Introduction to Ontology Semantics and Reasoning

Henriette Harmse

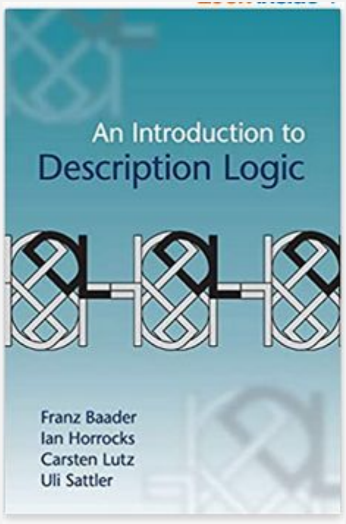
Ontology tools project lead

EMBL-EBI

Feb 2023

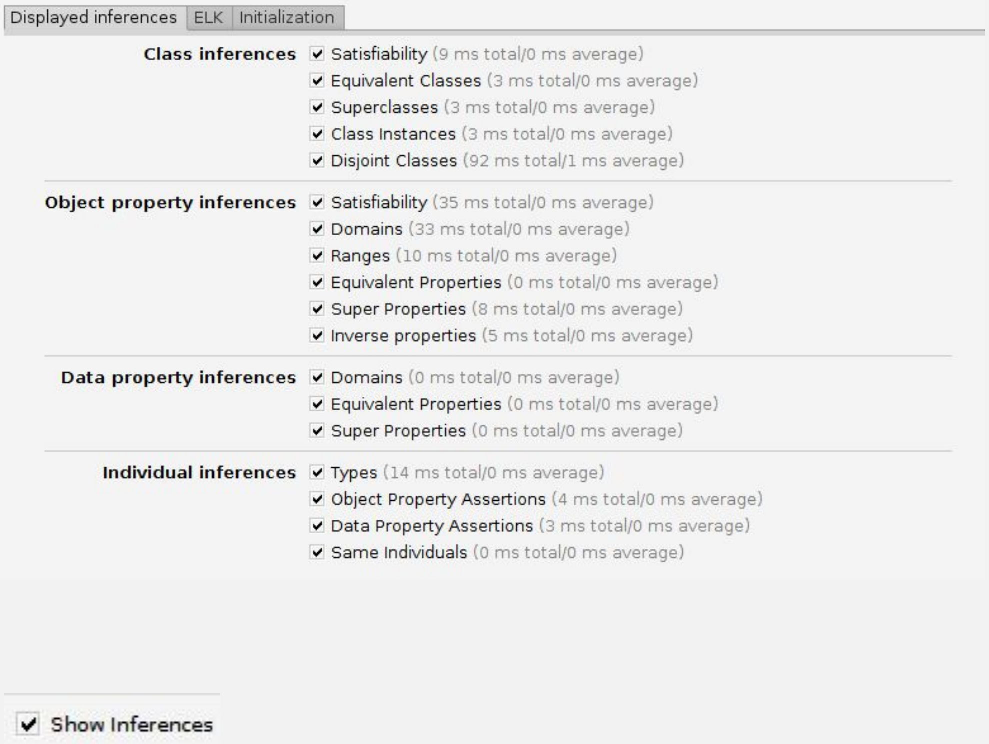
Outline

1. Preliminaries
2. Expressiveness, computational complexity and decidability
3. Why description logics?
   a. Propositional logic, Predicate logic, Description logics (DLs)
4. DL vs OWL terminology
5. What can you say in a DL?
   a. subclass, equivalence, subproperty, instance of and relations between individuals.
6. DL constructors and axioms
   a. Constructors: Negation, Qualified existential restriction, Qualified universal restrictions, Conjunction and Disjunction.
   b. Axioms: Disjointness, domain- and range restrictions



Preliminaries

1. **Objective:** To facilitate intuitive understanding of DLs rather than to represent formal math or recommendations regarding ontology design.
2. Only a small subset of most used DL constructors are dealt with.
3. To follow along in Protege, ensure you have
   a. all inferences enabled,
   b. that “Show Inferences” at ticked at the bottom of the screen, and
   c. ensure you have HermiT installed as this is reasoner used throughout.



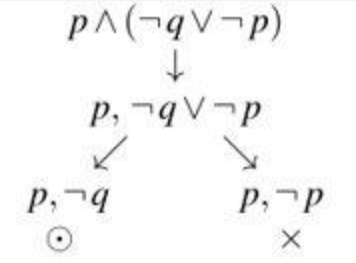
Forces driving DL research

1. **Expressivity of a logic:** How much or little you can say using a specific logic
2. **Computational complexity of an algorithm:** How much space or time an algorithm will take to complete.
3. **Decidability:** An algorithm is **decidable** for a decision problem if it always will be able to give you an answer. It is **undecidable** if it can sometimes give you an answer and sometimes not.

