Graduate Program in Production Engineering and Systems

(PPGEPS)

Important Terms in an Ontology (Part 2/6)

Dr. Yongxin Liao

Course Outlines

- Previously on IOE
- Important Terms in an Ontology (2/6)
  - Relationships
    - Properties
    - Property Assertions
    - Object Property
      - 7 Object Property Characteristics
      - 6 Object Property Descriptions
    - Data Property
      - 1 Data Property Characteristics
      - 5 Data Property Descriptions
- Protégé Practices

PUCPR

Previously on the IOE

- Axioms
  - An Axiom is a statement that says what is true in the domain.
  - An ontology is essentially a collection of Axioms (pieces of knowledge).
- Concepts
  - Concepts in an ontology are abstractions and representations of ideas, persons, processes, places, issues, locations, etc.
  - Taxonomy: The Classification of Concepts

PUCPR

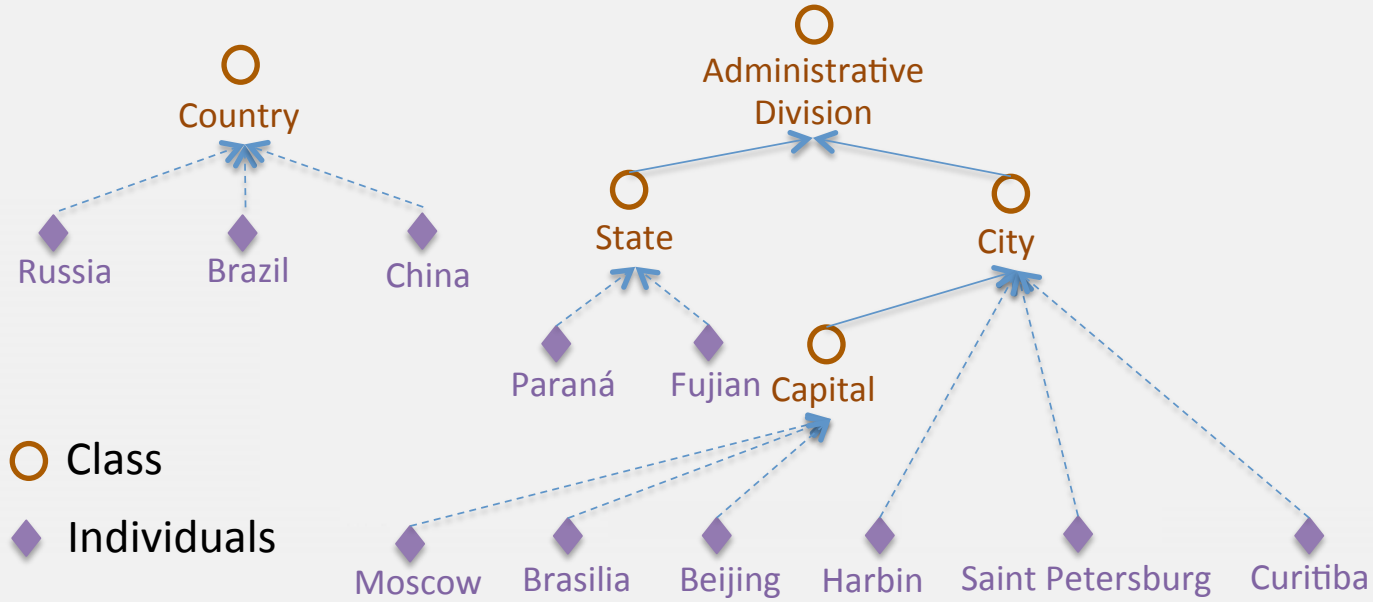
Previously on the IOE

- Individuals
  - Each individual simplifies and abstracts one actual object in a domain of interest.
  - The most specific concepts that **can not be** subdivided anymore in a domain of interest.

