IA301 - Logics and Symbolic AI
TP

Joaquim Minarelli Gaspar
Filipe Lacerda
Eduardo Guimarães
Yaël Gossec

Palaiseau, France

Novembre 2024

Table des matières

1 OWL with Protege

1.1 Simple Ontology (Preliminary Case)

We explored the tool using the given example. The results are presented below.

— What do we observe when we check the inferred classes ?

Answer : The inferred classes are the same as the asserted classes. This means that there are no additional logical consequences or inferences made by the reasoner beyond what has already been explicitly asserted in the ontology. This indicates that the ontology is consistent, and the asserted hierarchy or relationships between classes are correctly modeled according to the ontology's logical structure.

— Some DL queries examples and results.

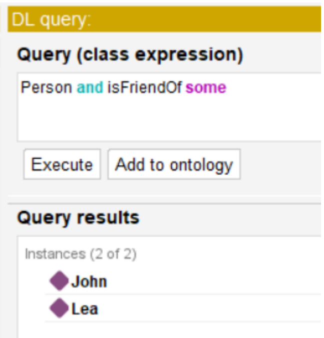


FIGURE 1 – Persons who have friends

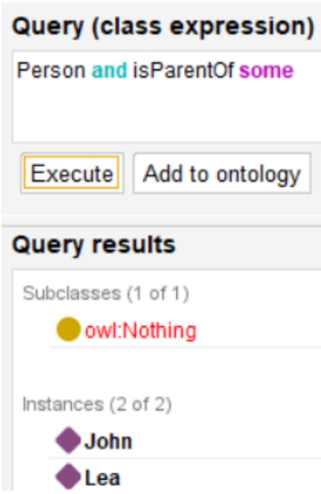


FIGURE 2 – Persons who are Parents

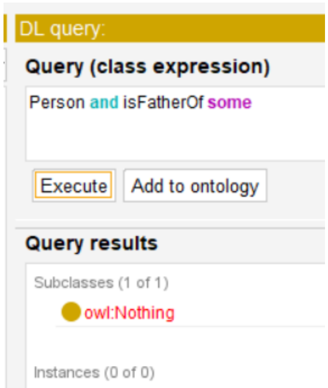


FIGURE 3 – Persons who are Fathers : Issue there

FIGURE 4 – Adding two axiom to the Father Class

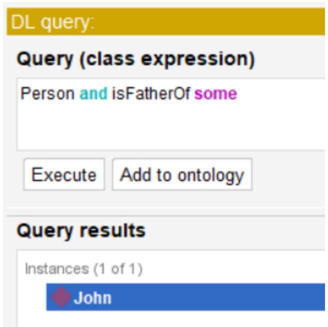


FIGURE 5 – This enabled the reasoner to infer that John is a Father

1.2 Ontology of Ecosystem Interactions and Resource Consumption

To explore the logical relationships related to the consumption of natural resources and the environment, the theme chosen for the ontology was the model of an ecosystem, with a focus on the relationships and interactions among organisms (such as animals, plants, and microorganisms), their environments, and the resources they require.

Overall, our model presents three main classes that interact with each other : the classes of **Organisms**, **Resources**, and **Habitats**. Organisms are further divided into three main subclasses : Animals, Plants, and Microorganisms. These organisms inhabit specific Environments and consume a set of natural resources provided by those environments.

Additionally, important relationships can be inferred from this model, with the main ones being :
— The relationship of coexistence, where two organisms cohabit the same environment without one being a predator of the other, or, when applied to microorganism, if they do not consume the same resource ;
— Relationships involving the consumption of certain natural resources based on the characteristics of the organisms and the environment in which they live ;
— Relationships concerning the food chain within the ecosystem, based on the type of food/energy source required by the organism along with characteristics of the environment.

The following image illustrates the relationship diagram that was implemented in the Protege software.

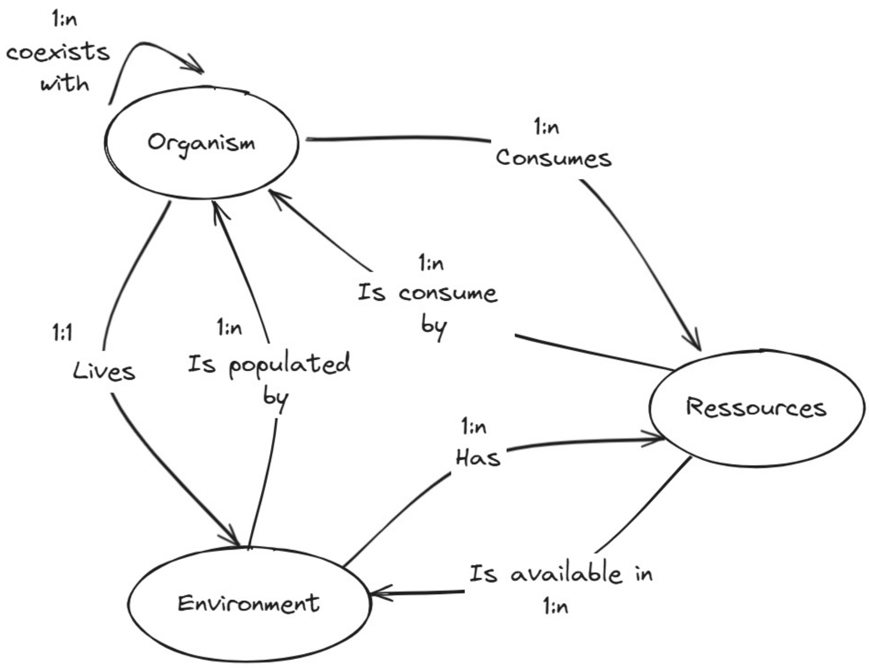


FIGURE 6 – Relationship diagram

In Figure 6, there is a cardinality relation above the arrows. These relate m individuals from the source category with n individuals from the target category. For example, the "Consumes" relation going from Organism to Resource indicates that 1 individual can consume n resources. Conversely, its symmetric relation, "Is consumed by," indicates that 1 resource can be consumed by n organisms.

