# **Graphs**

**Relation:** a subset of the cartesian product, the arity of the relation depends by the number of sets that we look into the product

R artify 2 R ⊆ N x N = N2 { < x, y > | x € N and y € N }

R artify 4 R ⊆ A x B x C x D { < x, y, z, w > |

A relation is a set of tuples, a binary relation is a set of tuples (in this case pairs), elements are ordered.

Rn ⊆ Xn

**Function:** a relation that associatines to an input at most a value (could have also no value)

R ⊆ N x Z

This is NOT a function. I do not associate to one element at most one element

| 0 | 0 |
| --- | --- |
| 1 | -1 |
| 1 | +1 |
| 2 | -2 |
| 2 | +2 |

R ⊆ Z x N

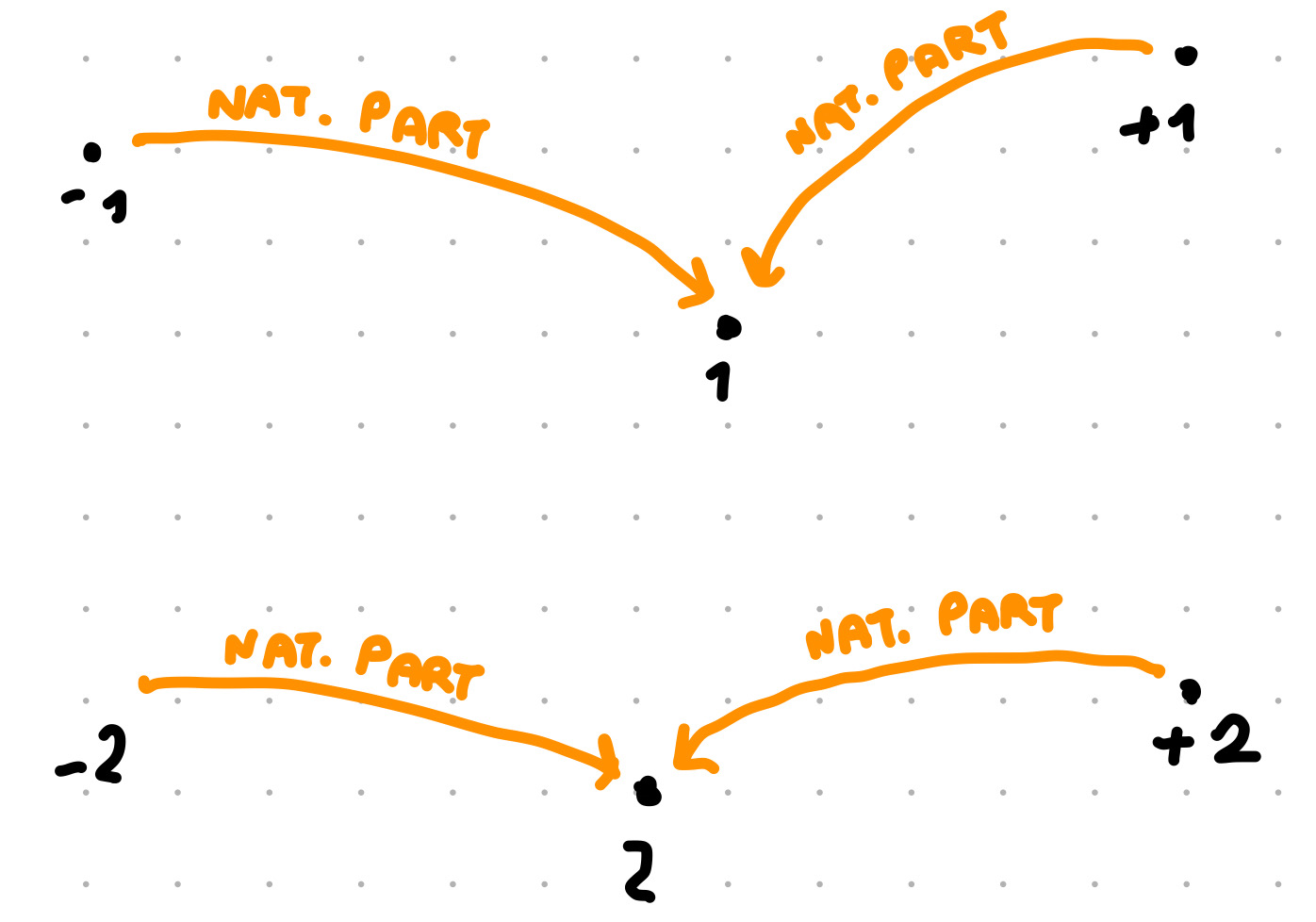
This is a function (inverse of the one before)

