

# Introduction to Ontology Semantics and Reasoning



**Henriette Harmse**

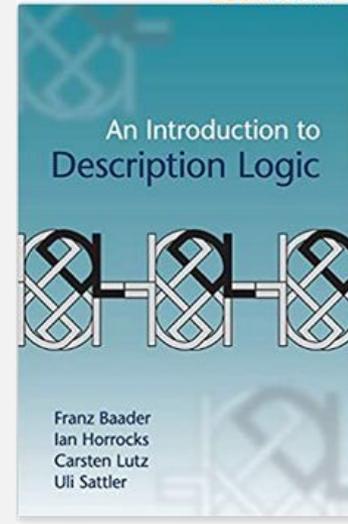
**Ontology tools project lead**

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

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

Displayed inferences ELK Initialization

---

**Class inferences**

- Satisfiability (9 ms total/0 ms average)
- Equivalent Classes (3 ms total/0 ms average)
- Superclasses (3 ms total/0 ms average)
- Class Instances (3 ms total/0 ms average)
- Disjoint Classes (92 ms total/1 ms average)

---

**Object property inferences**

- Satisfiability (35 ms total/0 ms average)
- Domains (33 ms total/0 ms average)
- Ranges (10 ms total/0 ms average)
- Equivalent Properties (0 ms total/0 ms average)
- Super Properties (8 ms total/0 ms average)
- Inverse properties (5 ms total/0 ms average)

---

**Data property inferences**

- Domains (0 ms total/0 ms average)
- Equivalent Properties (0 ms total/0 ms average)
- Super Properties (0 ms total/0 ms average)

---

**Individual inferences**

- Types (14 ms total/0 ms average)
- Object Property Assertions (4 ms total/0 ms average)
- Data Property Assertions (3 ms total/0 ms average)
- Same Individuals (0 ms total/0 ms average)

---

Show Inferences

# Forces driving DL research

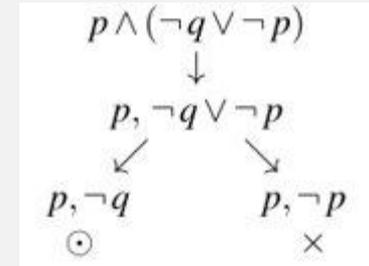
1. **Expressivity of a logic:** How much or little you can say using a specific logic
2. **Computational complexity of an algorithm:** How much space or time an algorithm will take to complete.
3. **Decidability:** An algorithm is **decidable** for a decision problem if it always will be able to give 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:  $a=Zuko, b=bark \Rightarrow$  true
  - b. Interpretation 2:  $a=Margo, b=bark \Rightarrow$  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

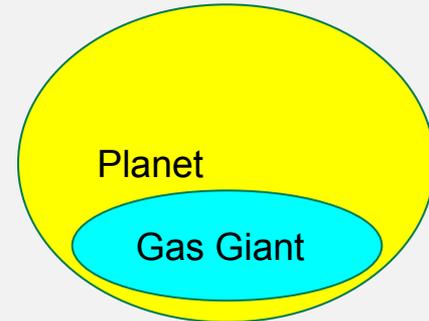
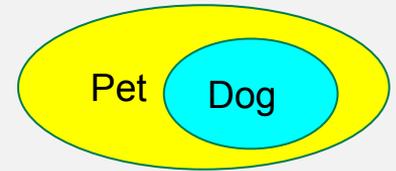
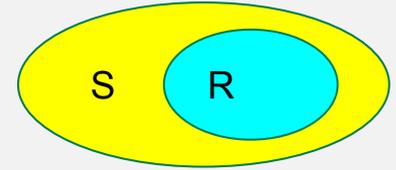
Semantic tableau



# 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.:  
 $S=\text{Pet}$ ,  $R=\text{Dog}$ ,  $P = \text{pets that are dogs}$ ,  
 $S=\text{Planet}$ ,  $R=\text{Gas Giant}$ ,  $P = \text{planets that are gas giants}$ .
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

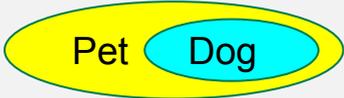
# OWL vs DL terminology

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 ( $\perp$ )	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

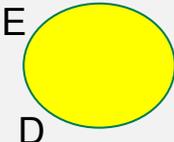
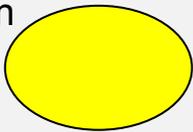
*In DL*: C is subsumed by D, or D subsumes C.