2 Protege implementation

The following describes the classes and subclasses used to create the ontology in Protege.

— **Classes** (that are subclasses of owl :Thing) :

— Organism ;
— Habitat ;
— Resource.

— **Subclasses** :
  — Animal, Plant and Microorganism as subclasses of Organism (all mutually disjoint from each other) ;
  — Herbivore, Carnivore and Omnivorous as subclasses of Animals, where Omnivorous is defined as Herbivore ∩ Carnivore ;
  — Bryophytes and Angiosperms as subclasses of Plants (all mutually disjoint from each other).

It is important to note that all classes and subclasses are mutually exclusive ; therefore, no individual from one class can belong to another, and no individual from one subclass can belong to another subclass. The hierarchy of classes can be better observed in the figure below.

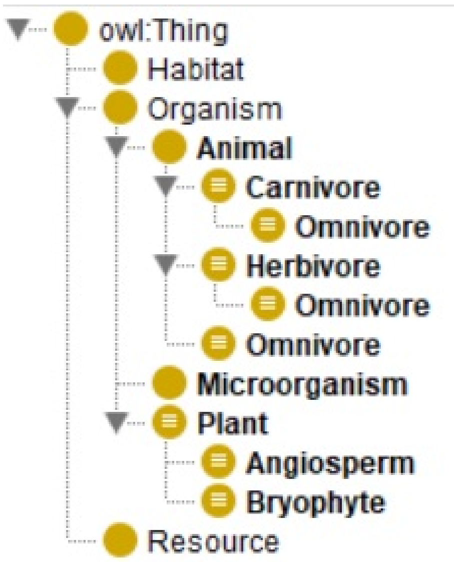


FIGURE 7 – Class hierarchy

Regarding the Class properties, we have the following **Data properties** :

— *weight* with domain Animal and range xsd :decimal ;
— *has_ Fruits* with domain Plant and range xsd :boolean. In addition, this property is set to be True for Angiosperms and False for Briophites ;
— *chemical_ formula* with domain Ressource and range xsd :string ;
— *scientific_ name* with domain Organism and range xsd :string.

The following two figures illustrate both the declaration of data properties and the method used to ensure the uniqueness of the *scientific_ name*, for example.

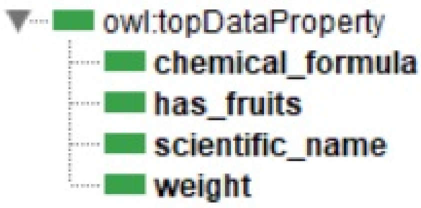


FIGURE 8 – Data properties

FIGURE 9 – Unique property

Finally, we have the following **Object properties** :

— *eat* with domain Animal and range Organism. This property is also irreflexive ;
— *foundInHabitat* with domain Organism and range Habitat ;
— *consumeTheResource* with domain Organism and range Resource ;
— *isEatenBy* with domain Organism and range Animal (is the inverse property *eat*) ;
— *isNotEatenBy* with domain Animal ∪ Plant and range Animal. It's disjoint with *isEatenBy* ;
— *isOccupiedBy* with domain Habitat and range Organism (is the inverse of *foundInHabitat*) ;
— *hasTheResource* with domain Habitat and range Resource ;
— textitisConsumedBy with domain Resource and range Organism (is the inverse of *consumeTheResource*) ;
— *isInTheHabitat* with domain Resource and range Habitat (inverse of the property *hasTheResource*)
— *coExistWith* with domain Organism and range Organism. This property is symmetric. In addition, this property is different if both organisms are animals/plants or if they are both microorganisms :

1. A and B are herbivores/plants : A coexists with B means that A and B lives at the same environment (foundInHabitat(A) == foundInHabitat(B)) ;
2. A and B are both microorganisms : A coexists with B means that A and B lives at the same environment (foundInHabitat(A) == foundInHabitat(B)) **and** they do not consume the same resource through the relation consumeTheResource ;
3. A and B are carnivores : A coexists with B means that A and B lives at the same environment (foundInHabitat(A) == foundInHabitat(B)) **and** they do not share neither the relation A *isNotEatenBy* B and B *isNotEatenBy* A ;

Finally, if two Organisms coexists, it means that they should be both at the same Habitat :

$$foundInTheHabitat(A, Hab) \wedge coExistWith(A, B) \rightarrow foundInTheHabitat(B, Hab)$$

The list of object properties in Protege can be seen in the figure below.

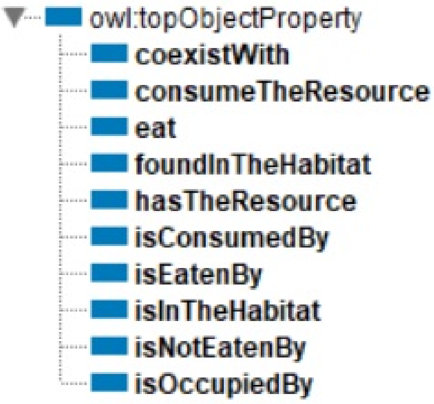


FIGURE 10 – Object properties

The inference rules were described using *SWRL* (Semantic Web Rule Language), which allows for writing it in a practical and concise manner. Below, we provide a detailed description of all the inference rules related to object and data properties.

