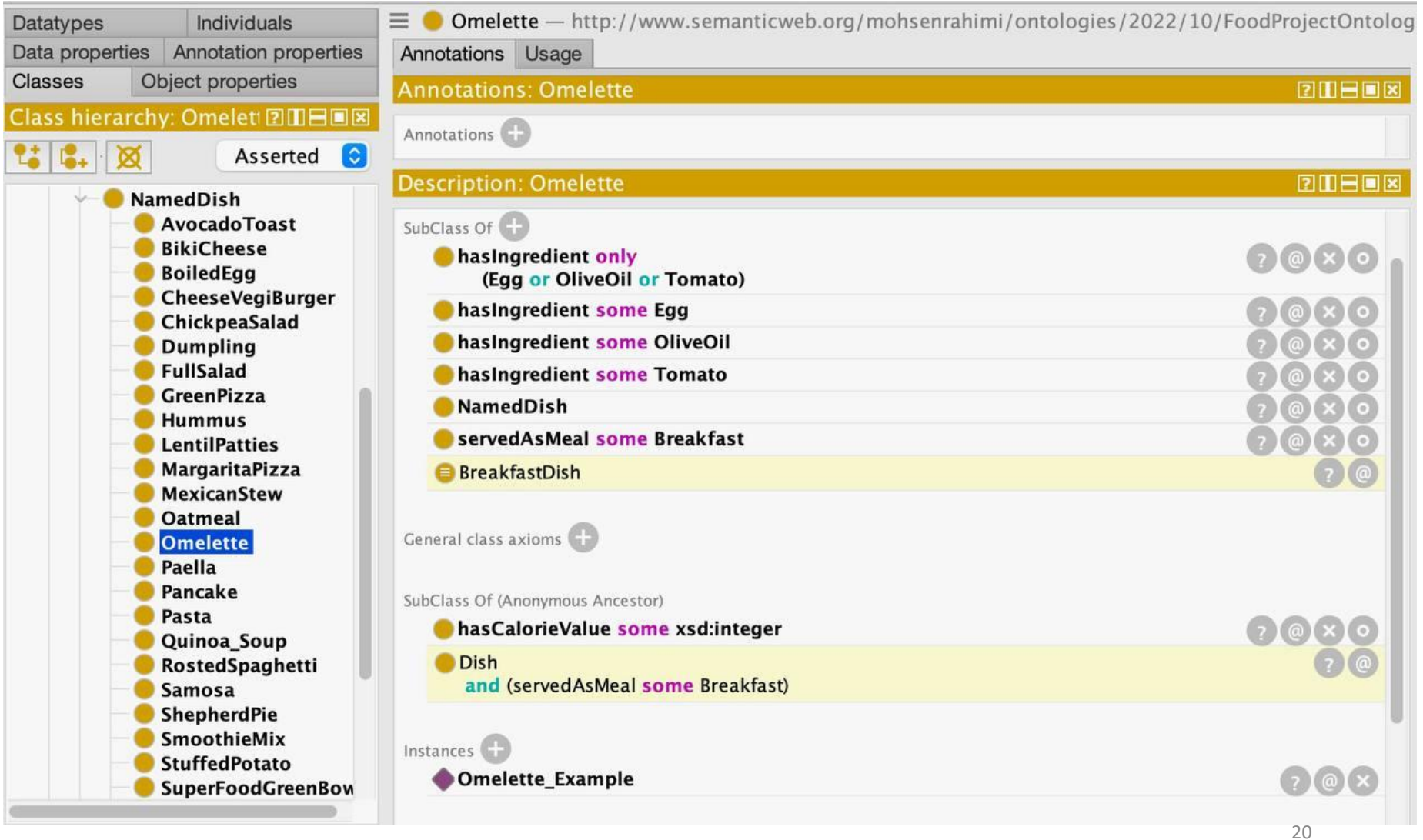

Datatypes
Individuals
Data properties
Annotation properties
Classes
Object properties
Class hierarchy: Omelet
Asserted
NamedDish
AvocadoToast
BikiCheese
BoiledEgg
CheeseVegiBurger
ChickpeaSalad
Dumpling
FullSalad
GreenPizza
Hummus
LentilPatties
MargaritaPizza
MexicanStew
Oatmeal
Omelette
Paella
Pancake
Pasta
Quinoa_Soup
RostedSpaghetti
Samosa
ShepherdPie
SmoothieMix
StuffedPotato
SuperFoodGreenBow
Omelette — http://www.semanticweb.org/mohsenrahimi/ontologies/2022/10/FoodProjectOntolog
Annotations
Usage
Annotations: Omelette
Annotations
Description: Omelette
SubClass Of
hasIngredient only
(Egg or OliveOil or Tomato)
hasIngredient some Egg
hasIngredient some OliveOil
hasIngredient some Tomato
NamedDish
servedAsMeal some Breakfast
BreakfastDish
General class axioms
SubClass Of (Anonymous Ancestor)
hasCalorieValue some xsd:integer
Dish
and (servedAsMeal some Breakfast)
Instances
Omelette_Example