| 0 | 0 |
| --- | --- |
| -1 | 1 |
| +1 | 1 |
| -2 | 2 |
| +2 | 2 |

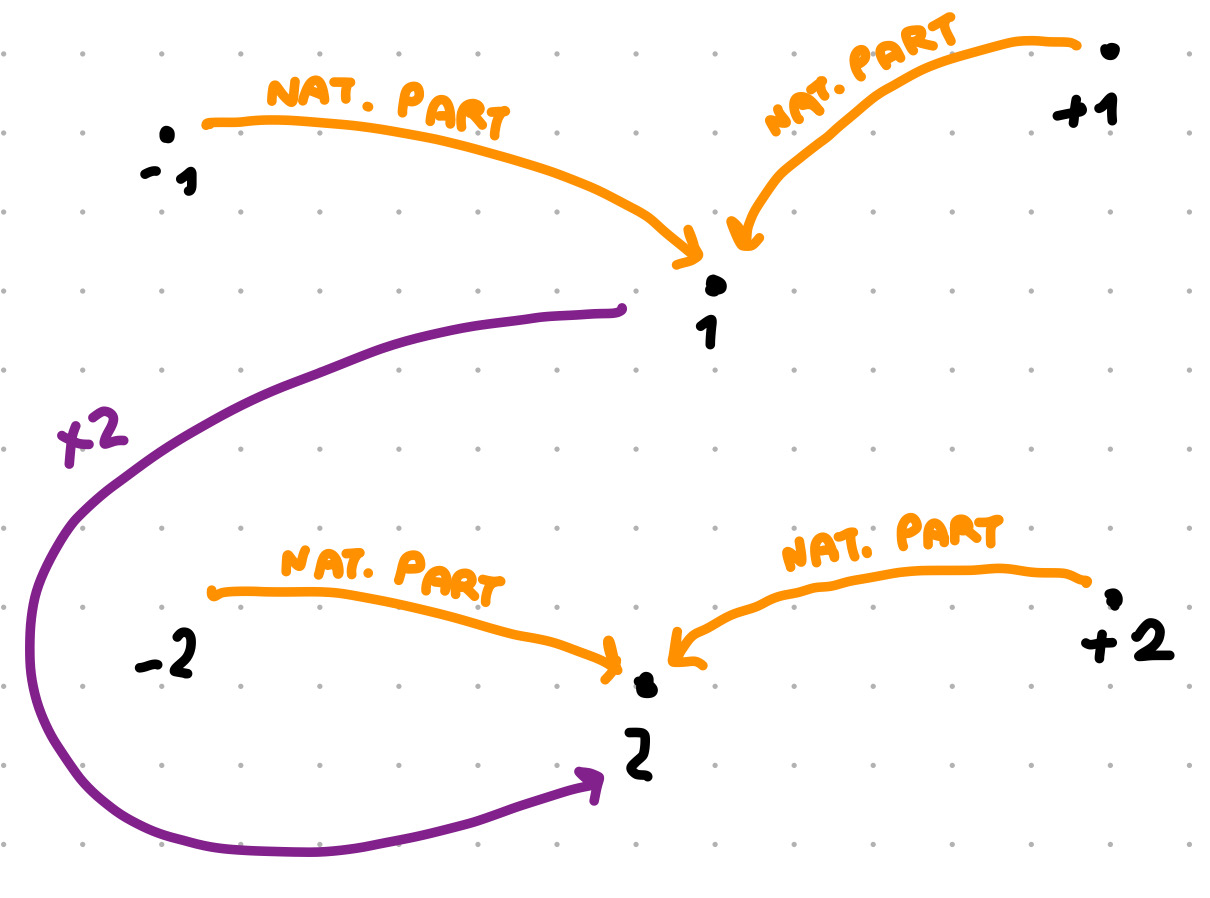
* if we take a function the inverse of that function may not be a function
* if a function is injective (1 to 1) is still a function

In binary relation is possible to view a relation as a graph, we can have as nodes of the graph the elements of the input and of the output. We add an edge between two nodes if the relation exists between the nodes

We can give a name to the relation name *natural part*



Add others relations



OWL have a concept to specify that the relation *natural part* is the same as *absolute value* are equivalent

Two relations are equivalent when their extension is the same, have the very same pairs of objects

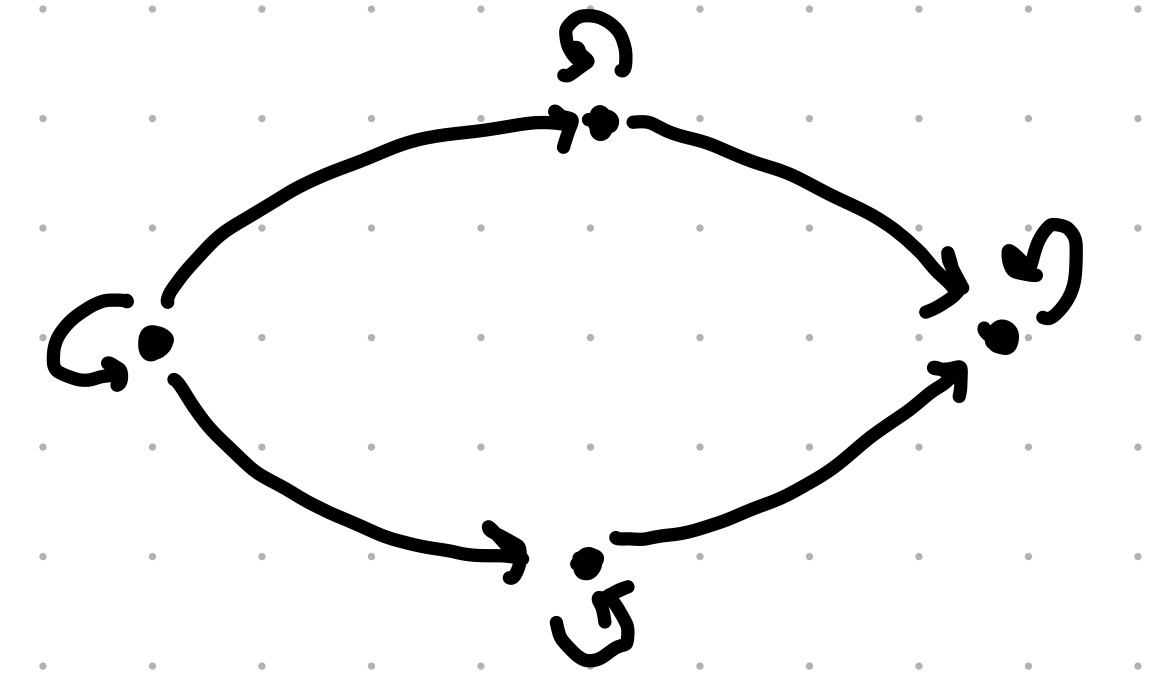
Being a function is a property than only some relations have