We start by the following inference rule : if Organism a eat Organism b, and Organism a live in the Habit env, then b should also live in env. This is described by :

```
eat(?a, ?b) ^ foundInHabitat(?a, ?env) -> foundInHabitat(?b, ?env)
```

Regarding the consumption of resources, we have the following equivalence : if Organism A consume Resource R and A is in the Habitat H, then R can be found in H. In addition, if Resource R can be found in the Habitat H and Organism A consumes this resource, then A lives in H. This can be described using SWRL by the following 2 statements :

```
consumeTheResource(?a, ?r), foundInTheHabitat(?a, ?h) -> isInTheHabitat(?r, ?h)
```

```
consumeTheResource(?a, ?r), isInTheHabitat(?r, ?h) -> foundInTheHabitat(?a, ?h)
```

The same way, if Organism a coexists with Organism b, and a live in the Habitat env, b should also live in env.

```
coexistWith(?a, ?b) ^ foundInHabitat(?a, ?env) -> foundInHabitat(?b, ?env)
```

From now on, we will focus on discussing the coexistence relationship, which is the most complex relation of the ontology. As previously described, according to our rules, a carnivorous animal will always coexist with another carnivorous animal as long as they inhabit the same environment and they share the relation isNotEatenBy. Additionally, we add the rule that animal a must be different from b, as we do not want the relation to be reflexive :

```
Carnivore(?a), Carnivore(?b), DifferentFrom (?a, ?b), foundInTheHabitat(?a, ?h)
    , foundInTheHabitat(?b, ?h), isNotEatenBy(?a, ?b), isNotEatenBy(?b, ?a) ->
    coexistWith(?a, ?b)
```

We use the same logic to describe the coexistence for the herbivores, as follows :

```
Herbivore(?a), Herbivore(?b), DifferentFrom (?a, ?b), foundInTheHabitat(?a, ?h)
    , foundInTheHabitat(?b, ?h) -> coexistWith(?a, ?b)
```

Next, we have the case of coexistence between plants, which is similar to the herbivore's case :

```
Plant(?a), Plant(?b), DifferentFrom (?a, ?b), foundInTheHabitat(?a, ?h),
    foundInTheHabitat(?b, ?h) -> coexistWith(?a, ?b)
```

Finally, the coexistence between two microorganism happens when they both live in the same environment **and** do not consume the same resource :

```
Microorganism(?a), Microorganism(?b), DifferentFrom (?a, ?b), foundInTheHabitat
    (?a, ?h), foundInTheHabitat(?b, ?h), consumeTheResource(?a, ?r),
    consumeTheResource(?b, ?s), DifferentFrom (?r, ?s) -> coexistWith(?a, ?b)
```

2.1 Inferences and Queries

We now create some individuals with the classes and object properties stated in the section above to analyze the reasoner's ability to infer them. We give only a limited set of details to each individual, in the hopes that the reasoner could fill in the holes and show us intuitive relations we thought beforehand.

All the examples created for Plants, Microorganisms, Animals, and Resources are in Tables 1,2, 3, and 4. We highlight the fact that some properties are clearly missing, like for Oxygen and Water, not clearly stated to be in Savanna. We also mention that the individual Habitat, Savanna, was not programmed with any properties, reason why it's not included in one of the tables.

TABLE 1 – Individuals for plants

Name	Attributes	Properties
Acacia	Has fruits Acacia dealbata	Consumes water Found in Savanna Consumes CO2 Consumes O2
Apple	Has fruits Malus domestica	Consumes water
Grass		Consumes CO2 Consumes water

TABLE 2 – Individuals for microorganisms

Name	Attributes	Properties
Bacteria	Escherichia coli	Consumes CO2
NitroBacteria	Nitrobacter alkalicus	Consumes N2

Table 3 – Individuals for resources

Name	Attributes	Properties
Carbon Dioxide	CO_2	Is in Savanna
Nitrogen	N_2	Is in Savanna
Oxygen	O_2	
Water	H_2O	

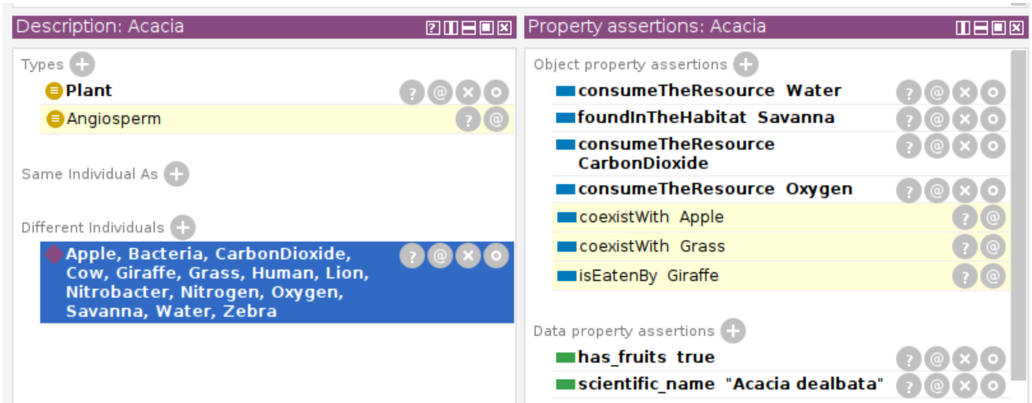
Table 4 – Individuals for animals

Name	Attributes	Properties
Cow	Weight Bos taurus	Eats grass Consumes water Consumes O2
Giraffe	Weight Giraffa camelopardalis	Eats Acacia Consumes water Consumes O2
Human	Weight	Eats Apple Eats Cow Not eaten by Lion Consumes water Consumes O2
Lion	Weight Panthera leo	Eats Giraffe Eats Zebra Found in Savanna Consumes water Not eaten by Human
Zebra	Weight Equus quagga	Eat Apple Consumes O2 Consumes water Eats Grass

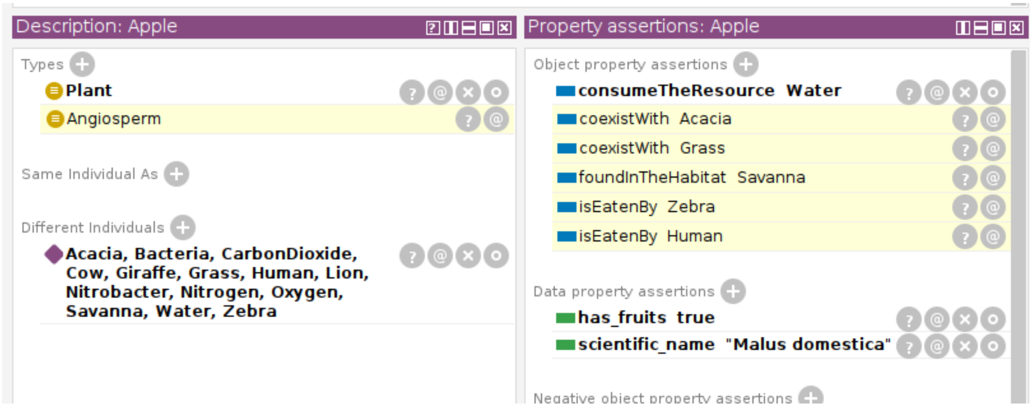
We now run the reasoner, HermiT, and present some snapshots of the inferences found by it.