Why Description Logics
Propositional Logic

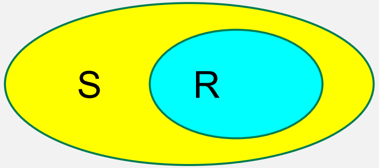
1. Example sentence: $a \wedge b$
    a. Interpretation 1: *a*=Zuko, *b*=bark => true
    b. Interpretation 2: *a*=Margo, *b*=bark => false
2. A sentence is **satisfiable** if 1 true interpretation exists.
3. Advantages:
    a. **Decidable** algorithms exist to **check satisfiability**
        i. E.g. semantic tableau
        ii. Semantic tableau has **Polynomial time complexity** for some family of sentences
4. Disadvantage:
    a. **Not very expressive**
    b. Geared towards making observations about specific individuals rather than general observations about population

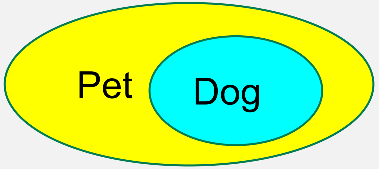
Semantic tableau

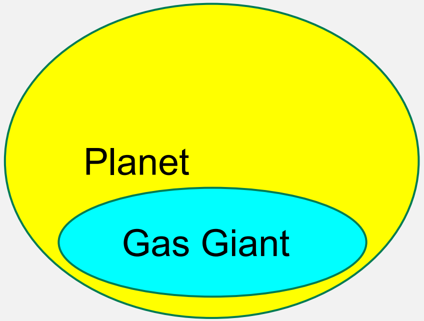


Why Description Logics
Predicate Logic

1. Definition: For S, R sets s.t. $R \subseteq S$, the predicate P is defined as $P(x) = \begin{cases} \text{true, if } x \in R \\ \text{false, if } x \notin R \end{cases}$
2. Various interpretations can exist, e.g.:
   S=Pet, *R*=Dog, *P* = pets that are dogs,
   S=Planet, *R*=Gas Giant, *P* = planets that are gas giants.
3. *P* is **satisfiable** if 1 interpretation exists in which *R* is not empty
4. Example sentence: $\forall x \forall y P(x,y) \rightarrow P(y,x)$
   a. Satisfying interpretations of *P* are *marriedTo* and *siblingOf*
5. Advantages: **Very expressive**.
6. Disadvantages: **Satisfiability algorithms are undecidable**.







Why Description Logics

Description Logics

1. **Description logics** is a group of logics where each logic in the group is a **subset of predicate logic** for which a **decidable satisfiability algorithm exists**.
2. DLs vary wrt their expressivity and time complexity.
3. SROIQ(D), which forms the basis of OWL 2 - (Reasoner: HermiT):
   a. Very expressive.
   b. Exponential time and space complexity.
4. EL++ - (Reasoner: ELK):
   a. Fairly expressive, though not as expressive as SROIQ(D)
   b. Polynomial time complexity

OWL vs DL terminology

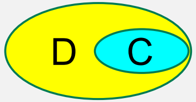
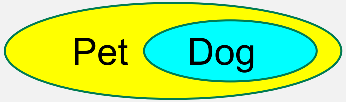
OWL	DL	Semantics
instance/individual	instance/individual	A member of a set
class	concept	Set of individuals
object property	role	Set of pairs of individuals
data property	concrete role	Set of pairs consisting of an individual linked to a data value
owl:Thing	Top ($\top$)	The complete domain of interpretation $\Delta^{\mathcal{I}}$
owl:Nothing	Bottom ($\bot$)	The empty set $\emptyset$

What can you say with a DL?

1. ***C subclassOf D***: the set C of individuals is a subset of the set D of individuals

   In DL: C is subsumed by D, or D subsumes C.

Manchester	**DL**	**Semantics**	**Example**

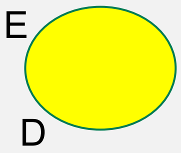
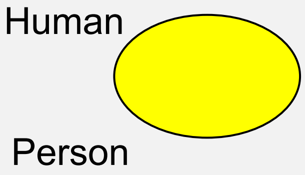
`Class: C`
`    SubClassOf: D`
`Class: D`
	$C \sqsubseteq D$		

What can you say with a DL?

2. ***D equivalentTo E***: the set D of individuals is equivalent to the set E of individuals

   In DL: D is equivalent to E, or E is equivalent to D.