Figure 18

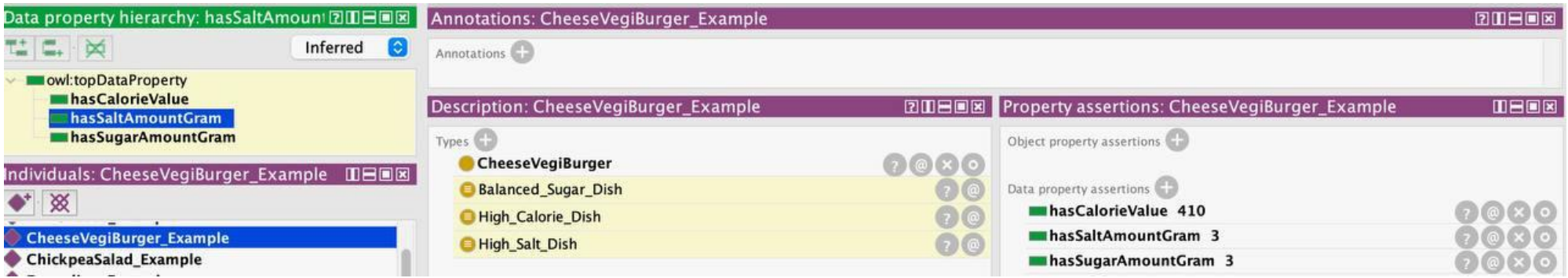

Data property hierarchy: hasSaltAmoun
Inferred
owl:topDataProperty
hasCalorieValue
hasSaltAmountGram
hasSugarAmountGram
Individuals: CheeseVegiBurger_Example
CheeseVegiBurger_Example
ChickpeaSalad_Example
Annotations: CheeseVegiBurger_Example
Annotations
Description: CheeseVegiBurger_Example
Types
CheeseVegiBurger
Balanced_Sugar_Dish
High_Calorie_Dish
High_Salt_Dish
Property assertions: CheeseVegiBurger_Example
Object property assertions
Data property assertions
hasCalorieValue 410
hasSaltAmountGram 3
hasSugarAmountGram 3

Figure 19

Illustrating the subclasses of diseases, minerals that are recommended to use for the specified diseases and then the food that contain those specific minerals.

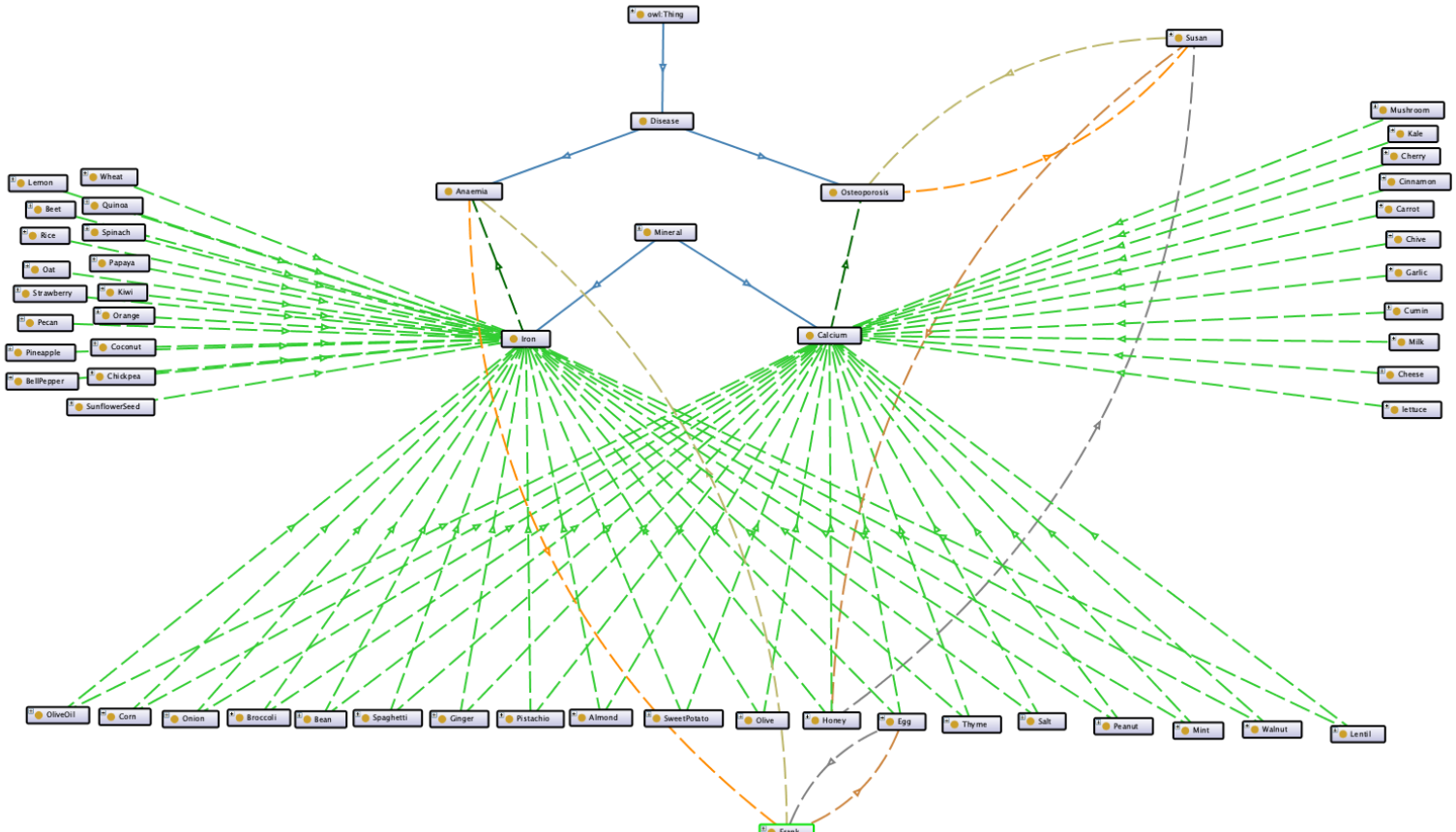


Figure 20

Representing the Meal and showing Breakfast is a subclass of Meal and BreakfastDish. Moreover, showing the ingredients and nutrients which exist in some type of breakfasts.