PUCPR

Previously on the IOE

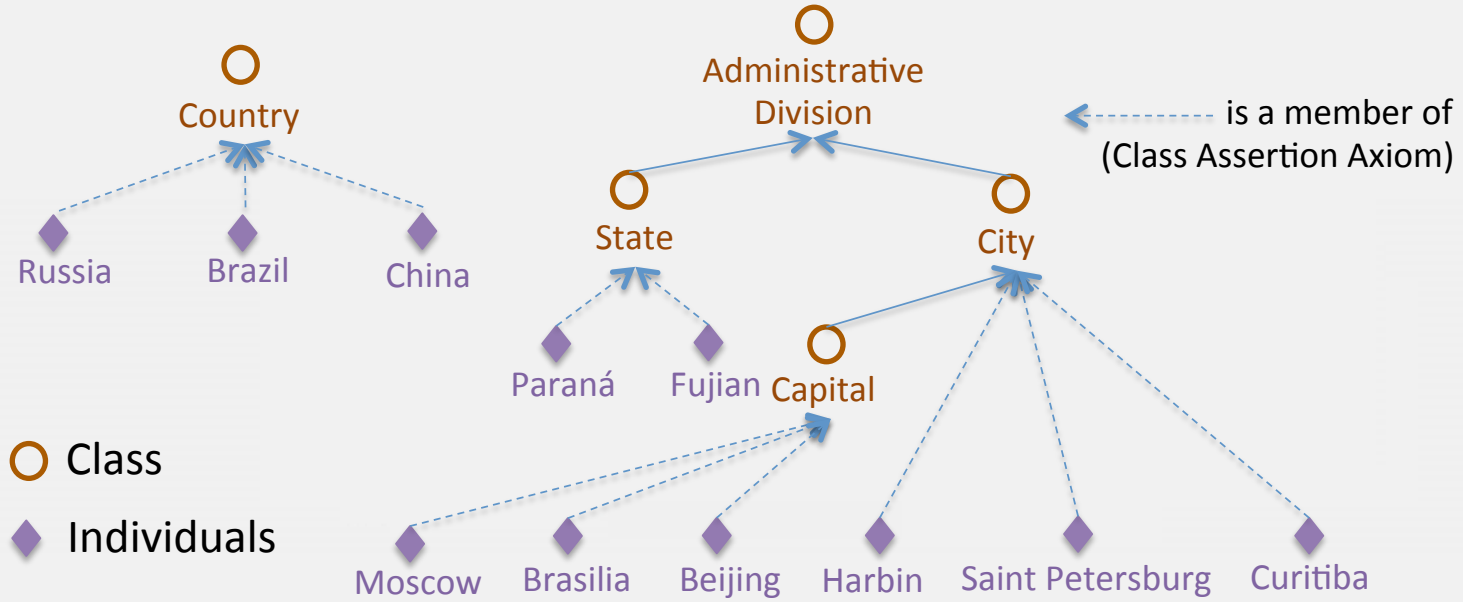
- Classes
  - Each class simplifies and abstracts a set of actual objects that share some common properties in a domain of interest.
  - The concepts that **can be** subdivided into more specific concepts in a domain of interest.



PUCPR

Previously on the IOE

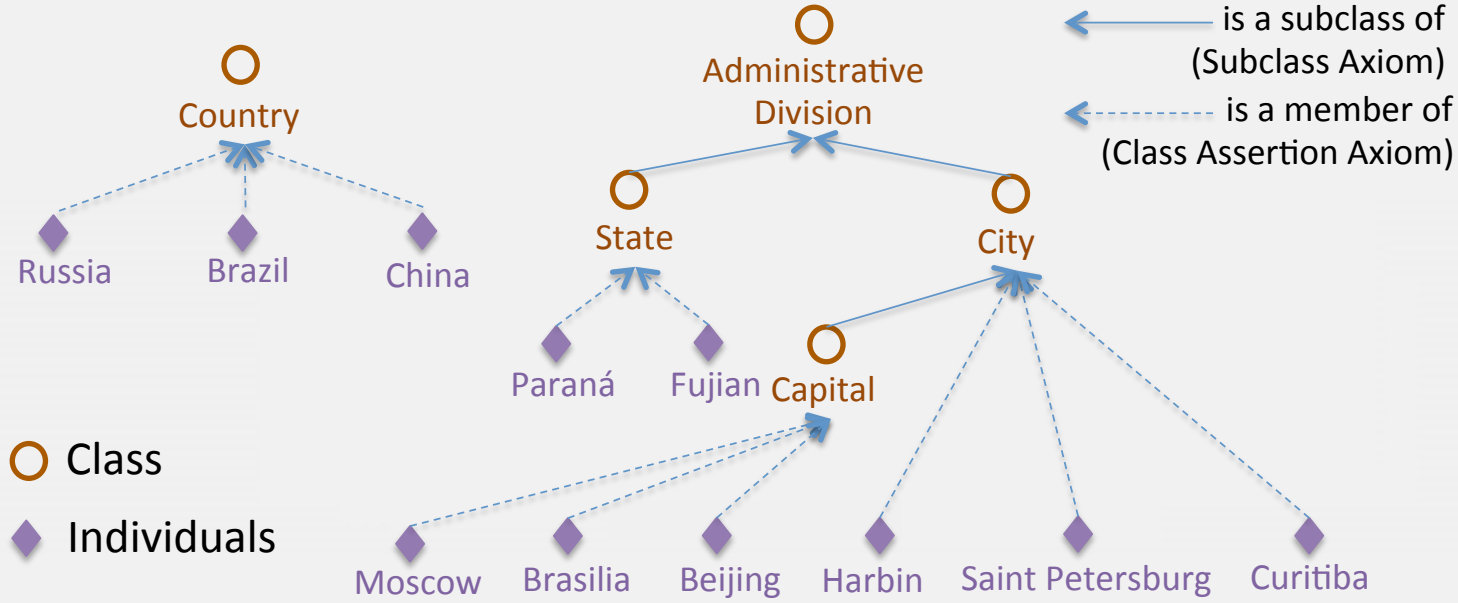
- Relationships
  - Class Assertion Axioms
    - Relationships between an individual and a class that it belongs
    - The statement of an individual is a member of a class