2.1.1 Plant Inferences

We begin with the first two plants in our individual's collection : Acacia and Apple.



(a) Acacia's description on Protégé.



(b) Apple's description on Protégé.

FIGURE 11 – Inference showcase for both Acacia and Apple individuals of the Plant subclass.

As we can see by figure 11, the reasoner is capable of inferring that both plants are Angiosperms, just by the presence of the fruit in their attributes. It's also able to infer their coexistence, between themselves and with Grass, Acacia's property of being eaten by Giraffe, Apple's localization in Savanna and the fact its consumed by both Humans and Zebras.

In figure 12 we see Grass' inferences for its coexistence with Apple and Apacia, its localization in Savanna, and the fact it's eaten by Zebras and Cows.

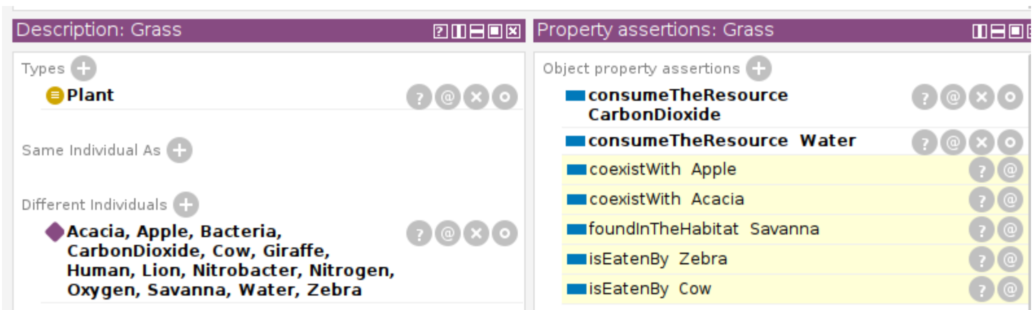
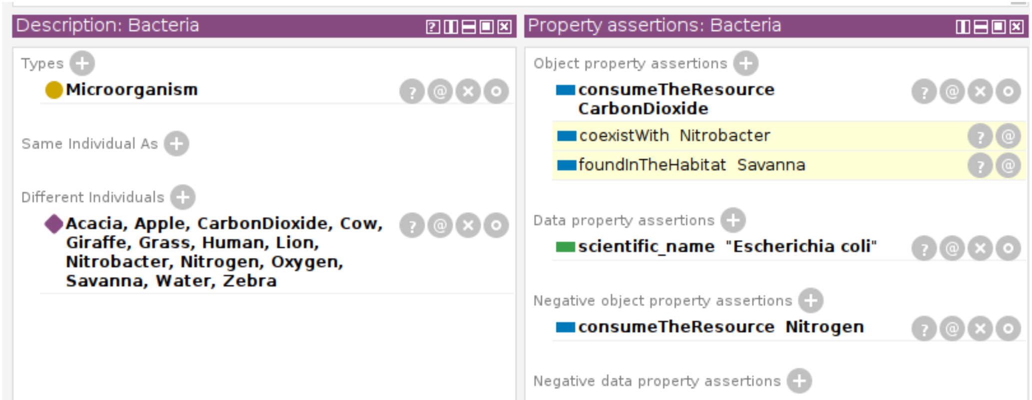


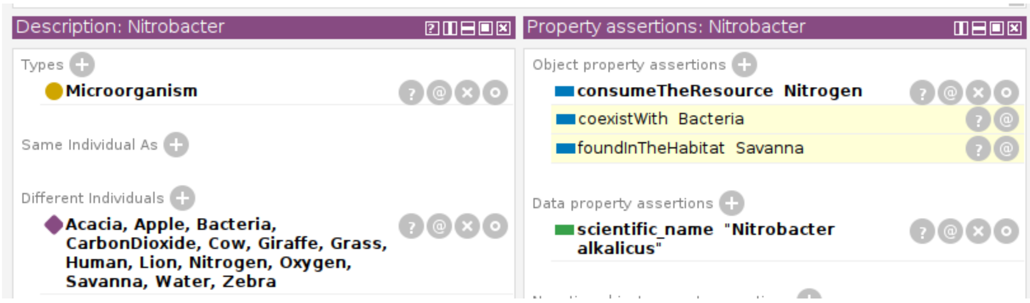
FIGURE 12 – Grass' description on Protégé.

2.2 Microorganism Inferences

For the microorganisms, we check figure 13. Both of their localization was inferred to be Savannas, and also the fact they coexist, that could be deduced by the fact both consume different resources : Bacteria consumes CO_2, and NitroBacteria, N_2.



(a) Bacteria's description on Protégé.

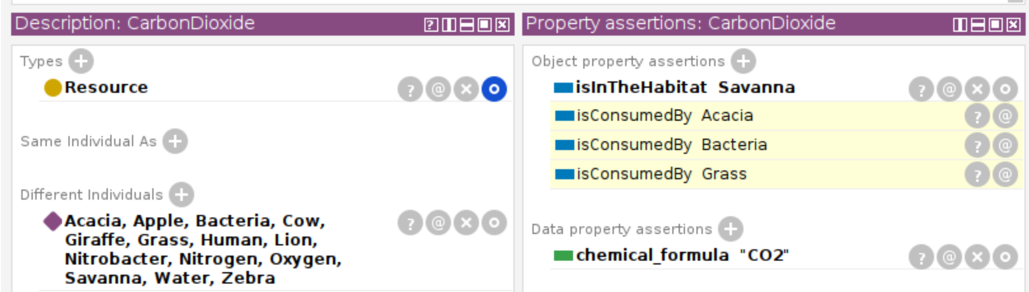


(b) NitroBacteria's description on Protégé.