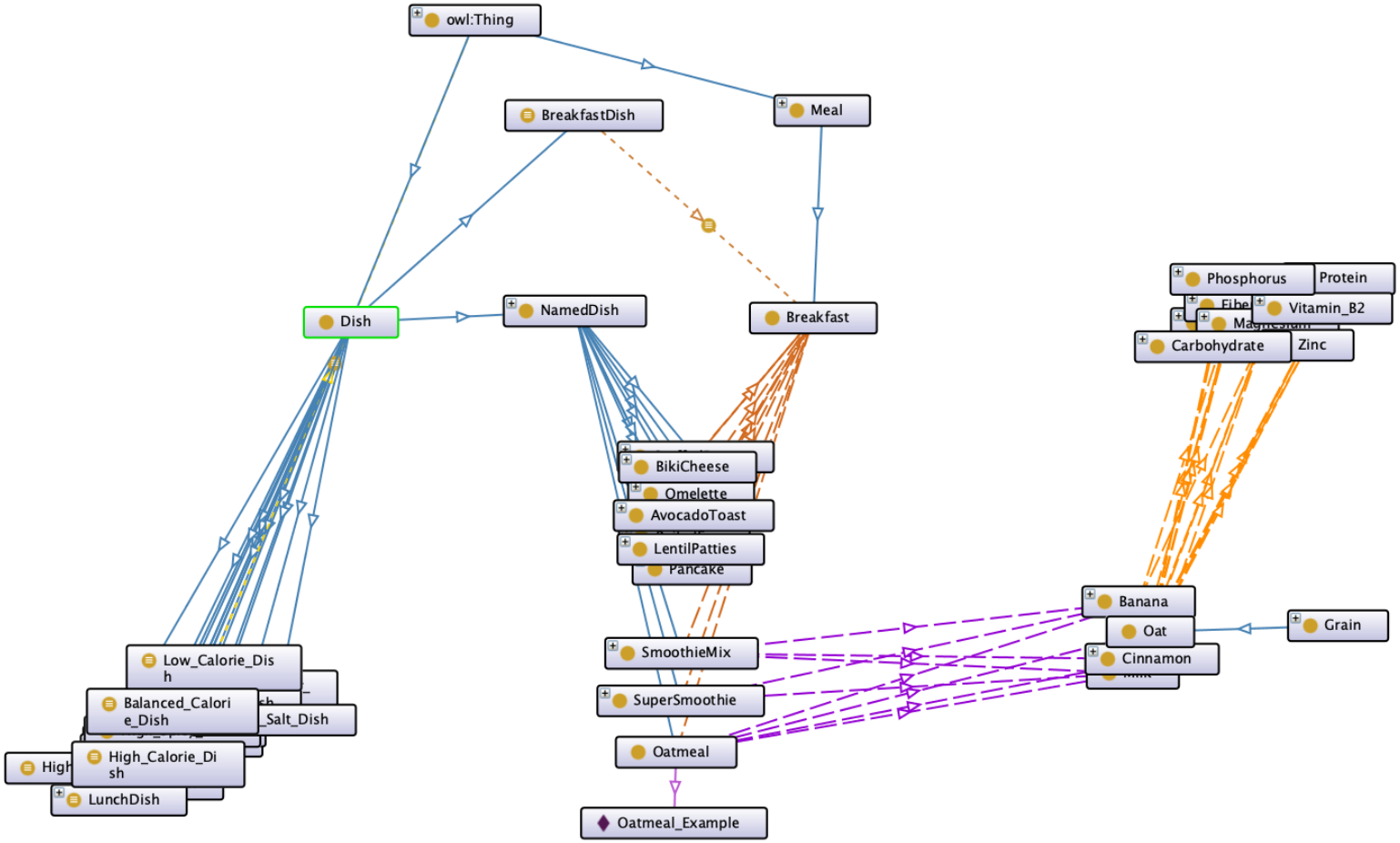


Figure 21

Representing that Phosphorous is a mineral, and the ingredients contain Phosphorous. For example, Honey has Phosphorous which is used in some Dishes and showing that Susan has allergy to Honey.