# Properties

## Reflexive

R is reflexive if and only if forall x <x,x> € IR

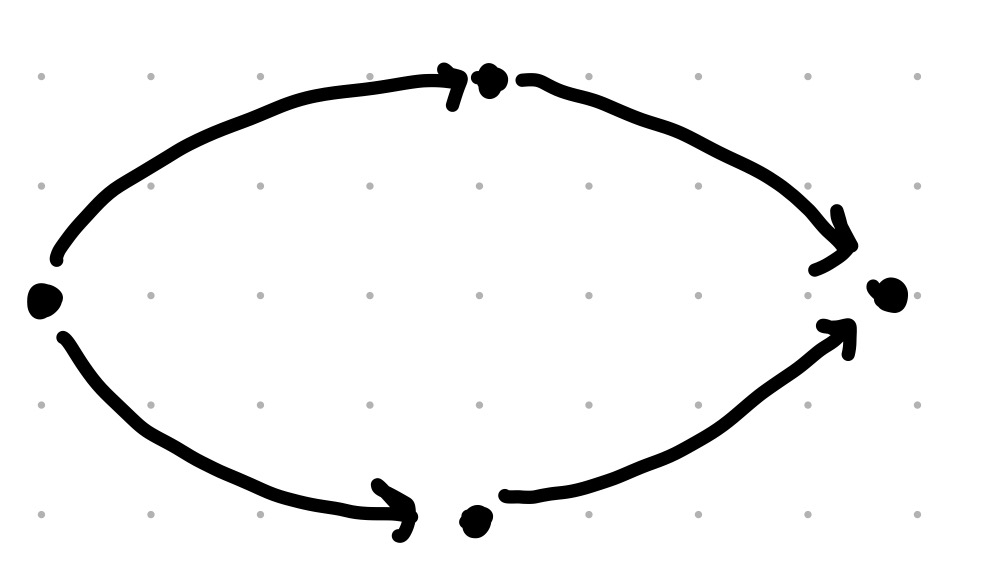
↓ is reflexive



## Irreflexive

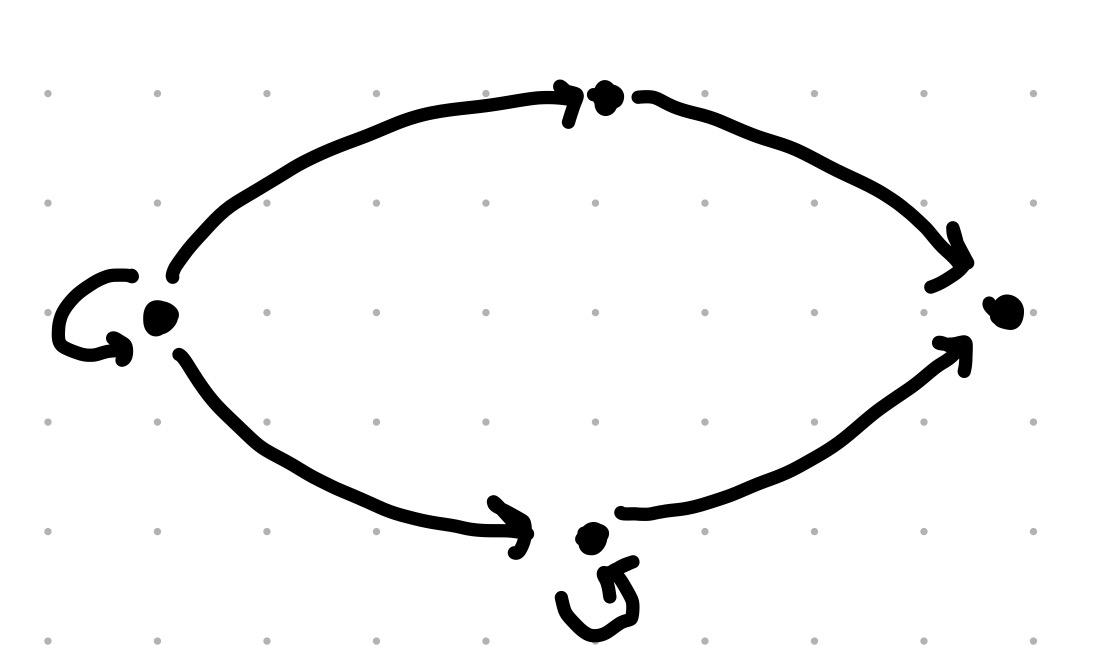
R is irreflexive if and only if forall x <x,x> ∉ IR