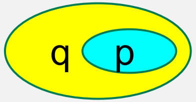
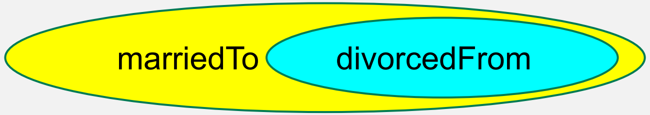
Manchester	**DL**	**Semantics**	**Example**

`Class: E`
`    EquivalentTo: D`
`Class: D`
	$E \equiv D$ which is shorthand for
	$E \sqsubseteq D, D \sqsubseteq E$
			

What can you say with a DL?

3. ***p subpropertyOf q***: the set p of pairs of individuals is a subset of the set q of pairs of individuals

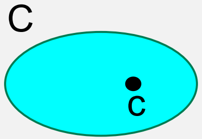
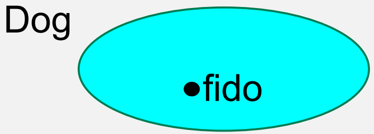
Manchester	DL	Semantics	Example

`ObjectProperty: p`
`    SubPropertyOf: q`
`ObjectProperty: q`
	$p \sqsubseteq q$		

What can you say with a DL? (cont.)

4. ***c instanceOf C***: the individual c is a member of the set C

In DL: Concept assertion

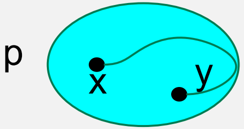
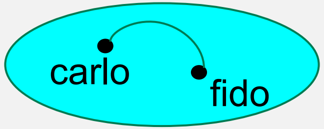
Manchester	**DL**	**Semantics**	**Example**
`Individual: c` `Types: C`	$C(c)$ $c : C$		

What can you say with a DL? (cont.)

5. Individual *x* is related to individual *y* via the property *p*

In DL: Role assertion

Manchester	**DL**	**Semantics**	**Example**

`ObjectProperty: p`
`Individual: x`
`    Facts: p  y`
`Individual:y`
	$p(x, y)$
	$(x, y) : p$
			owns 

DL assumptions

1. Open world assumption:
   a. Typical databases make a **closed world assumption**: This means the absence of information is often deemed meaningful. I.e., at EBI if you are not SAP, you are not an EBI employee.
   b. DLs make the **open world assumption**: No assumptions are made based on the absence of information. Inferences are made only based on:
      i. axioms and
      ii. explicitly stated information.

2. DLs **do not make the unique name assumption**. 2 differently named classes/properties/individuals can refer to the exact same class/property/individual.

Running example: humans and pets

```
Class: Pet
Class: Dog
    SubClassOf: Pet
Class: Cat
    SubClassOf: Pet
Class: Human

Individual: Doggy
    Types: Dog
Individual: Woof
    Types: Dog
Individual: Kitten
     Types: Cat
Individual: Mary
     Types: Human
Individual: John
     Types: Human
Individual: Sam
     Types: Human
```

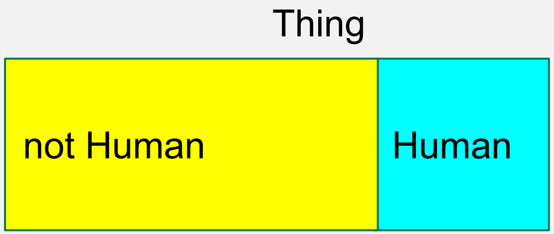
$$\mathcal{T} = \{Human \sqsubseteq \top, \; Dog \sqsubseteq Pet, \; Cat \sqsubseteq Pet\}.$$

$$\mathcal{A} = \{Dog(Woof), \; Dog(Doggy), \; Cat(Kitten), \; Human(Mary), \; Human(John), \; Human(Sam)\}.$$