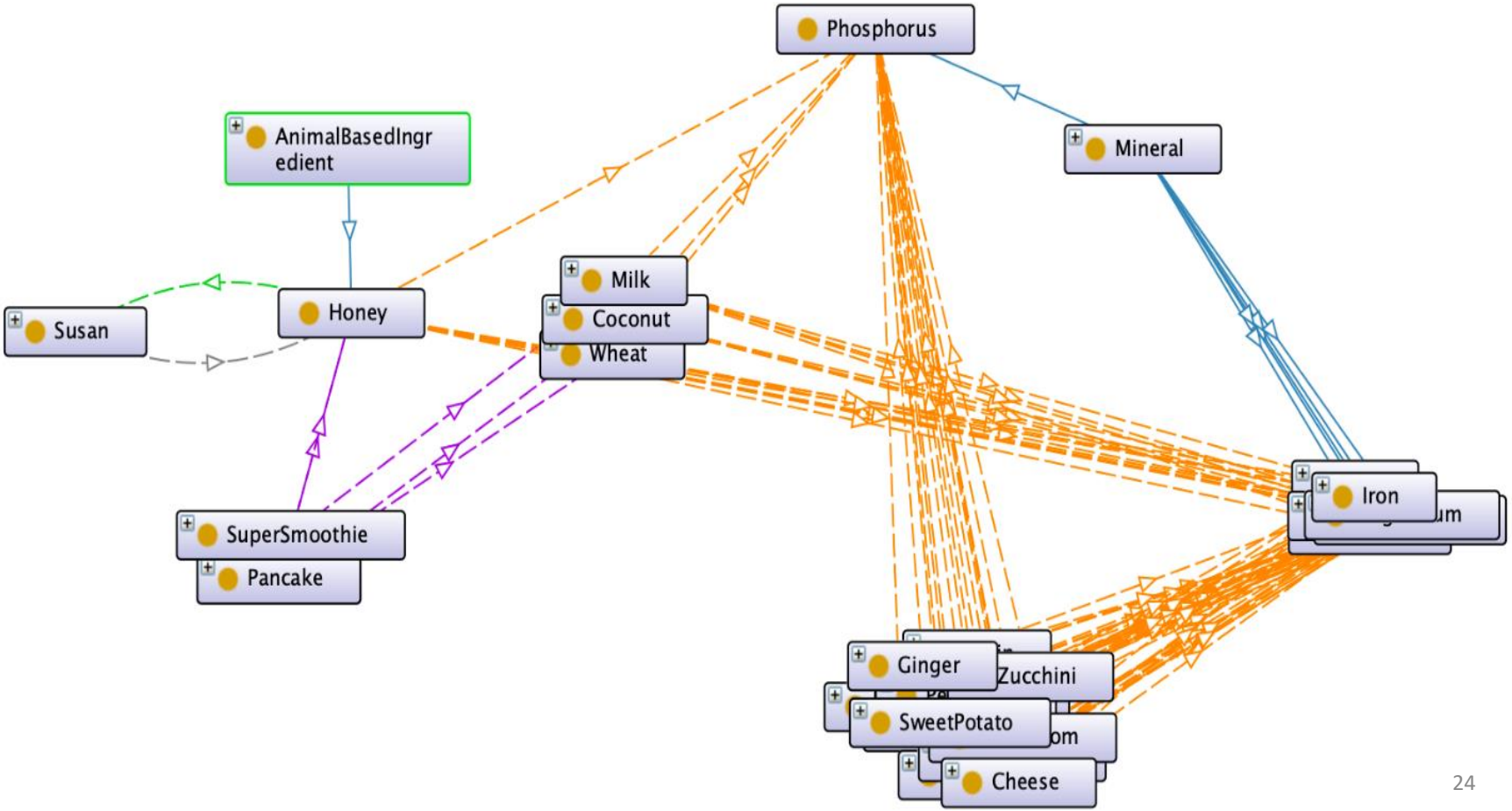


Figure 22

Representing the Level_Of_Spicy based on two different ingredients which have some nutrients in common. Moreover, representing the dishes in these two different categories (mild and hot).

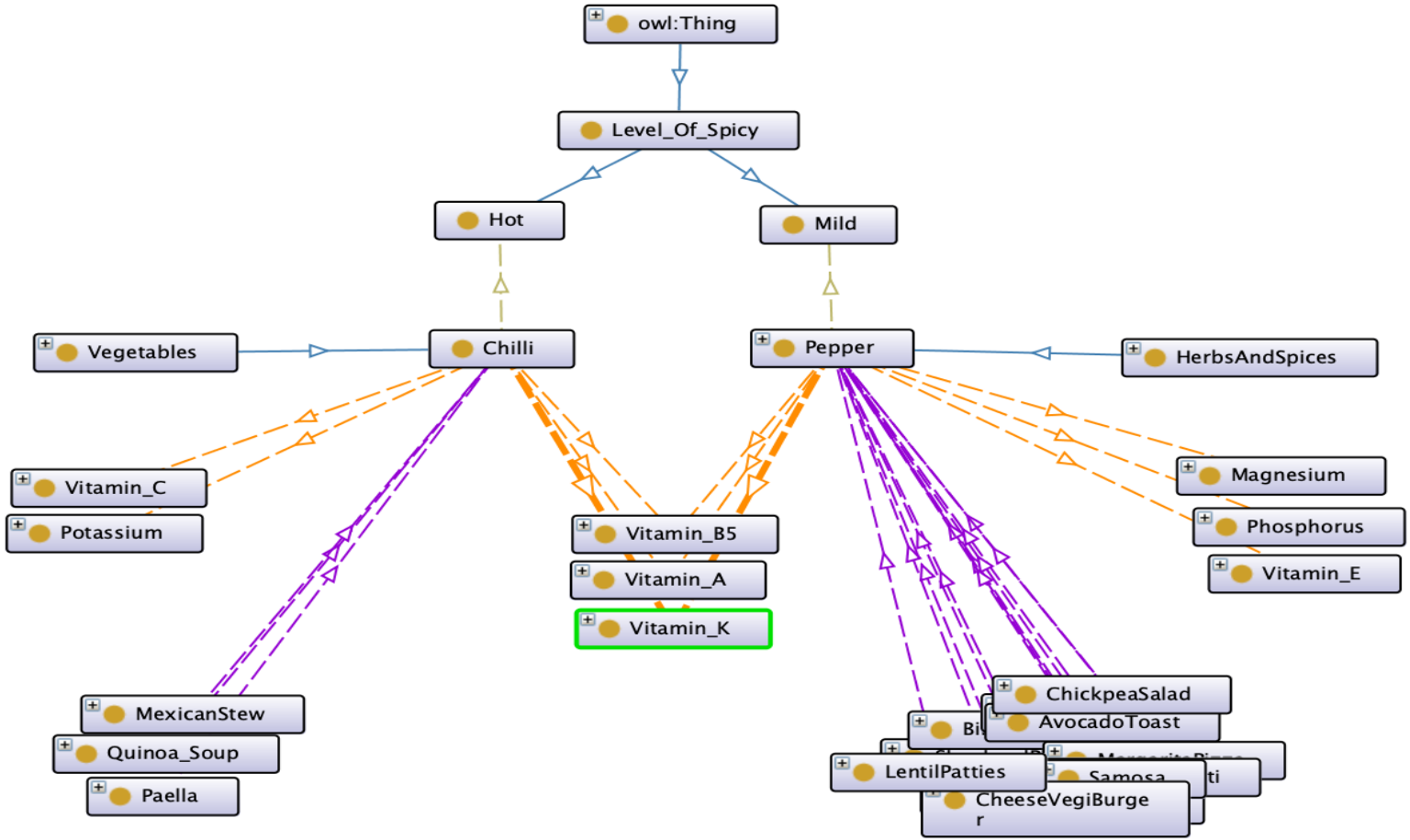


Figure 23

Query 1 and the answer

- Recommend the user dishes which contain corn or oat or potato but without egg.
- Dish that hasIngredient some (Corn or Oat or Potato) and not(hasIngredient some Egg)