↓ is irrefliexive



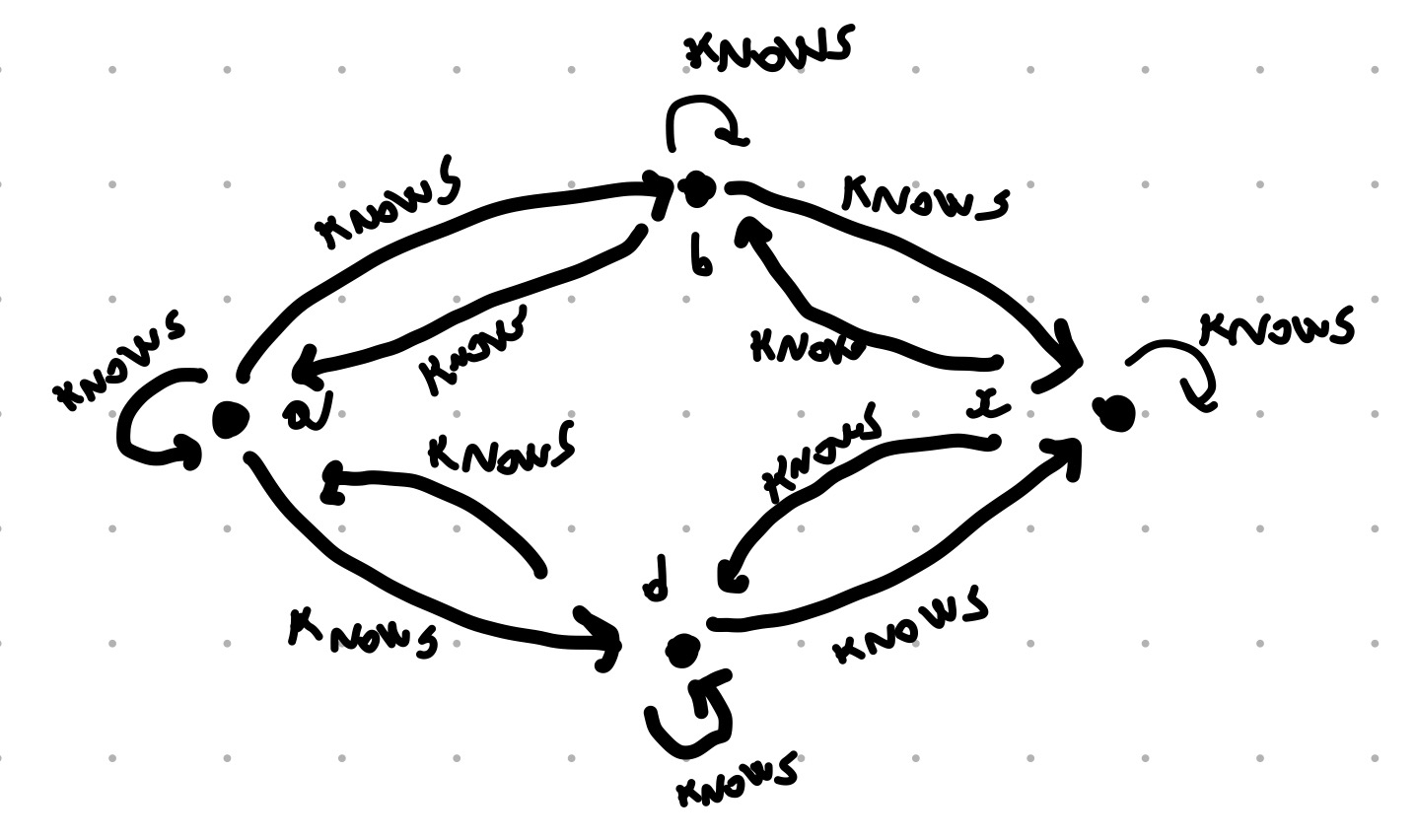
## Neither reflexive nor irreflexive

Given a relation there is the possibility of having not the reflexive relation but also not the irreflexive relation



## Symmetric

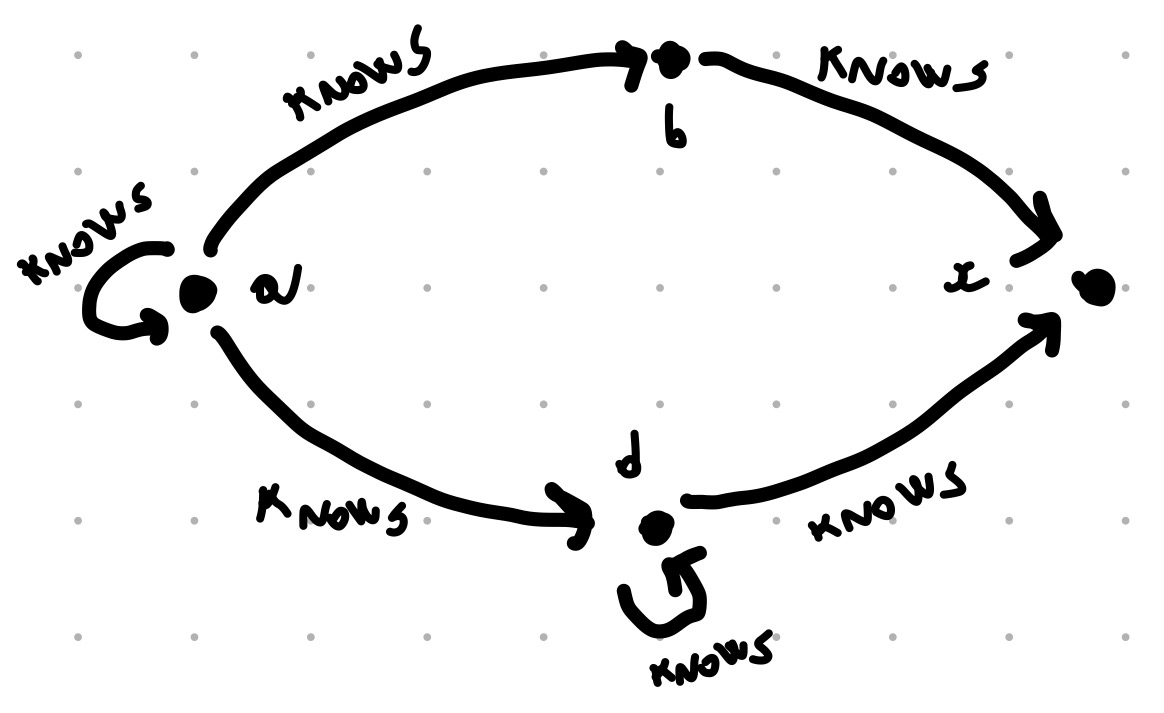
R is symmetric if and only if forall x,y if <x,y> € IR then <y,x> € IR



## Asymmetric

R is asymmetric if and only if forall x,y if <x,y> € IR then <y,x> ∉ IR

## Neither symmetric or asymmetric



## Antisymmetric

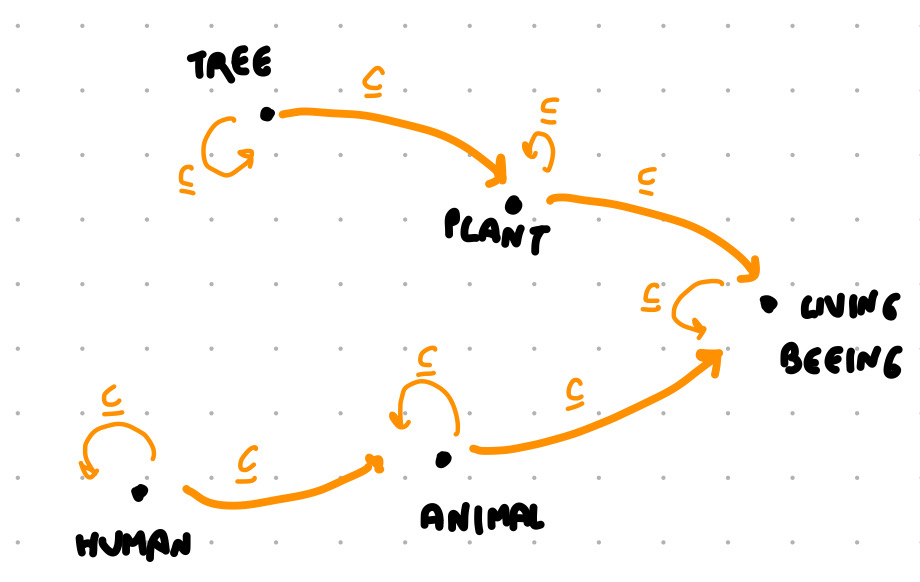
R is antisymmetric if and only if forall x,y if <x,y> € IR and <y,x> € IR then x = y