Manchester	DL	Semantics	Example
Class: C SubClassOf: D Class: D	$C \sqsubseteq D$		

# What can you say with a DL?

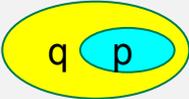
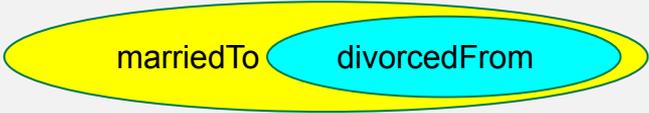
2. ***D equivalentTo E***: the set D of individuals is equivalent to the set E of individuals

***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 $E \sqsubseteq D, D \sqsubseteq E$		Human  Person

# What can you say with a DL?

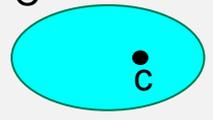
3. ***p* subpropertyOf *q***: the set *p* of pairs of individuals is a subset of the set *q* of pairs of individuals

Manchester	DL	Semantics	Example
ObjectProperty: <i>p</i> SubPropertyOf: <i>q</i> ObjectProperty: <i>q</i>	$p \sqsubseteq q$		

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

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

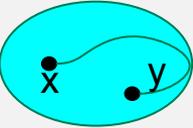
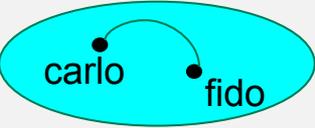
*In DL*: Concept assertion

Manchester	DL	Semantics	Example
Individual: $c$ Types: $C$	$C(c)$ $c : C$	$C$  A light blue oval representing a set $C$ with a small black dot inside labeled $c$ .	Dog  A light blue oval representing a set $Dog$ with a small black dot inside labeled $fido$ .