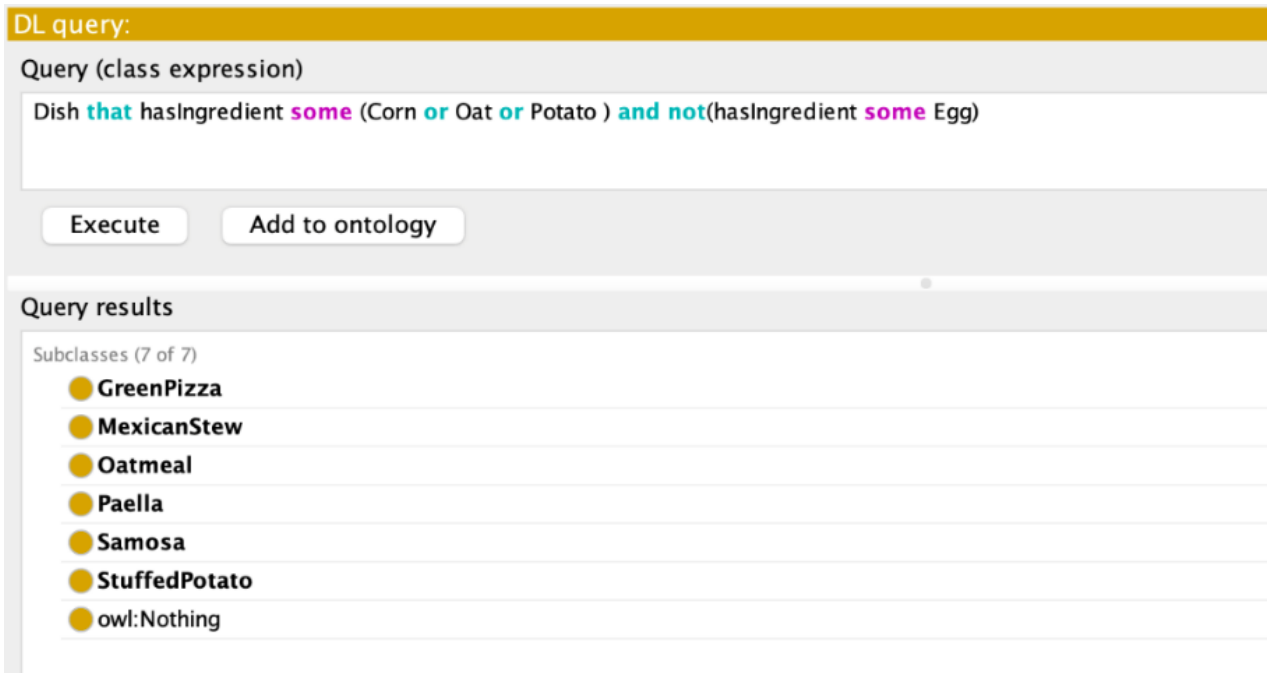


Figure 24

Query 2 and answer

- Recommend the user a high-protein dish rich in zinc or phosphorus.
- HighProteinDish and hasIngredient some (hasNutrient some(Zinc or Phosphorus))

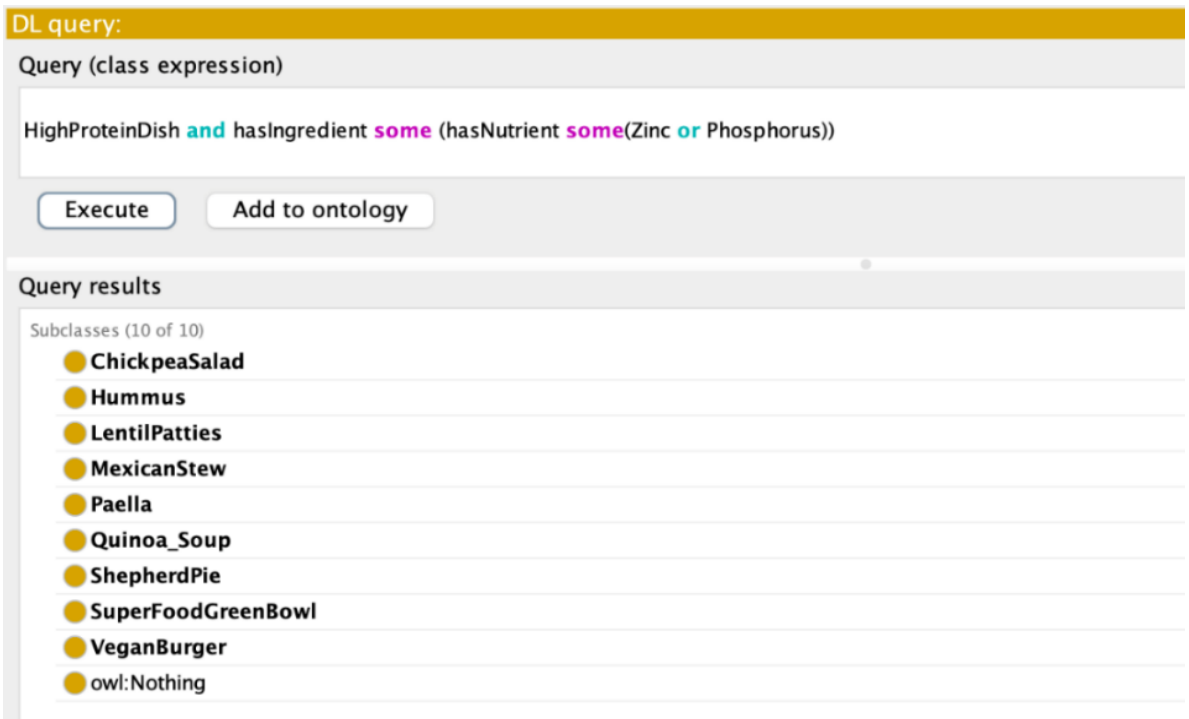


Figure 25

Query 3 and the answer

- Recommend the user dishes with the calorie value between 300 and 420, also contain kale or spinach or tomato but no peanut.