You don’t have <x,y> <y,x> except when x and y are the same element

Every asymmetric relation is also antisymmetric, but there are relations that are antisymmetric and not asymmetric



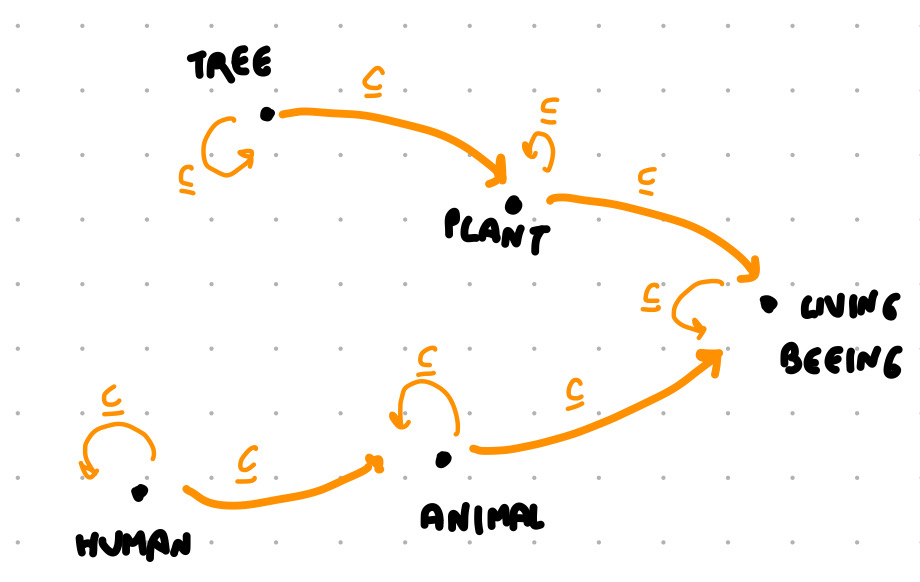
You can’t go back unless is on yourself



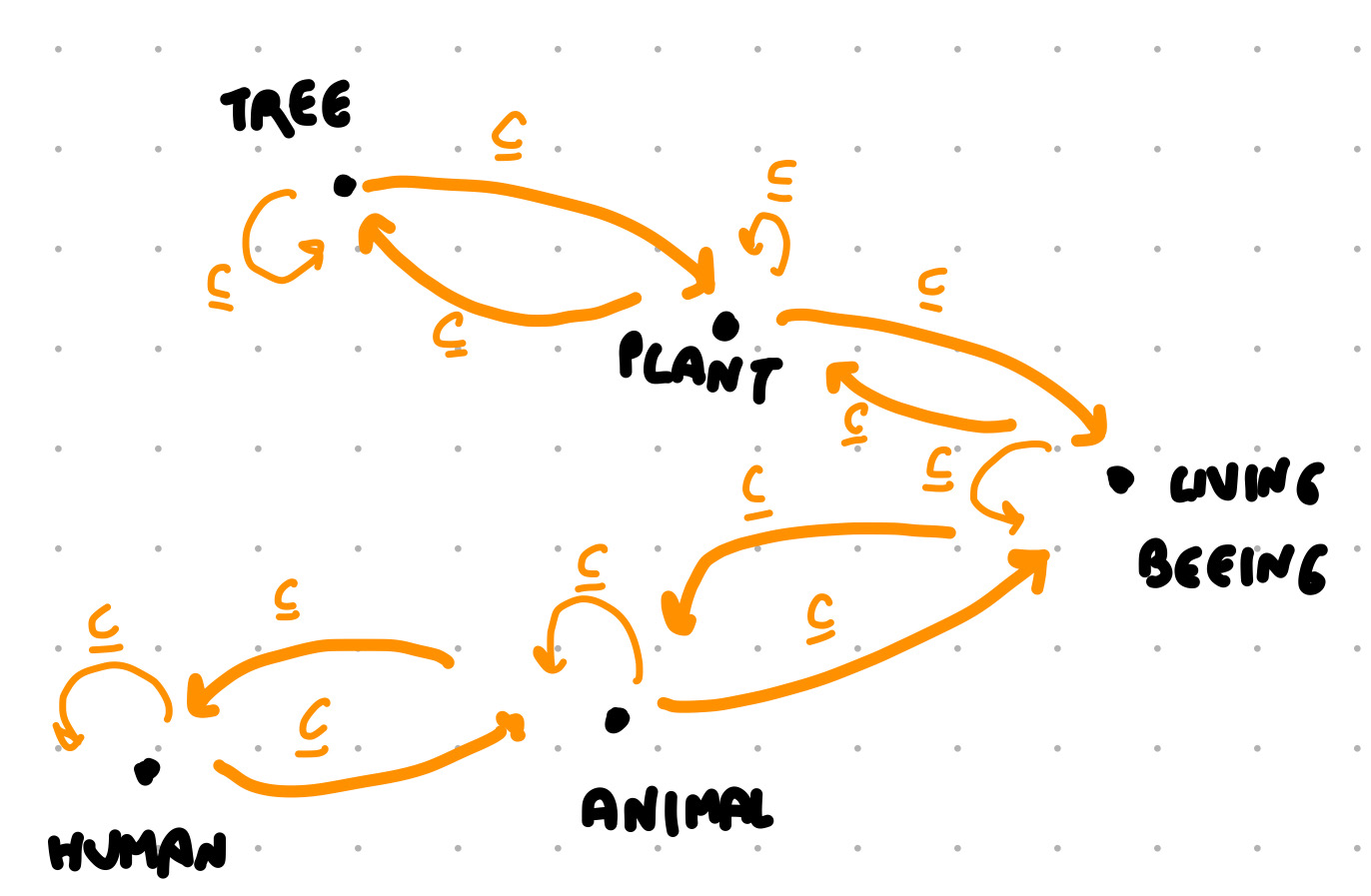
## Transitive

R is transitive if <x,y> € IR then <y,z> € IR then <x,z> € IR

↓ this relation is not transitive



↓ this relation is transitive



## Equivalence

R is an equivalence relation if and only if is

* reflexive