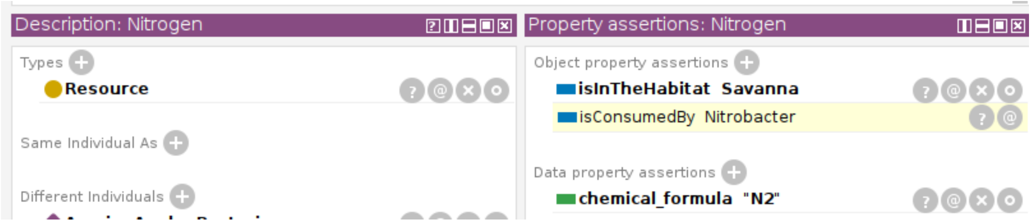
FIGURE 13 – Inference showcase for both Bacteria and NitroBacteria individuals of the Microorganism subclass.

2.3 Resources inferences

In figure 14, we see the inferences for Carbon Dioxide(CO_2) and Nitrogen(N_2), which most involves they're being consumed by the right individuals, without explicit mention of those in their properties. For Oxygen(O_2) and water(H_2O), similar inferences are showcased in figure 15, but now involve a way bigger number of *isConsumedBy* properties, and also the fact they're in Savanna.

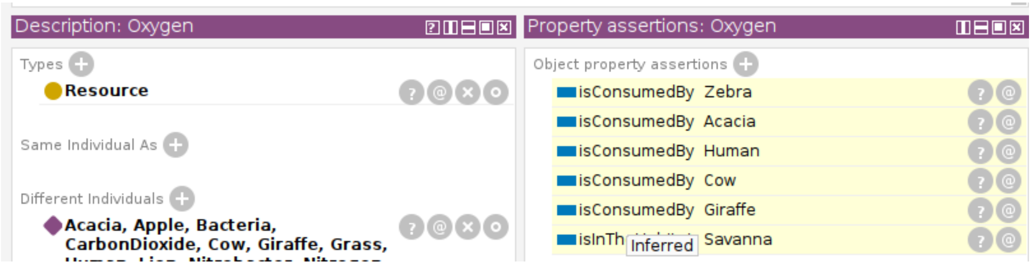


(a) Carbon Dioxide's description on Protégé.

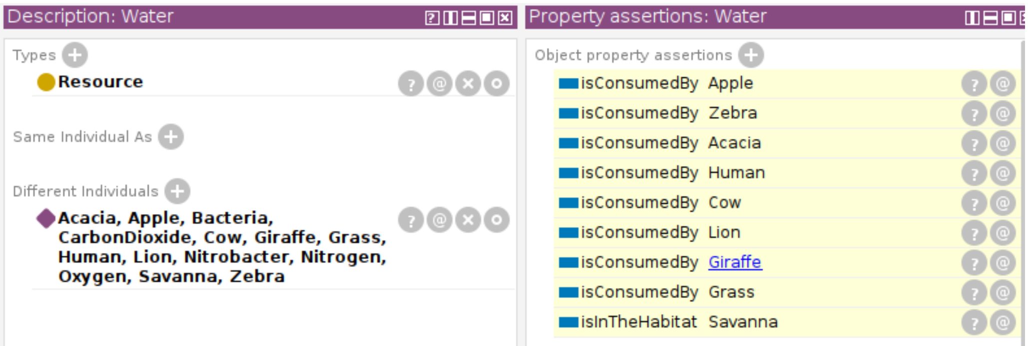


(b) Nitrogen's description on Protégé.

FIGURE 14 – Inference showcase for both Carbon Dioxide and Nitrogen individuals of the Resource subclass.



(a) Oxygen's description on Protégé.

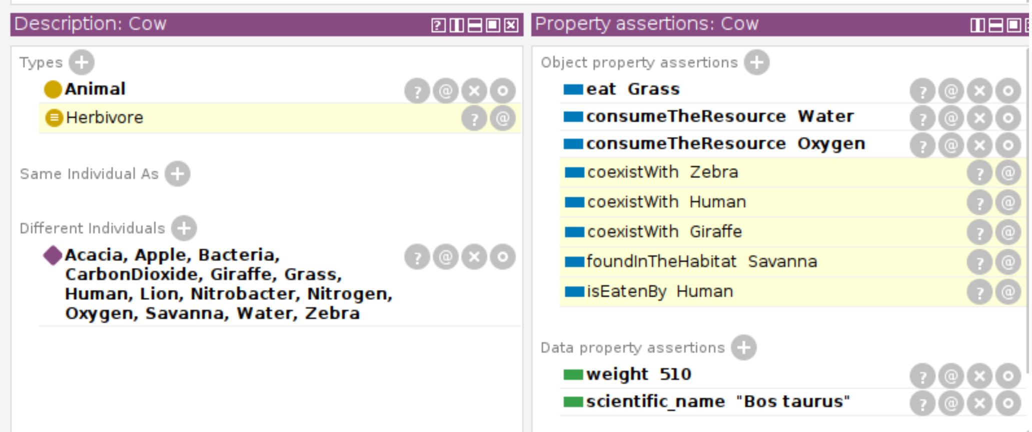


(b) Water's description on Protégé.