PUCPR

Previously on the IOE

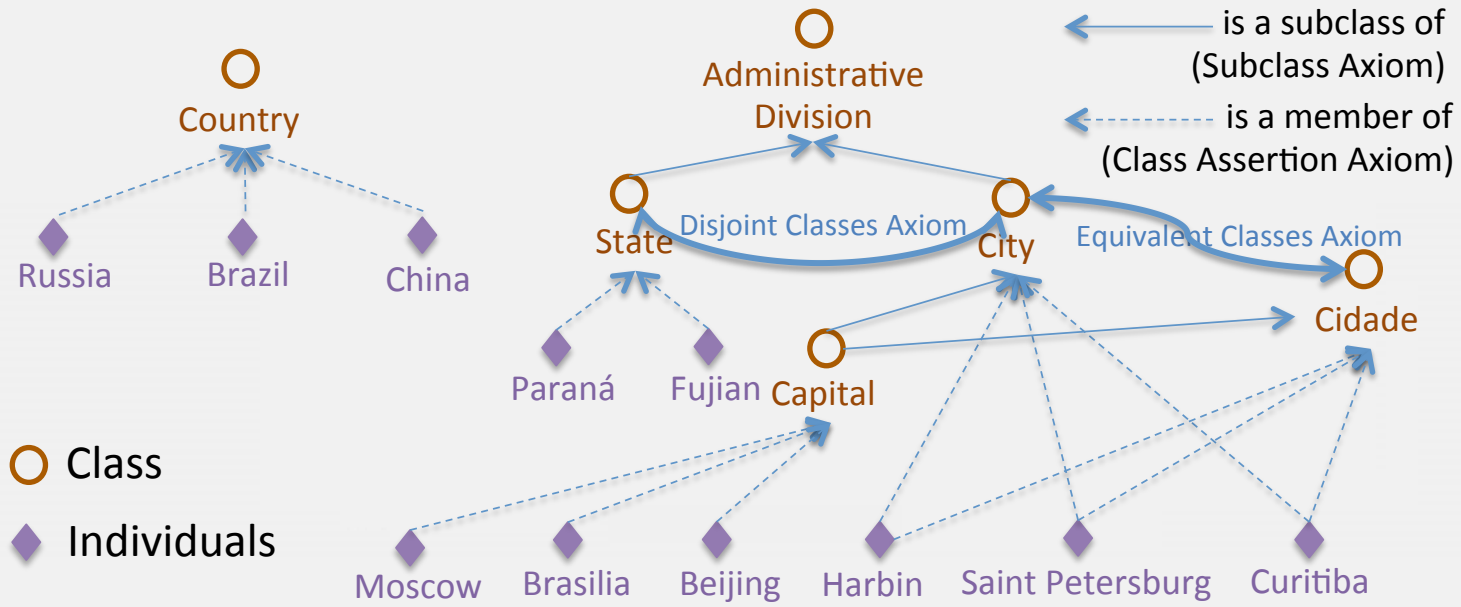
- Relationships
  - Subclass Axioms
    - Relationships between a class and its subclasses
    - A class contains *a set of objects* and its subclass contains a subset of those objects