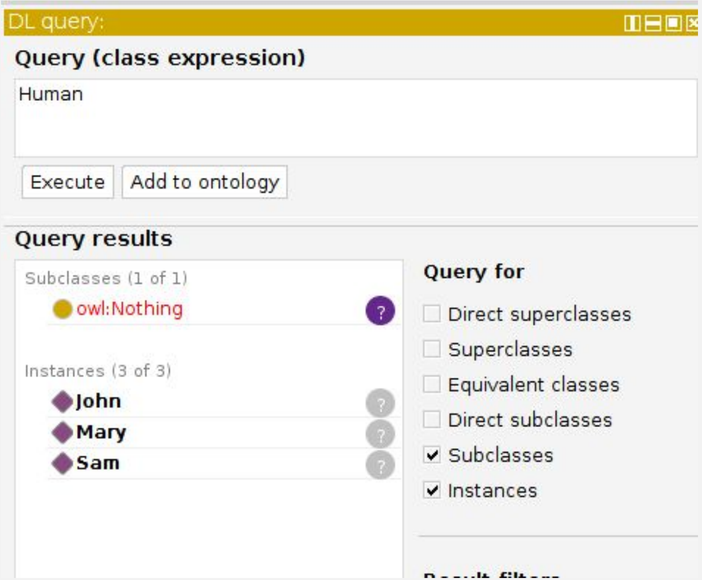
Negation $\neg C$

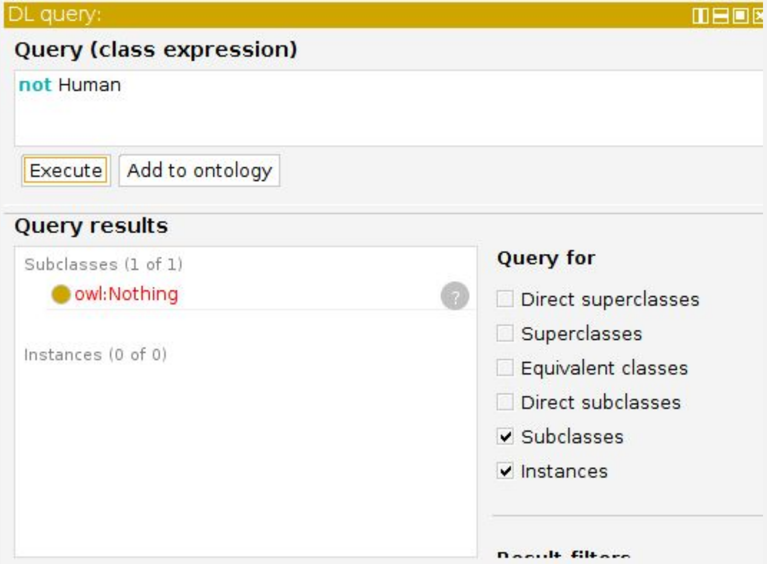
$$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$$

not C is the set of individuals in our domain of interpretation that does not belong to C.

Manchester	DL	Semantics
not Human	$\neg Human$	

Negation: running example

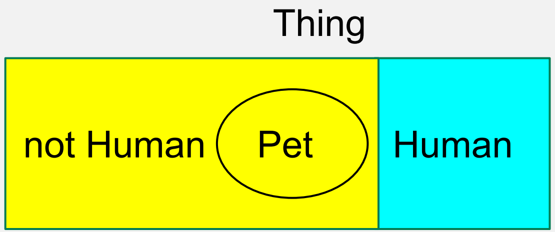




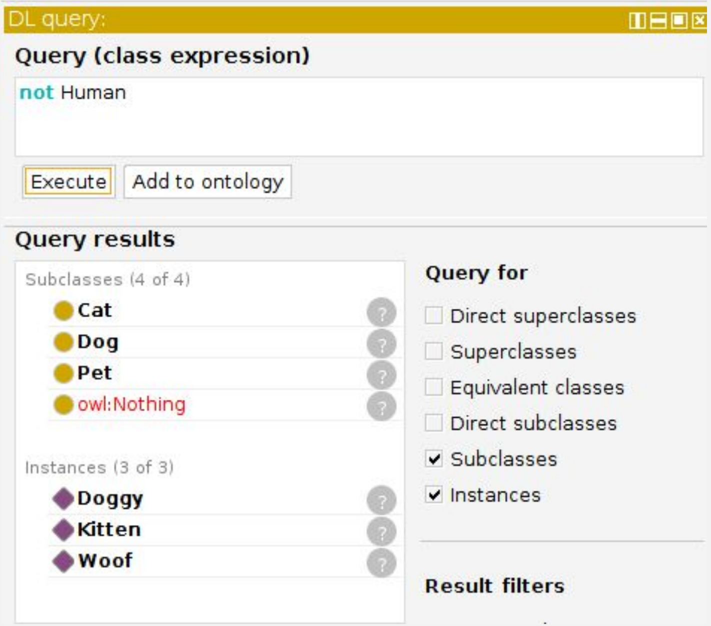
Why is *not Human* not showing *Woof, Doggy, Kitten* as instances?

Disjointness $C \sqsubseteq \neg D$

Manchester	DL	Semantics

`Class: Pet`
`DisjointWith: Human`
Or
`Class: Pet`
`SubClassOf: not Human`
	$Pet \sqsubseteq \neg Human$	

Disjointness: running example



Since a pet is either a cat or a dog, we also state that *Cat* is disjoint from *Dog*.