# 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: $p \quad y$ Individual: $y$	$p(x, y)$ $(x, y) : p$	$p$ 	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
```

$$\mathcal{T} = \{ \textit{Human} \sqsubseteq \top, \\ \textit{Dog} \sqsubseteq \textit{Pet}, \\ \textit{Cat} \sqsubseteq \textit{Pet} \}.$$
$$\mathcal{A} = \{ \textit{Dog}(\textit{Woof}), \\ \textit{Dog}(\textit{Doggy}), \\ \textit{Cat}(\textit{Kitten}), \\ \textit{Human}(\textit{Mary}), \\ \textit{Human}(\textit{John}), \\ \textit{Human}(\textit{Sam}) \}.$$

# Negation $\neg C$

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

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

Manchester	DL	Semantics
not Human	$\neg Human$	 <p>The diagram illustrates the semantic interpretation of the expression 'not Human'. It shows a large rectangle labeled 'Thing' at the top. Inside this rectangle, there are two smaller, side-by-side rectangles. The left rectangle is yellow and labeled 'not Human'. The right rectangle is cyan and labeled 'Human'. This visualizes that the set of 'not Human' individuals is a subset of the set of 'Thing' individuals, and is disjoint from the set of 'Human' individuals.</p>

# Negation: running example

DL query: ⏏ ⏏ ⏏ ⏏

Query (class expression)

Human

Execute Add to ontology

---

Query results

Subclasses (1 of 1)

- owl:Nothing

Instances (3 of 3)

- John
- Mary
- Sam

Query for

- Direct superclasses
- Superclasses
- Equivalent classes
- Direct subclasses
- Subclasses
- Instances

Result filters

DL query: ⏏ ⏏ ⏏ ⏏

Query (class expression)

not Human

Execute Add to ontology

---

Query results

Subclasses (1 of 1)

- owl:Nothing

Instances (0 of 0)

Query for

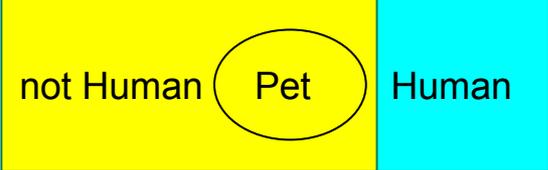
- Direct superclasses
- Superclasses
- Equivalent classes
- Direct subclasses
- Subclasses
- Instances

Result filters

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

# Disjointness

$$C \sqsubseteq \neg D$$

Manchester	DL	Semantics
<p data-bbox="104 347 525 456">Class: Pet DisjointWith: Human</p> <p data-bbox="162 473 208 506"><b>Or</b></p> <p data-bbox="104 532 575 631">Class: Pet SubClassOf: not Human</p>	<p data-bbox="598 430 877 467"><math>Pet \sqsubseteq \neg Human</math></p>	<p data-bbox="1406 390 1499 428">Thing</p> 

# Disjointness: running example

DL query: ⏏ ⏏ ⏏ ⏏

Query (class expression)

not Human

---

Query results

Subclasses (4 of 4)

- Cat** ?
- Dog** ?
- Pet** ?
- owl:Nothing** ?

Instances (3 of 3)

- Doggy** ?
- Kitten** ?
- Woof** ?

**Query for**

- Direct superclasses
- Superclasses
- Equivalent classes
- Direct subclasses
- Subclasses
- Instances

---

**Result filters**

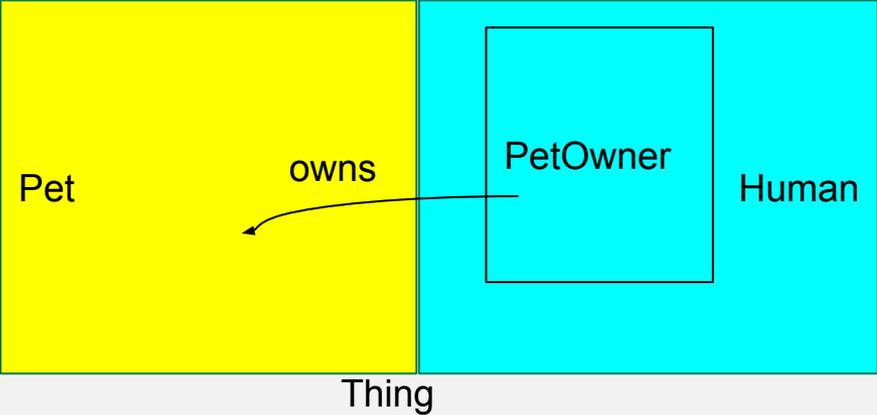
Since a pet is either a cat or a dog, we also state that *Cat* is disjoint from *Dog*.

# Qualified existential restriction $\exists r.C$

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

$\exists r.C$  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
<pre>ObjectProperty: owns Class: PetOwner   EquivalentTo:     owns some Pet   SubClassOf: Human</pre>	<pre><math>PetOwner \equiv \exists \text{owns}.Pet,</math> <math>PetOwner \sqsubseteq Human</math></pre>	

# Qualified existential restriction: running example

Individual: Mary  