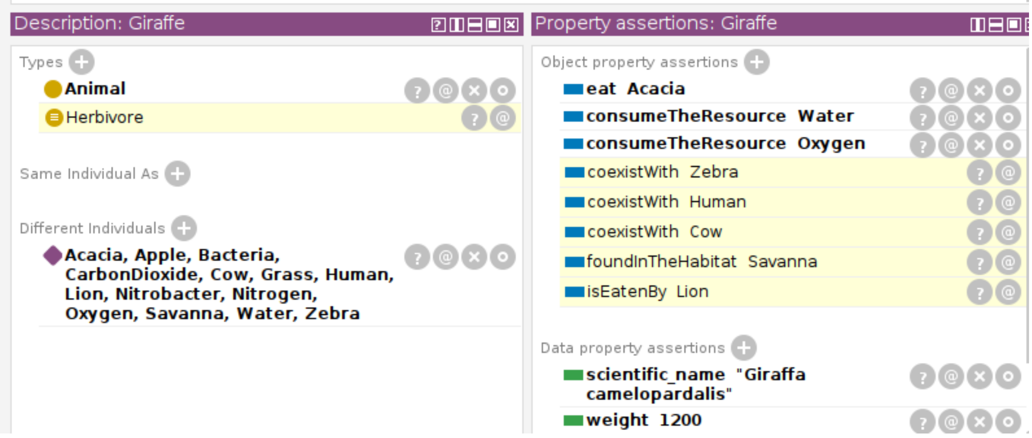
FIGURE 15 – Inference showcase for both Oxygen and Water of the Resource subclass.

2.4 Animals inferences

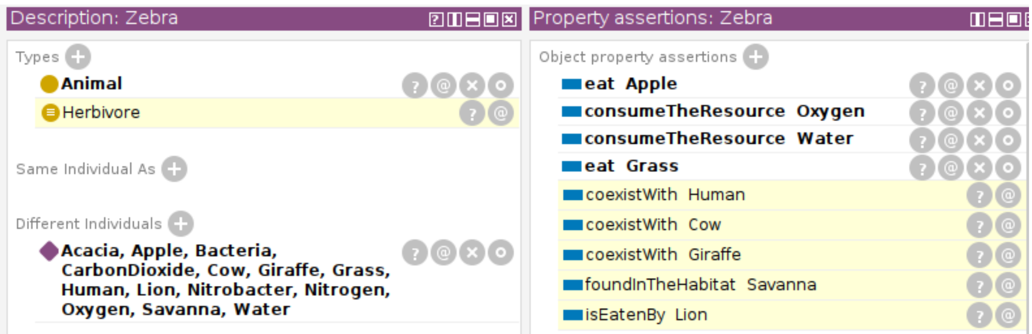
We begin by the showcase of our Herbivores inferences, in figure 16. Aside from the correct coexistence between each other and their predators, the reasoner is also capable of classifying each one of them as *Herbivore*, from the fact they only eat Plants.



(a) Cow's description on Protégé.



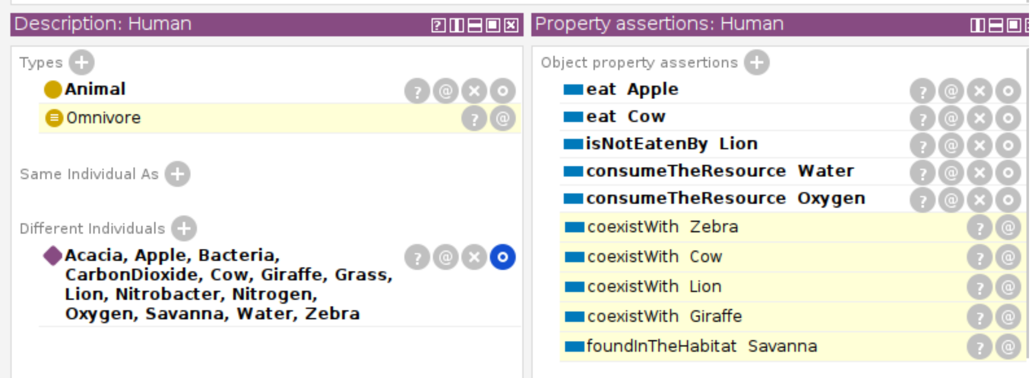
(b) Giraffe's description on Protégé.



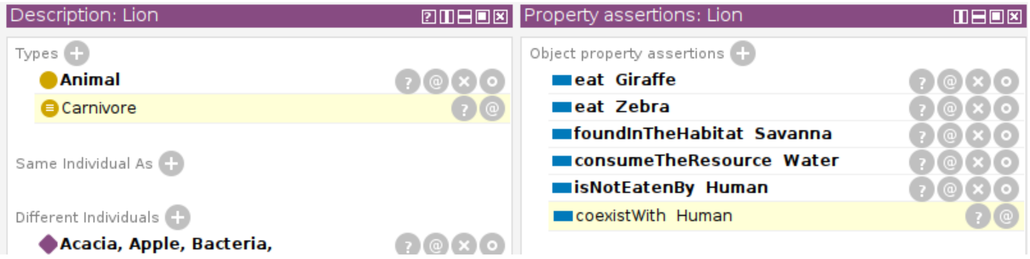
(c) Zebra's description on Protégé.

FIGURE 16 – Inference showcase for Cow, Giraffe, and Zebra individuals of the Animal subclass.

Human's and Lion's inferences can be found in figure 17. First we highlight the inference that Lion is a Carnivore from the fact he only eats meat. Second, Human's classification as a Omnivore for the diversity in the resources and Animals it consumes.



(a) Human's description on Protégé.



(b) Lion's description on Protégé.

FIGURE 17 – Inference showcase for Human and Lion individuals of the Animal subclass.

2.5 DL Queries

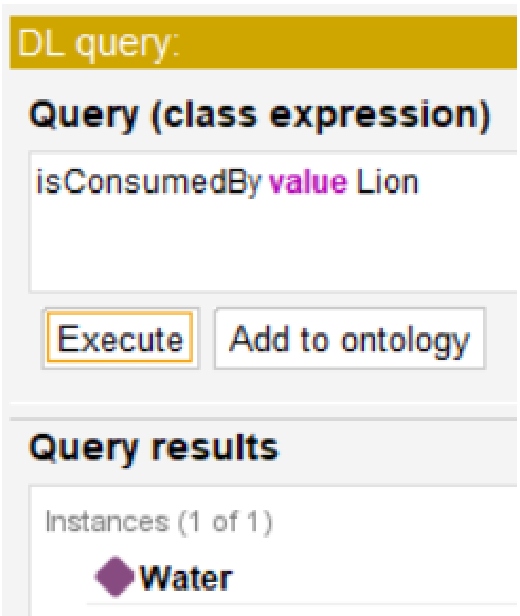


FIGURE 18 – Query 1

Explanation : This query is a simple example of inference, it translates to "What does the Lion consume ?". In the World generated, the only answer is Water.