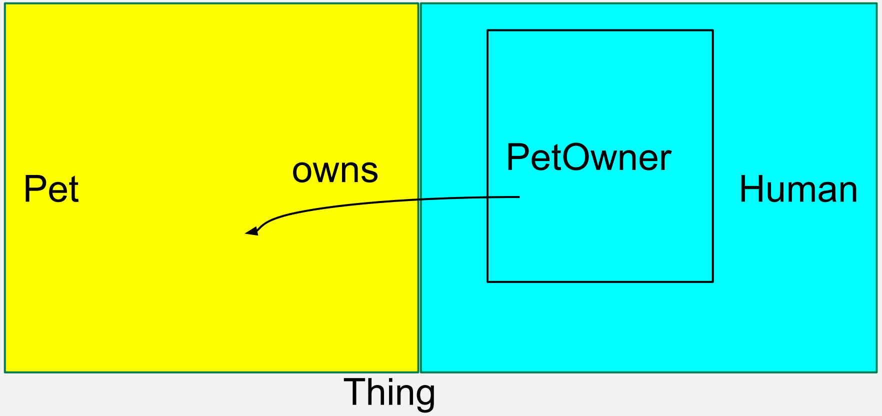
Qualified existential restriction $\exists r.C$

$$(\exists r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} | \text{there is an } e \in \Delta^{\mathcal{I}} \text{ with } (d,e) \in r^{\mathcal{I}} \text{ and } e \in C^{\mathcal{I}}\}$$

$\exists r.C$ is the set of individuals such that for each individual d there is at least 1 element e that is linked to an individual d of type C via the role r.

Read as: there exists an r-filler that belongs to C

Manchester	**DL**	**Semantics**

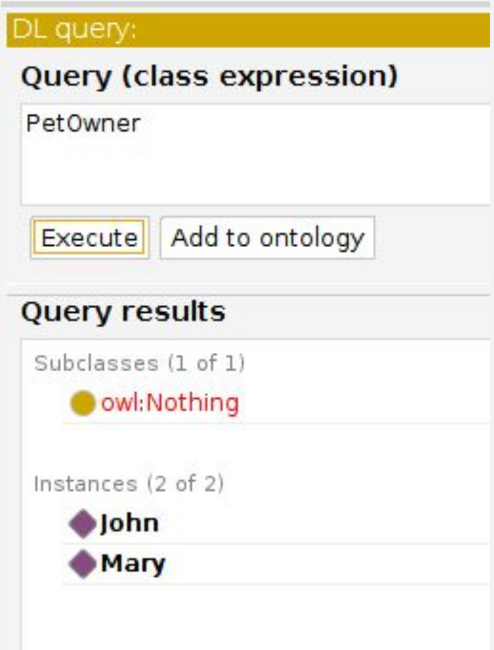
`ObjectProperty: owns`
`Class: PetOwner`
`EquivalentTo:`
`owns some Pet`
`SubClassOf: Human`
	$PetOwner \equiv \exists owns.Pet,$
	$PetOwner \sqsubseteq Human$
		

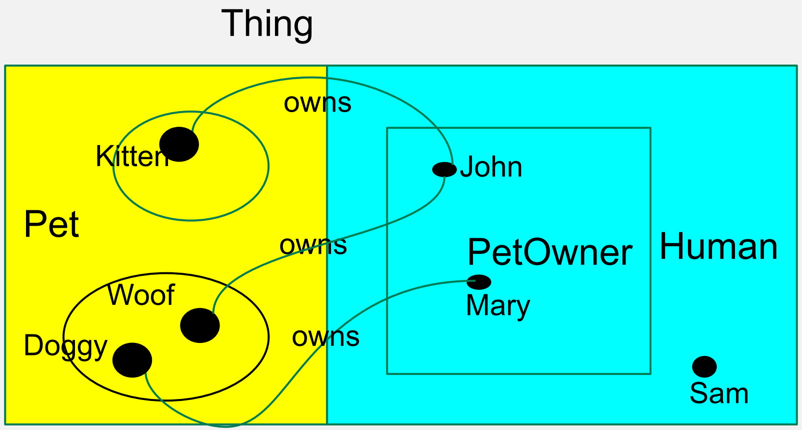
Qualified existential restriction: running example

```
Individual: Mary
    Facts:
        owns  Doggy

Individual: John
    Facts:
        owns  Kitten,
        owns  Woof
```

$$owns(Mary, Doggy),$$
$$owns(John, Kitten),$$
$$owns(John, Woof)$$





Qualified universal restriction $\forall r.C$

$$(\forall r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} | \text{for all } e \in \Delta^{\mathcal{I}}, \text{ if } (d,e) \in r^{\mathcal{I}}, \text{ then } e \in C^{\mathcal{I}}\}$$

$\forall r.C$ Is the set of individuals where for each individual d of the set, it holds that whenever d is linked to an individual e via r, then e is of type C.