* symmetric
* transitive

2 = 4/2 = 8/4

## Partial order

R is a partial ordered if and only if is

* reflexive
* asymmetric
* transitive

2 ≤ 2 symmetric

2 ≤ 4 antisymmetric

POSET: partially ordered set

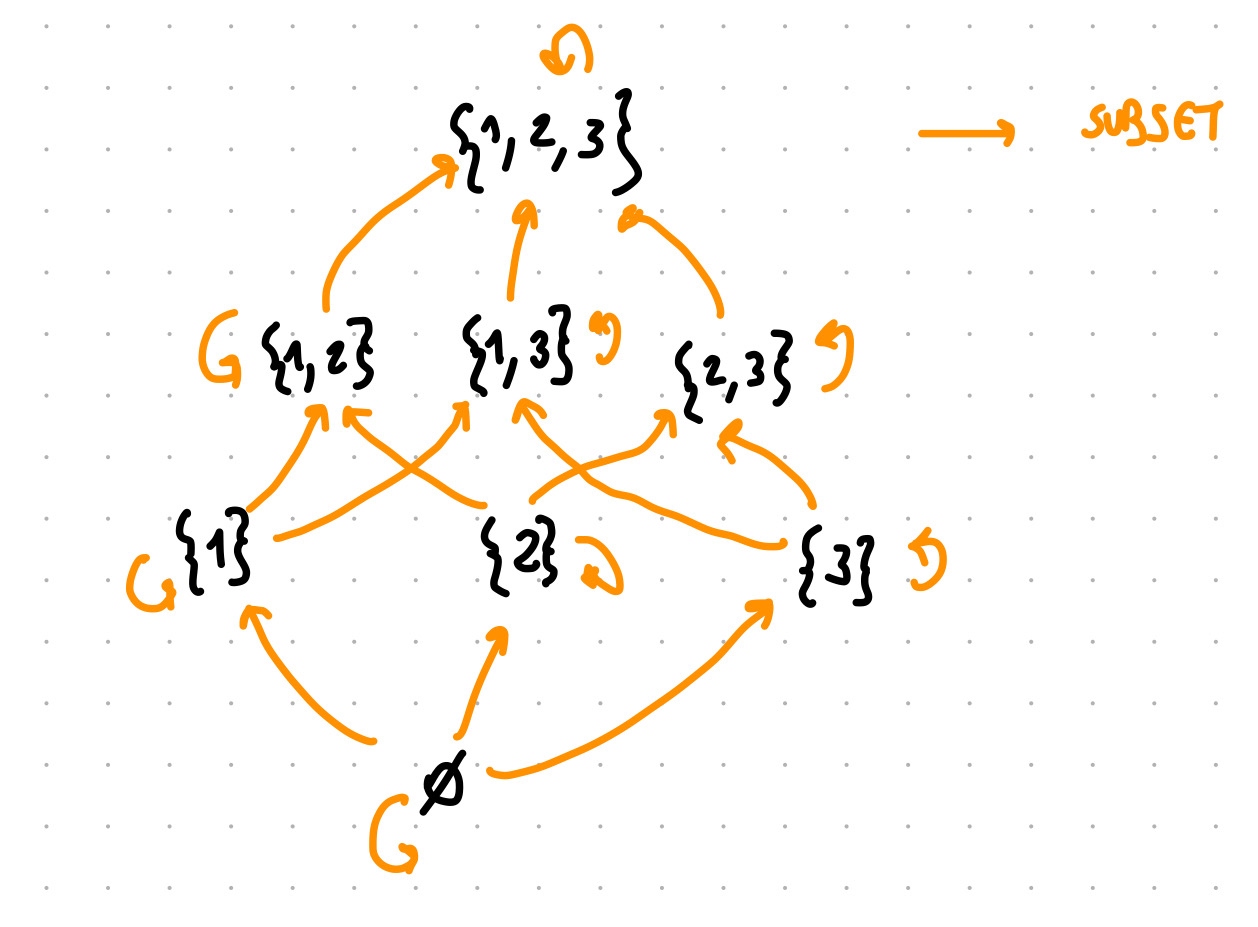
## Powerset

Set of all the subset related by the ⊆ relation

A = {1,2,3}

P(A) = { ø {1}, {2}, {3}, {1,2}, {1,3}, {2,3}, {1,2,3} }

Each element is subset of itself

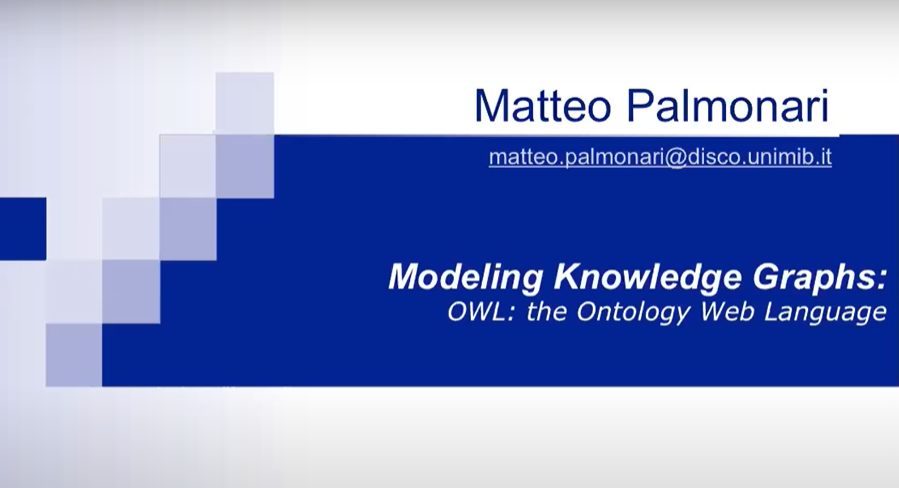


{1,2} ⊈ {1,3}

{1,3} ⊈ {1,2}

{1,2,3} maximum of the set, called T (top)