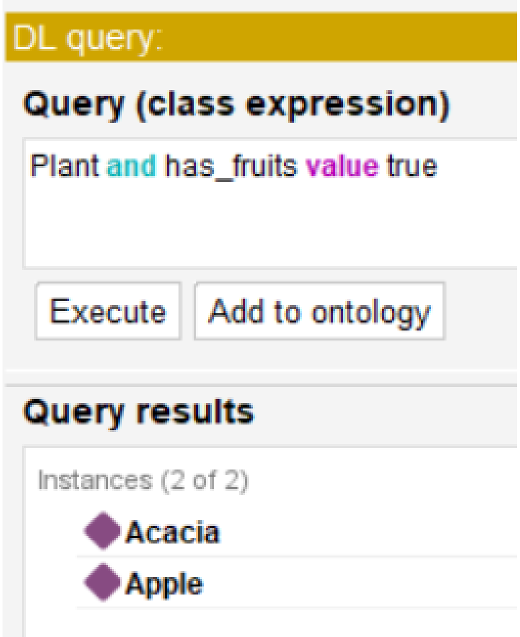


FIGURE 19 – Query 2 - Data Property

Explanation : Pretty straight forward Query once again, making use of the "has_fruits" Data Property which applies to plant with a boolean range.

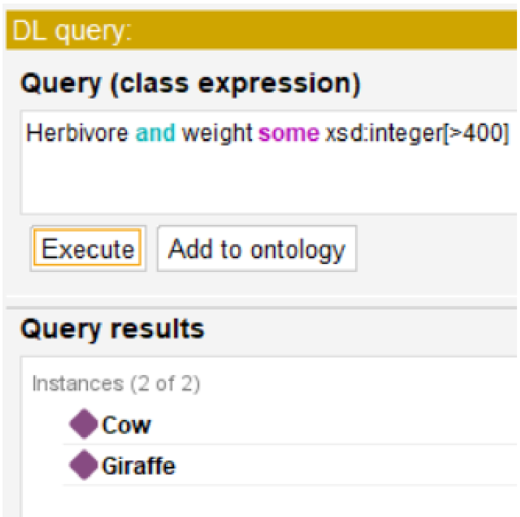


FIGURE 20 – Query 3 - Data property

Explanation : Here we filter on another Data Property "weight", we combine this filter with a filter on the Herbivore Class.

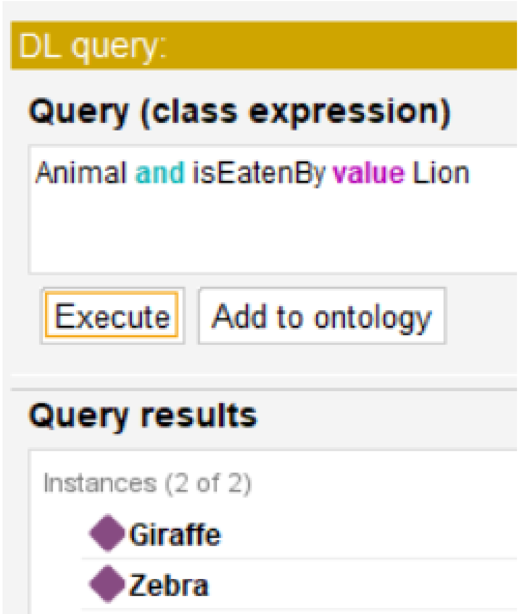


FIGURE 21 – Query 4 : Is Eaten By Property

Explanation : Simple inference of the ‘isEatenBy‘ property.

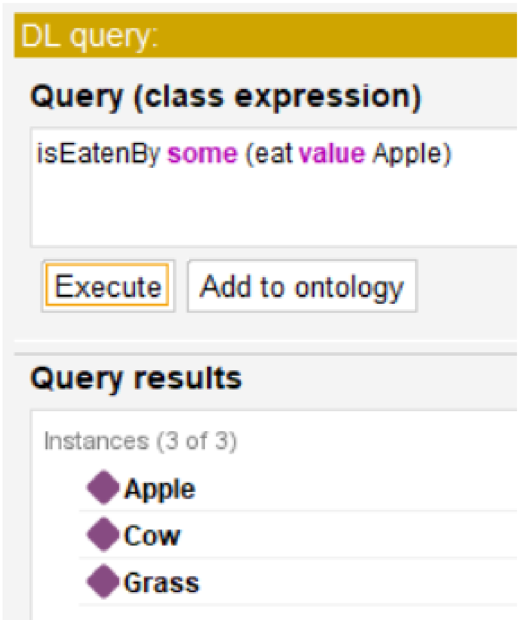


FIGURE 22 – Query 5 : Eaten by Entities Eating Specific Fruits

Explanation : This query returns the individuals eaten by another individuals that eat a specific type of fruit (here apple). It tests the ontology's ability to handle nested relational logic.
It returns the trivial answer "Apple" (it is obvious that "Apple" is eaten by individuals that eat "Apple"), but also Cow (Cow is eaten by Human, and human eat Apple) and Grass (Grass is eaten by Zebra and Zebra eat apple)