Read as: all r-fillers are of type C

Manchester	**DL**

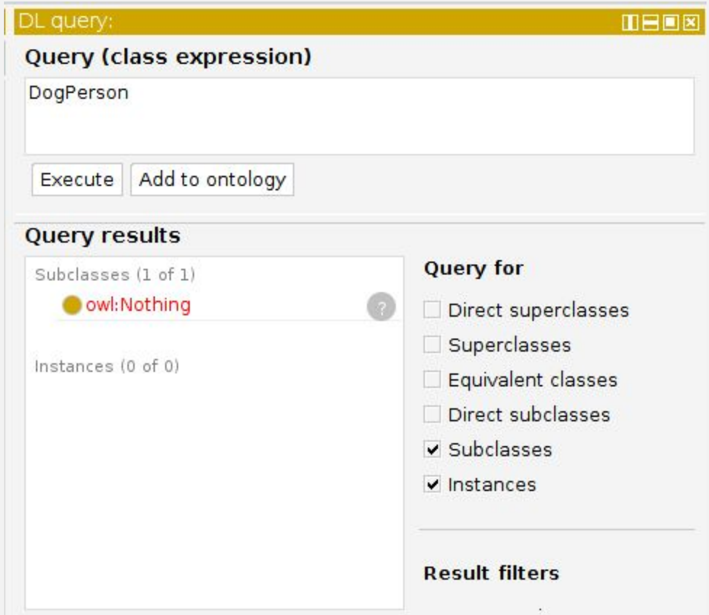
`Class: DogPerson`
`EquivalentTo: owns only Dog`
`SubclassOf: Human`
	$DogPerson \equiv \forall owns.Dog,$
	$DogPerson \sqsubseteq Human$

Qualified universal restriction: running example

A dog person is someone who owns only dogs.

```
Class: DogPerson
    EquivalentTo:
        owns only Dog
```

$$DogPerson \equiv \forall owns.Dog$$



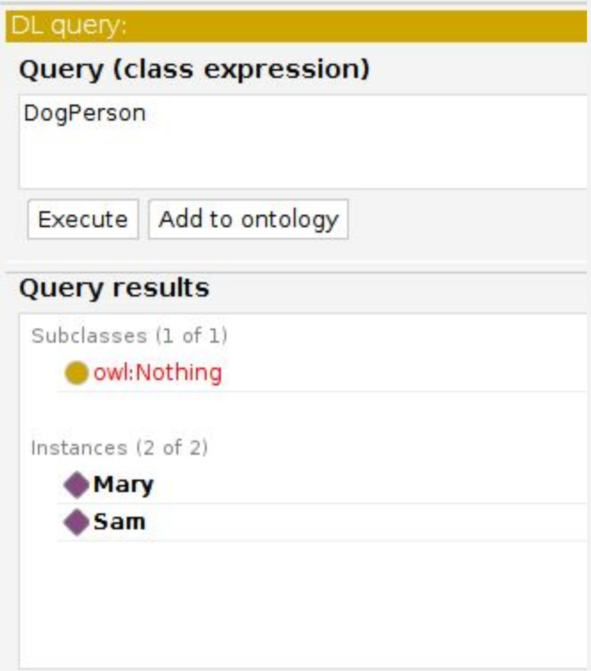
For *Mary* we said that she only has a dog. Why is it not showing up?

Qualified universal restriction: running example

```
Individual: Mary
   Types:
      not (owns some (not Dog))

Individual: Sam
   Types:
      not (owns some owl:Thing)
```

$$\neg(\exists owns.(\neg Dog))(Mary),$$
$$\neg(\exists owns.\top)(Sam)$$



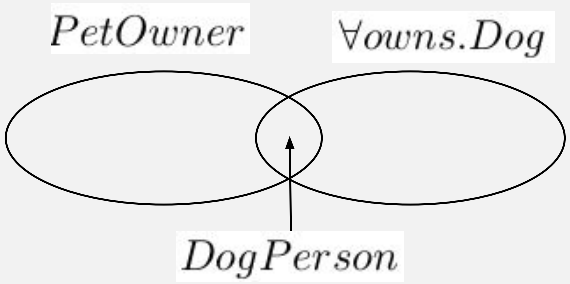
Why did *Sam* come up as someone that is a *DogPerson* when they do not have a dog?

Existential vs Universal restrictions comparison

Existential: *r some C* $\exists r.C$	**Universal:** *r only C* $\forall r.C$
Every individual has at least 1 link via *r* to an individual in *C*	Individuals may have zero links via *r*
Individuals may have links via *r* to individuals that are not of type *C*	Individuals have links via *r* to individuals only of type *C*
Individuals may have multiple links via *r* to multiple individuals of type *C*	
Individuals may have zero or more links via properties/roles other than *r*.	

Conjunction $C \sqcap D$

Manchester	DL	Semantics

`Class: DogPerson`
`EquivalentTo:`
`PetOwner`
`and (owns only Dog)`
	$DogPerson \equiv PetOwner \sqcap \forall owns.Dog$	

Disjunction: running example

Our assumption: Our domain deals only with cats and dogs. However, our ontology does not say that… To illustrate we will update our ontology.