- (hasCalorieValue some xsd:integer[>=300,<420]) and (hasIngredient some(Kale or Spinach or Tomato)) and (not(hasIngredient some Peanut))

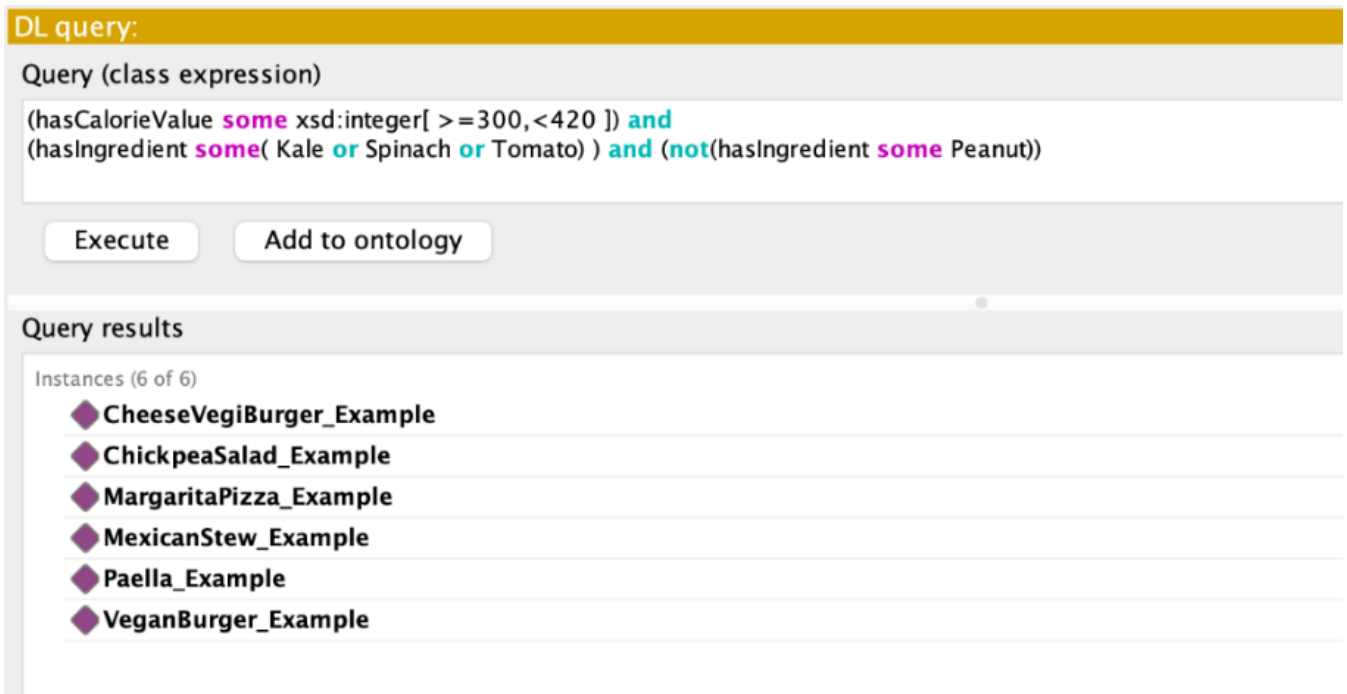


Figure 26

Query 4 and the answer

- Recommend the user a balanced-salt vegan lunch which contains olive oil or sunflower seed.
- VeganDish and Balanced_Salt_Dish and (servedAsMeal some Lunch) and (hasIngredient some (SunflowerSeed or OliveOil))