Facts:  
owns Doggy

Individual: John  
Facts:  
owns Kitten,  
owns Woof

DL query:

Query (class expression)

PetOwner

Execute

Add to ontology

Query results

Subclasses (1 of 1)

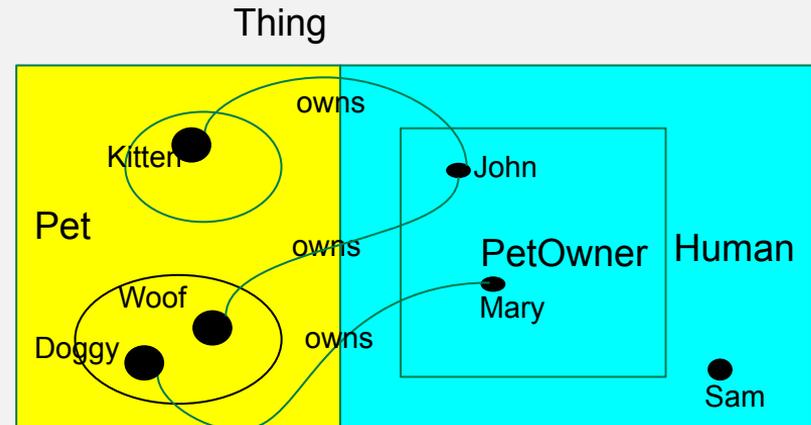
● owl:Nothing

Instances (2 of 2)

◆ John

◆ Mary

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



# Qualified universal restriction

 $\forall r.C$ 

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

 $\forall r.C$ 

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
<pre>Class: DogPerson   EquivalentTo: owns only Dog   SubclassOf: Human</pre>	<pre><i>DogPerson</i> <math>\equiv</math> <math>\forall \text{owns}.Dog</math>, <i>DogPerson</i> <math>\sqsubseteq</math> <i>Human</i></pre>

# Qualified universal restriction: running example

A dog person is someone who owns only dogs.

Class: DogPerson  
EquivalentTo:  
owns only Dog

$DogPerson \equiv \forall owns.Dog$

The screenshot shows a web interface for running a DL query. At the top, a yellow header bar contains the text "DL query:" and window control icons. Below this, a white box labeled "Query (class expression)" contains the text "DogPerson". Underneath the query box are two buttons: "Execute" and "Add to ontology".

The "Query results" section is divided into two parts. On the left, under "Subclasses (1 of 1)", there is a single entry: "owl:Nothing" with a yellow circle icon and a question mark icon to its right. Below this, under "Instances (0 of 0)", the area is empty. On the right side of the results section, there is a "Query for" panel with a list of checkboxes: "Direct superclasses", "Superclasses", "Equivalent classes", "Direct subclasses", "Subclasses" (checked), and "Instances" (checked). Below the checkboxes is a "Result filters" section which is currently empty.

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)

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

DL query:

Query (class expression)

DogPerson

Execute

Add to ontology

Query results

Subclasses (1 of 1)

● owl:Nothing

Instances (2 of 2)

◆ Mary

◆ Sam

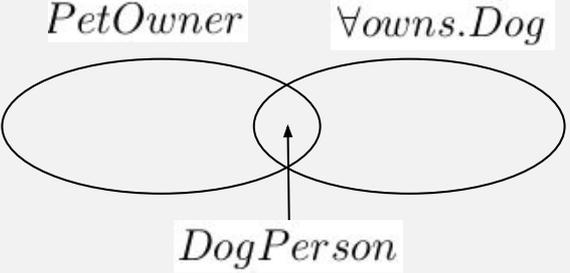
Why did *Sam* come up as someone that is a *DogPerson* when they do not have a dog?

# Existential vs Universal restrictions comparison

<b>Existential: <math>r</math> some <math>C</math></b> $\exists r.C$	<b>Universal: <math>r</math> only <math>C</math></b> $\forall r.C$
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
<p>Class: DogPerson EquivalentTo:   PetOwner   and (owns only Dog)</p>	<p><math>DogPerson \equiv PetOwner \sqcap \forall owns.Dog</math></p>	 <p>The diagram shows two overlapping ovals. The left oval is labeled <i>PetOwner</i> and the right oval is labeled <math>\forall owns.Dog</math>. The intersection of the two ovals is indicated by an upward-pointing arrow from the label <i>DogPerson</i> below.</p>

DL query:

Query (class expression)

DogPerson

Execute Add to ontology



Query results

Subclasses (1 of 1)

- owl:Nothing

Instances (1 of 1)

- Mary

# 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

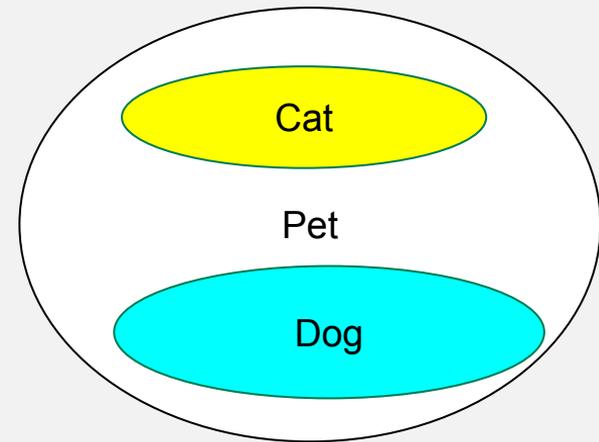
Individual: Jill
  Types:
    Human,
    not (owns some (Cat or Dog))
  Facts:
    owns APet
```