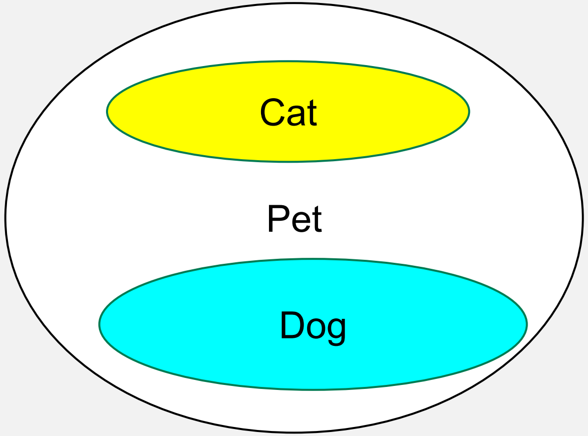
We add *Jill* as a *Human* that owns a pet that is not a cat or a dog.

```
Individual: APet
    Types: Pet

Individual: Jill
   Types:
        Human,
        not (owns some (Cat or Dog))
   Facts:
        owns APet
```

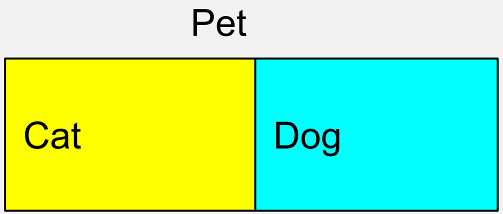
Disjunction: running example (cont.)

If our domain is only dealing with pets that are cats and dogs, why is this not giving an inconsistency?

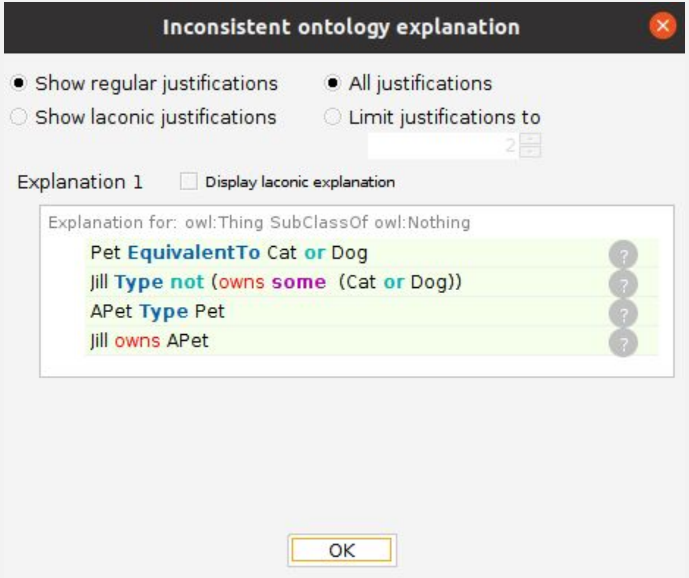


Disjunction $C \sqcup D$

Manchester	DL	Semantics

`Class: Pet`
`    EquivalentTo:`
`        Cat or Dog`
	$Pet \equiv Cat \sqcup Dog$	

Disjunction: running example



In general, to remove an inconsistency, you need to remove at least one of the justifications for the inconsistency. Here we will remove the fact that
Jill owns APet

Domain and range: running example

For illustration purposes we add x and y as individuals our ontology as follows:

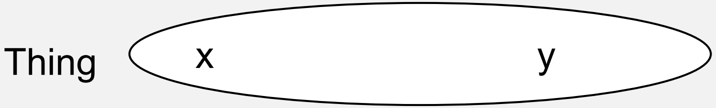
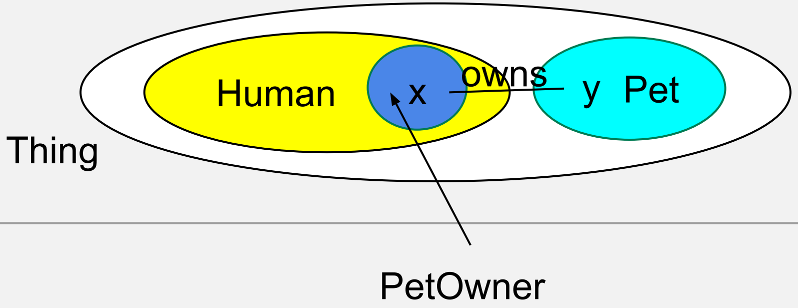
```
Individual: x
   Facts:
      owns   y

Individual: y
```