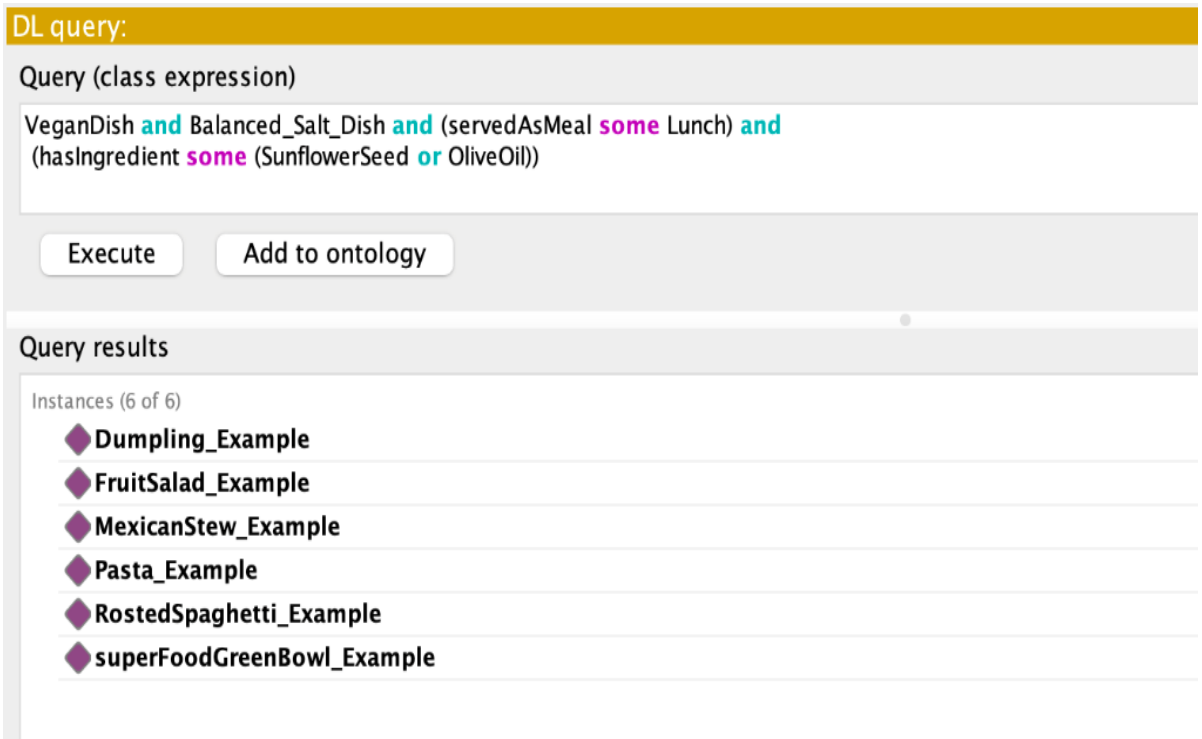


Figure 27

Query 5 and the answer

- Recommend Sara a mild-spicy dinner without the ingredients which she has allergy to them.

- Dish that not(hasIngredient some(isAllergicTo only Sara)) and Mild_Spicy_Dish and servedAsMeal some Dinner

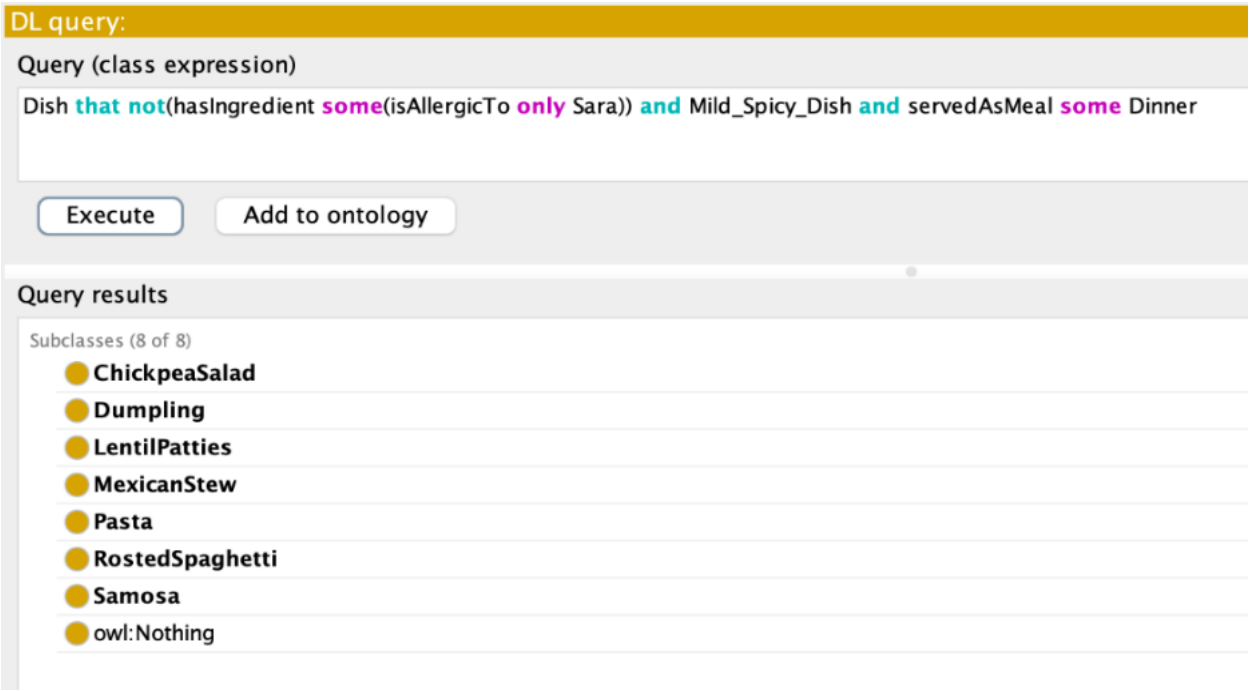


Figure 29

Query 6 and the answer

- Recommend a high protein dish to Frank which contains the necessary nutrient for his disease.
- HighProteinDish that hasIngredient some(hasNutrient some (helpWithDisease some(affectUserLife some Frank)))