ø minimum of the set, called ⊥ (bottom)



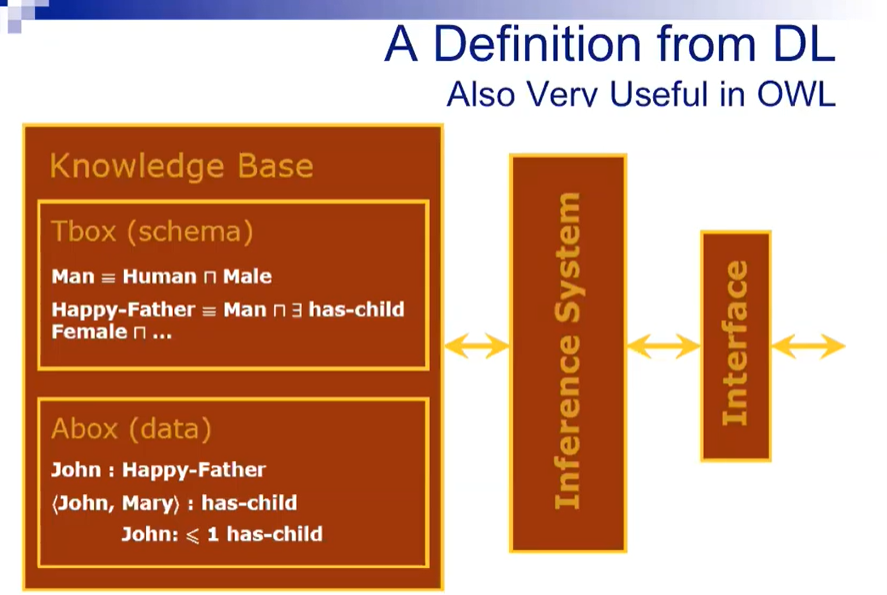
**OBW from a more practical view**

Definition of functions usually tend to consider total functions

An ontology in OWL is set of assertions

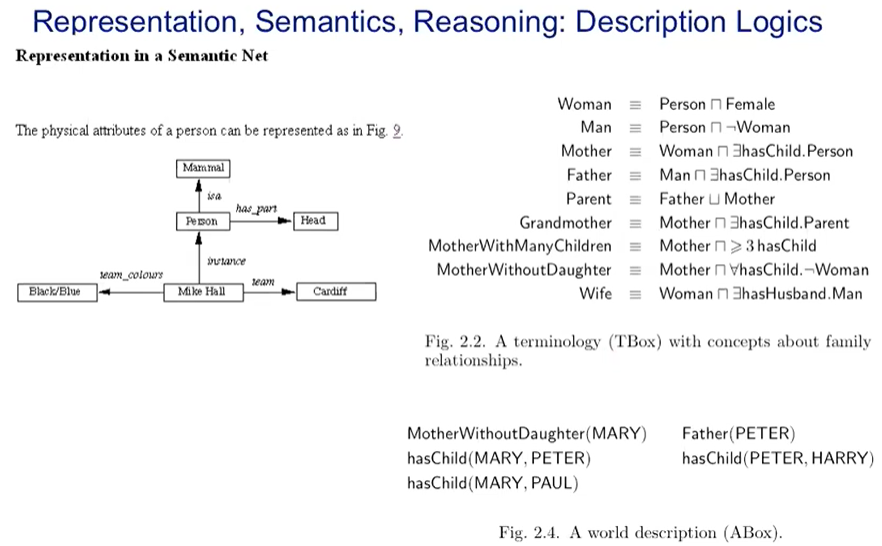
Don’t want to invent a lot, Import the good ontology

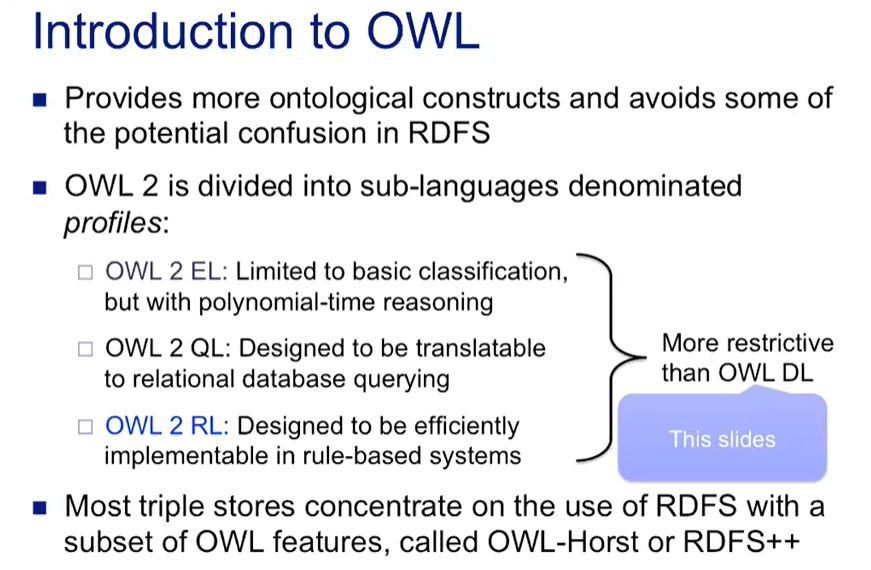
OWL: web-compliant language based on description logic