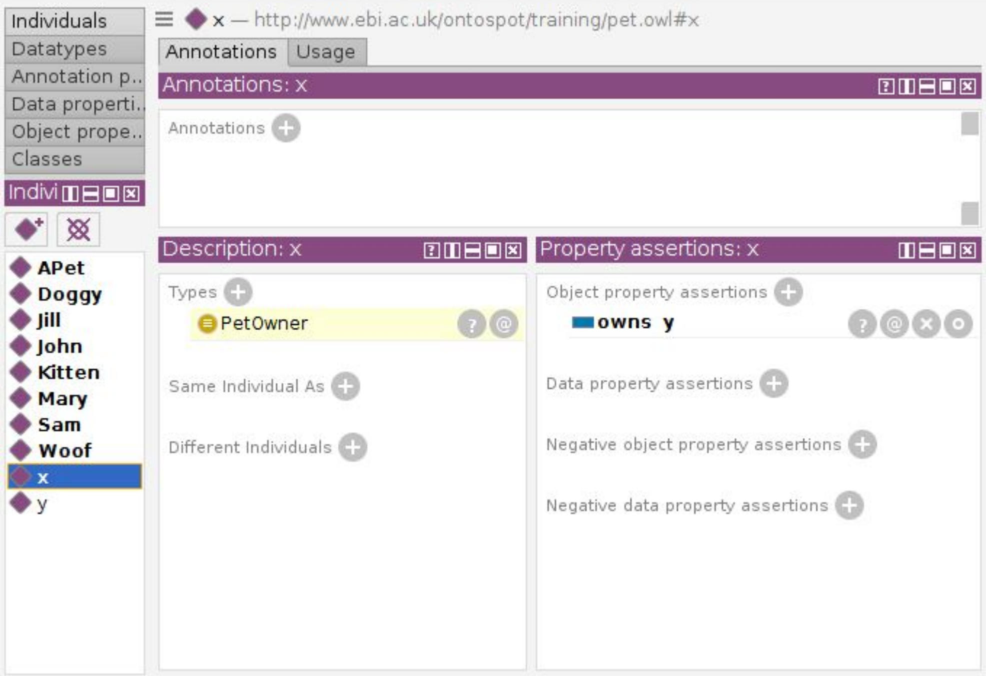
Now run the reasoner. What happens? … Nothing …

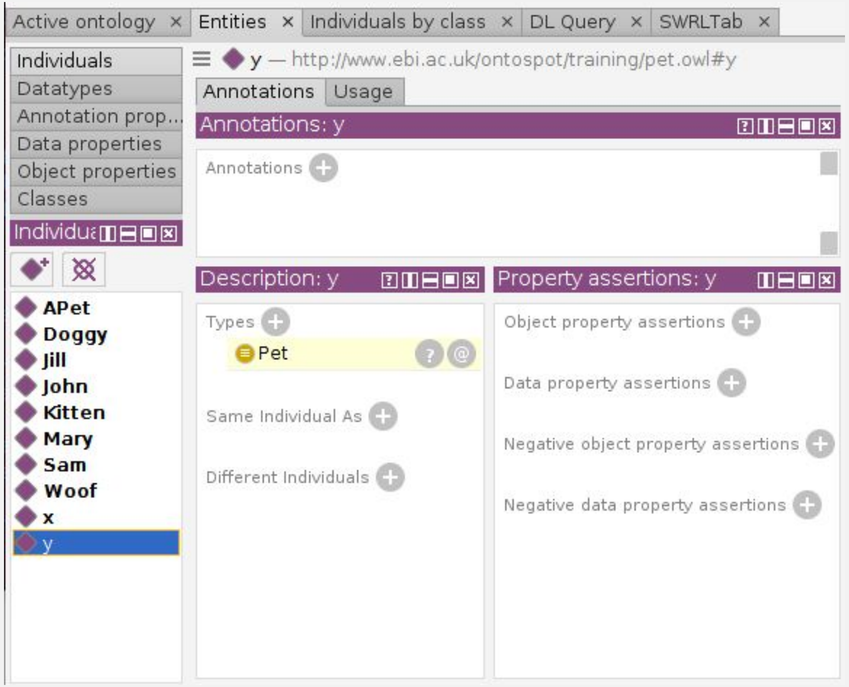
Domain and range

Manchester	DL	Semantics

`ObjectProperty: owns`
`Domain: Human`
`Range: Pet`
	$\exists owns.\top \sqsubseteq Human,$
	$\top \sqsubseteq \forall owns.Pet$
		1. *x*, *y* have no type information
		
		2. We link *x*, *y* via *owns*
		

Domain and range: running example





Lastly

1. If you really want to understand this well, I recommend you read “An introduction to description logics”: https://www.cambridge.org/core/books/an-introduction-to-description-logic/6D329698AFC2E6C6C5C15801ED9B6D07
2. Questions?