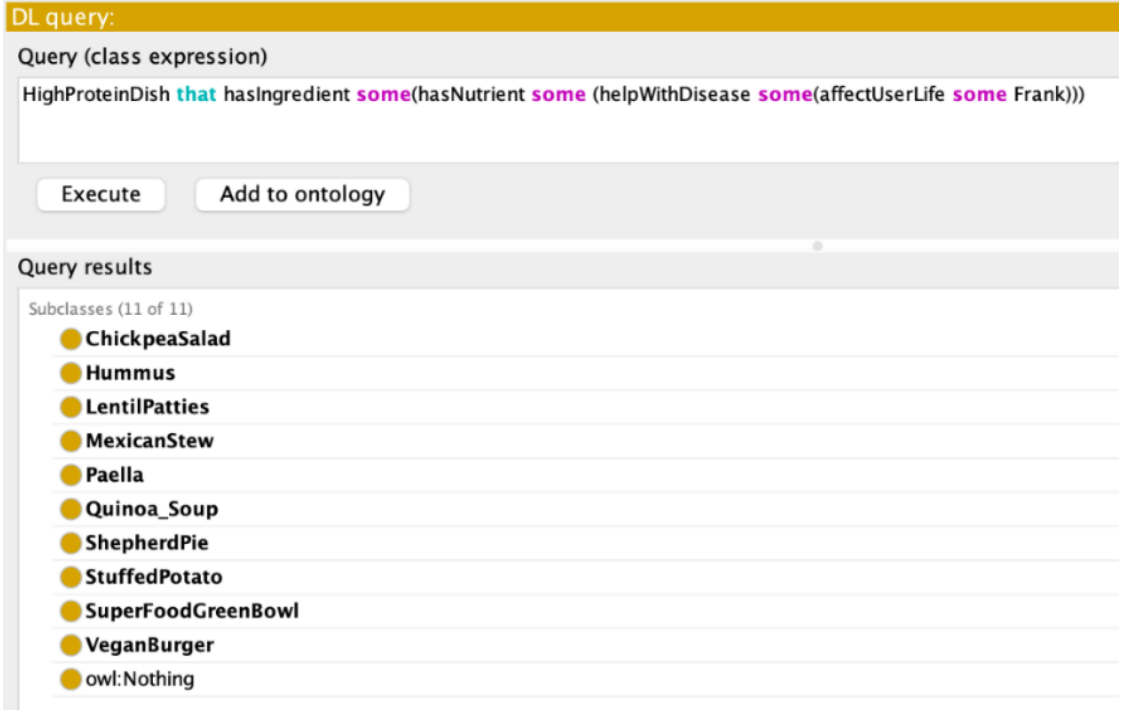


Figure 30

Query 7 and the answer

- Recommend a high protein dinner to Susan and Frank which can help them with their both diseases.

- HighProteinDish and DinnerDish and (hasIngredient some((hasNutrient some (helpWithDisease some(affectUserLife some Frank))) and (hasNutrient some(helpWithDisease some (affectUserLife some Susan)))))

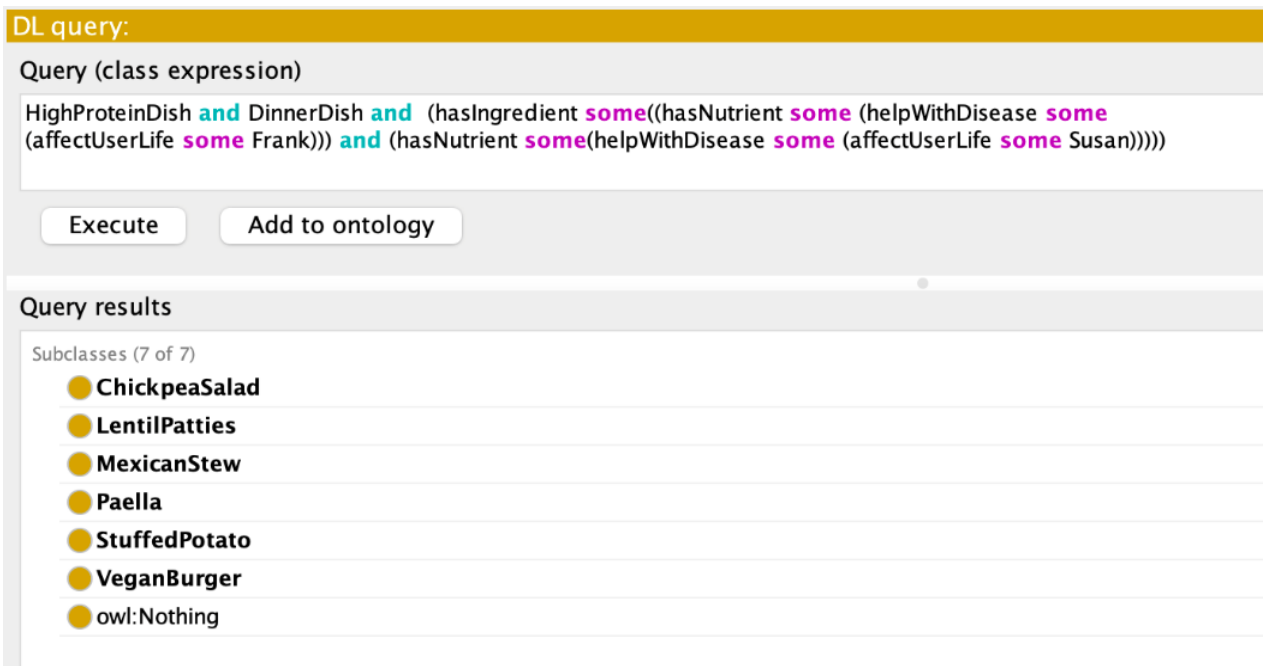


Figure 31

Conclusion

According to the current ontology, user can have their favorite food with their desired ingredients which also contains the required nutrition for their health conditions.

Moreover, this ontology can suggest different category of dishes like vegan dish, high protein dish, low salt dish.

The Food Recommendation Ontology can be applied in various domains such as restaurants, food industry and domestic uses.

Bibliography

- [1] Dooley, D., Andres-Hernandez, L., Bordea, G., Carmody, L., Cavalieri, D., Chan, L., Castellano-Escuder, P., Lachat, C., Mougin, F., Vitali, F. and Yang, C., 2021, September. Obo foundry food ontology interconnectivity. In CEUR Workshop Proceedings (Vol. 2969).
- [2] Horridge, M., Drummond, N., Goodwin, J., Rector, A.L., Stevens, R. and Wang, H., 2006, November. The Manchester OWL syntax. In OWLed (Vol. 216).
- [3] Horridge, M., Jupp, S., Moulton, G., Rector, A., Stevens, R. and Wroe, C., 2009. A practical guide to building owl ontologies using protégé 4 and co-ode tools edition1. 2. The university of Manchester, 107.
- [4] Neuhaus, F. and Brodaric, B., 2022, January. NAct: The Nutrition & Activity Ontology for Healthy Living. In Formal Ontology in Information Systems: Proceedings of the Twelfth International Conference (FOIS 2021) (Vol. 344, p. 129). IOS Press.
- [5] Stevens, R., Stevens, M., Matentzoglu, N. and Jupp, S., 2015. Manchester family history advanced OWL tutorial. Manchester: The University of Manchester.