In DL you have distinction between the schema (Tbox, general axioms) and Abox contains the data. Sharp distinction

The Abox is very simple, can just say that an individual is a member of a class or that John is an happy father





RDF no predicate for equivalence, only have subclass

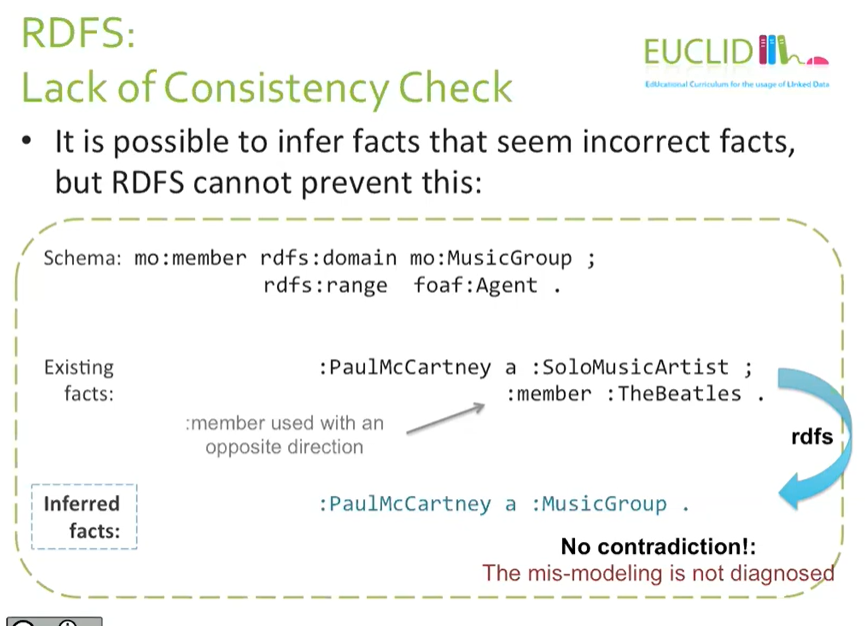


Equivalence class relation

We can say that musician is equivalent to music artist

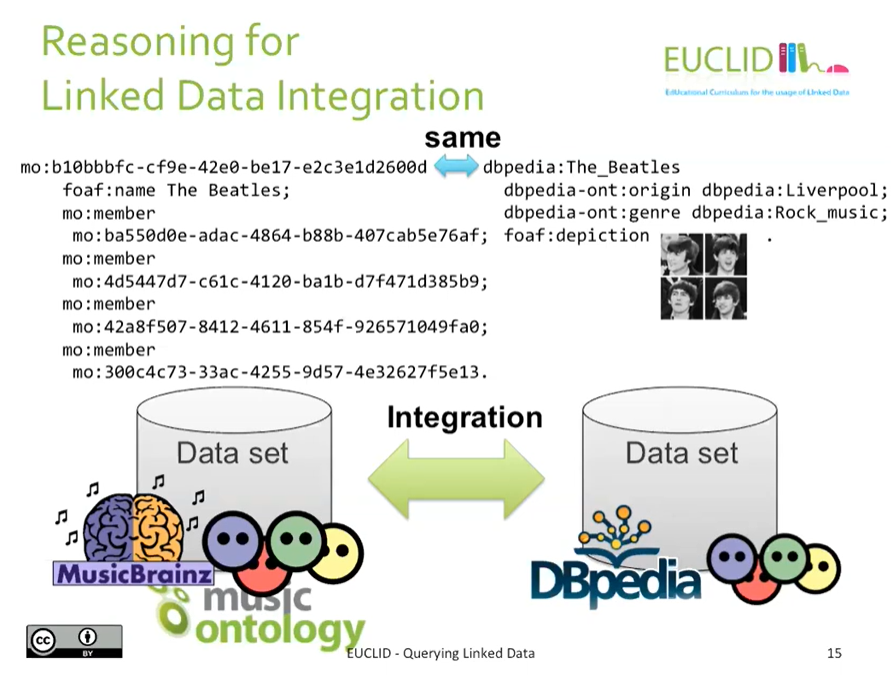
Equivalence relationship is:

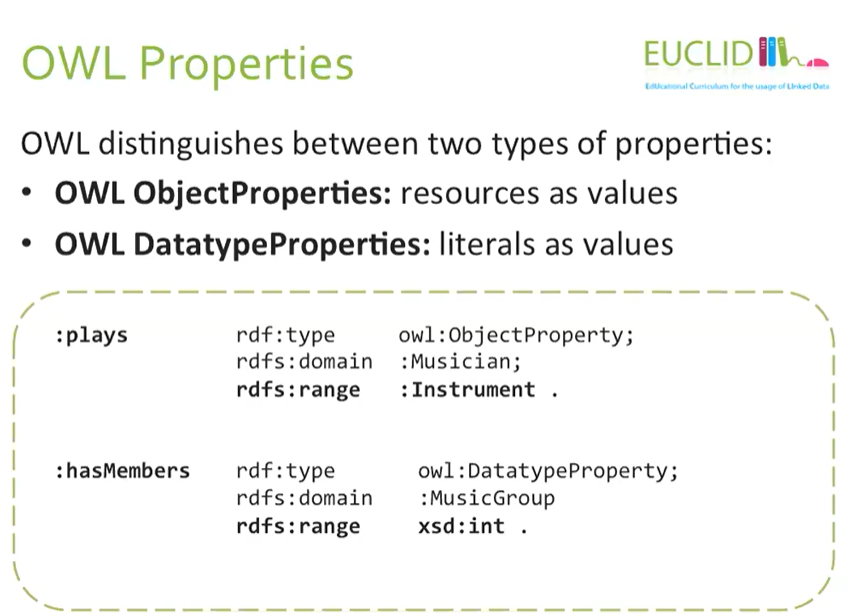
* reflexive
* transitive
* symmetric



With OWL can detect inconsistency







Object properties: properties that we use in such a way that the object is a resource

datatype properties: properties whose objects are literals