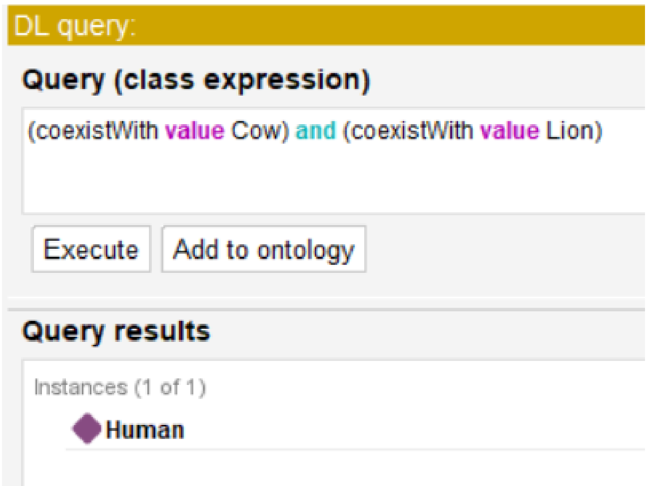


FIGURE 23 – Query 6 : Double Coexistence Query

Explanation : This query return individuals that can coexist with both Cow and Lion. It tests the ontology's ability to manage multi-entity relationships.

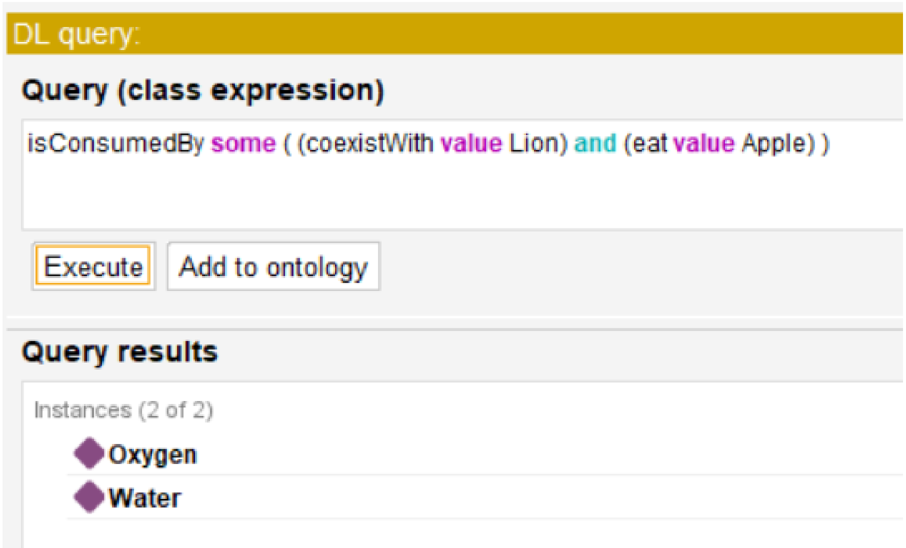


FIGURE 24 – Query 7 Part 1 : Complex Query

Explanation : This query finds resources consumed by individuals that "coexist with Lion" and "eat Apple." It shows how the ontology can handle relationships involving coexistence and specific eating behaviors.

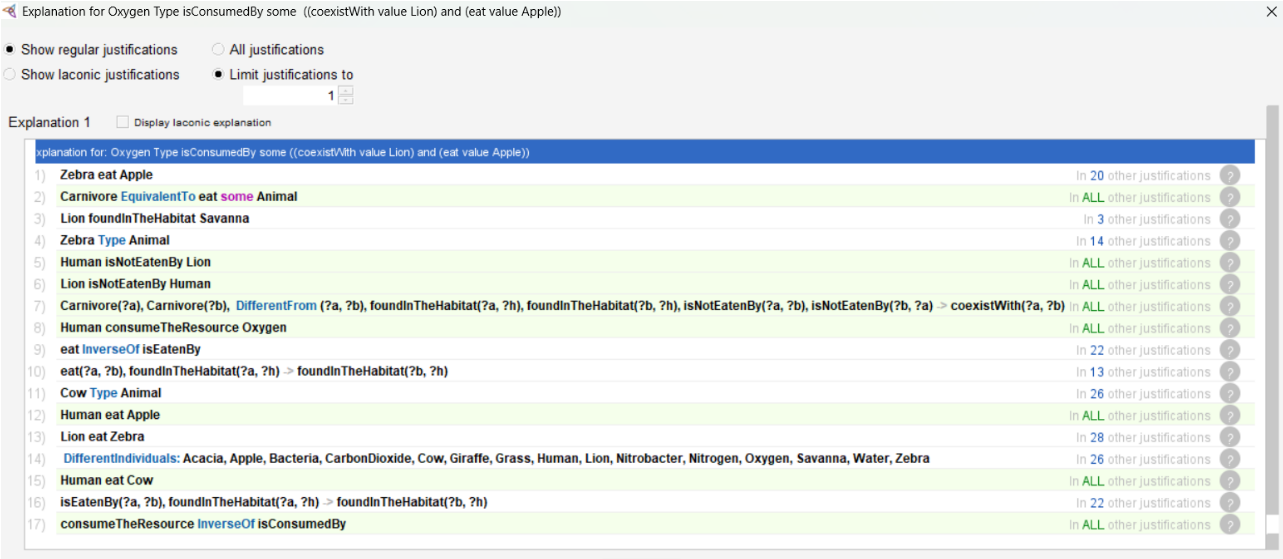


FIGURE 25 – Query 7 Part 1 : Complex Query Explanation

Explanation : Above is the detail of the inference of "Oxygen" as an output of the previous query. Lines 5) through 7) infer that Lion and Human coexists. Line 8) infers that Human consumes "Oxygen", and line 12) proves that Human eats "apple", hence it meets all the requirements to be outputted as a query result.

Références

[1] Franz Baader. « The Description Logic Handbook : Theory, Implementation, and Applications ». In : *Cambridge University Press google schola* 2 (2003), p. 7-26.

[2] The CO-ODE Project Stanford Center for Biomedical Informatics Research. *Protégé OWL Software*. `https://protege.stanford.edu`. 2012.

[3] United Nations. *Transforming Our World : The 2030 Agenda for Sustainable Development*. `https://doi.org/10.1007/s13398-014-0173-7.2`. 2015.