# Disjunction: running example (cont.)

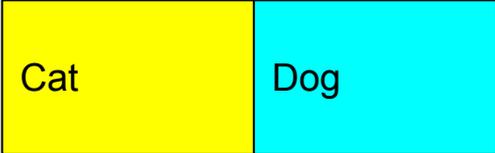
The screenshot shows a web-based ontology editor interface. On the left is a sidebar with a tree view of classes: APet, Doggy, jill (selected), John, Kitten, Mary, Sam, and Woof. The main area is divided into several panels:

- Annotations: jill**: A panel with a plus sign to add annotations.
- Description: jill**: A panel showing the class description for 'jill'. Under 'Types', it lists: Human, not (owns some (Cat or Dog)), and PetOwner. The 'not (owns some (Cat or Dog))' type is highlighted in yellow.
- Property assertions: jill**: A panel showing an object property assertion: owns APet.

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
<p data-bbox="104 416 452 565">Class: Pet EquivalentTo: Cat or Dog</p>	<p data-bbox="490 467 805 514"><math>Pet \equiv Cat \sqcup Dog</math></p>	<p data-bbox="1070 372 1132 401">Pet</p> 

# Disjunction: running example

### Inconsistent ontology explanation ✕

Show regular justifications     All justifications  
 Show laconic justifications     Limit justifications to

Explanation 1     Display laconic explanation

Explanation for: owl:Thing SubClassOf owl:Nothing

- Pet **EquivalentTo** Cat **or** Dog ?
- Jill **Type not** (**owns some** (Cat **or** Dog)) ?
- APet **Type** Pet ?
- Jill **owns** APet ?

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*

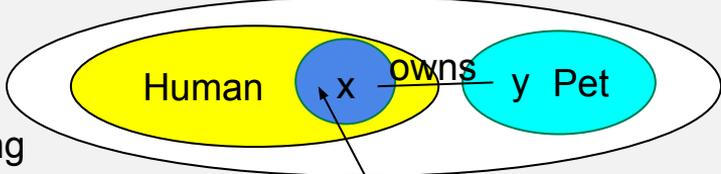
# Domain and range: running example

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
<p data-bbox="73 412 498 547">ObjectProperty: owns Domain: Human Range: Pet</p>	<p data-bbox="562 437 952 547"><math>\exists \text{owns}.\top \sqsubseteq \text{Human},</math> <math>\top \sqsubseteq \forall \text{owns}.\text{Pet}</math></p>	<p data-bbox="1010 352 1572 390">1. <math>x, y</math> have no type information</p> <p data-bbox="1010 437 1721 500">Thing </p> <p data-bbox="1039 587 1470 625">2. We link <math>x, y</math> via <i>owns</i></p> <p data-bbox="993 653 1785 827">Thing </p>

# Domain and range: running example

Individuals x — <http://www.ebi.ac.uk/ontospot/training/pet.owl#x>

Annotations Usage

Annotations: x

Description: x

Property assertions: x

Types

- PetOwner

Object property assertions

- owns y

Data property assertions

Negative object property assertions

Negative data property assertions

Same Individual As

Different Individuals

Individuals

- APet
- Doggy
- Jill
- John
- Kitten
- Mary
- Sam
- Woof
- x
- y

Active ontology x Entities x Individuals by class x DL Query x SWRLTab x

Individuals y — <http://www.ebi.ac.uk/ontospot/training/pet.owl#y>

Annotations Usage

Annotations: y

Description: y

Property assertions: y

Types

- Pet

Object property assertions

Data property assertions

Negative object property assertions

Negative data property assertions

Same Individual As

Different Individuals

Individuals

- APet
- Doggy
- Jill
- John
- Kitten
- Mary
- Sam
- Woof
- x
- y

# 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?