PUCPR

Previously on the IOE

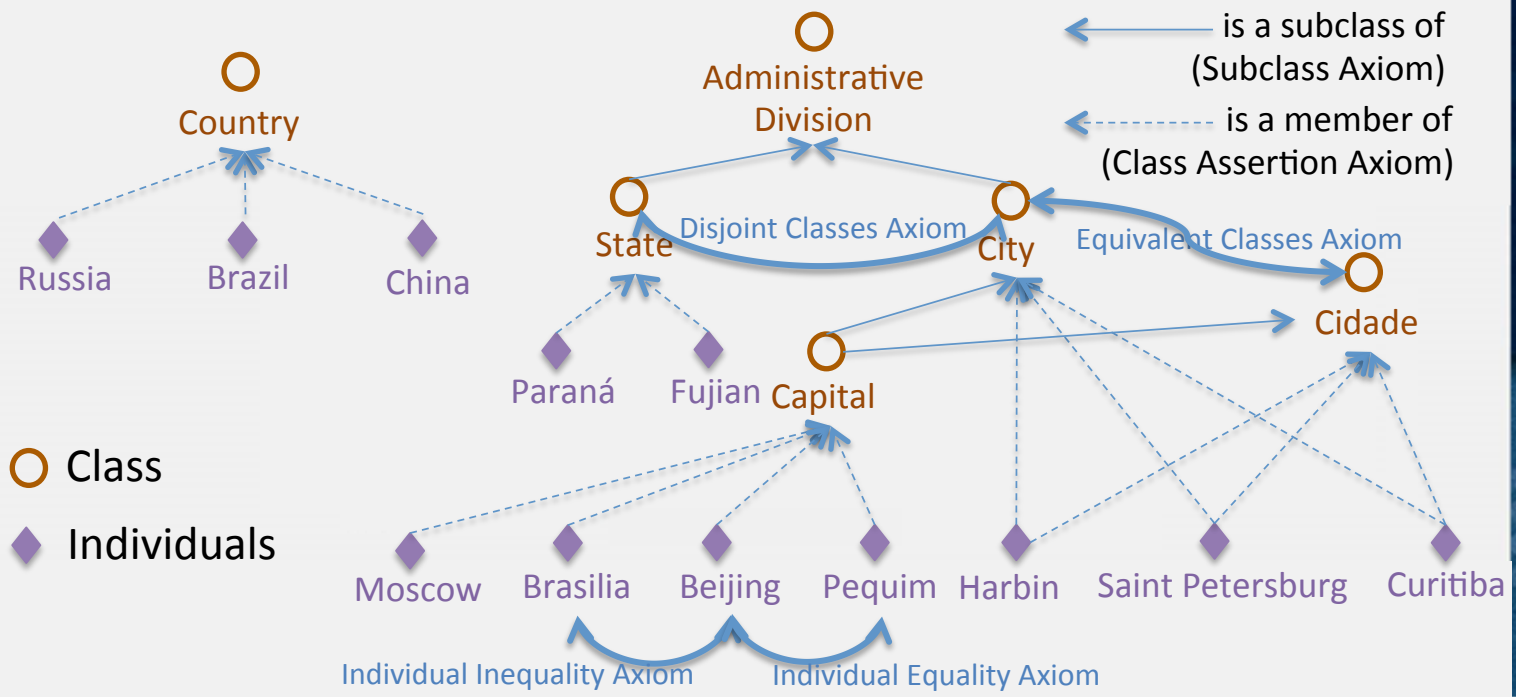
- Relationships
  - Disjoint Classes Axioms
    - Two or more classes have no element in common.
  - Equivalent Classes Axioms
    - Two or more classes are equal to each other.



PUCPR

Previously on the IOE

- Relationships
  - Individual Inequality Axioms
    - Two or more individuals are different from each other.
  - Individual Equality Axioms
    - Two or more individuals are equal to each other



Important Terms in an Ontology

- Several Important Terms
  - Axioms
  - Concepts (Individuals and Classes)
  - Relationships (Class Assertions, Subclasses Disjoint/Equivalent Classes, Individual Equality/Inequality
  - **Properties, Property Assertions, Property Characteristics, and Property Descriptions)**
  - Complex Class Expressions (Enumeration of Individuals, Propositional Connectives, Object Property Restrictions, Necessary and Sufficient Conditions, Data Property Restrictions)
  - Data Ranges (Data Types and Data Type Restrictions)
  - Reasoning Rules
  - Knowledge Base (T-box and A-box)
  - SPARQL Query

Part 1/6

Part 2/6

Part 3/6

Part 4/6

Part 5/6

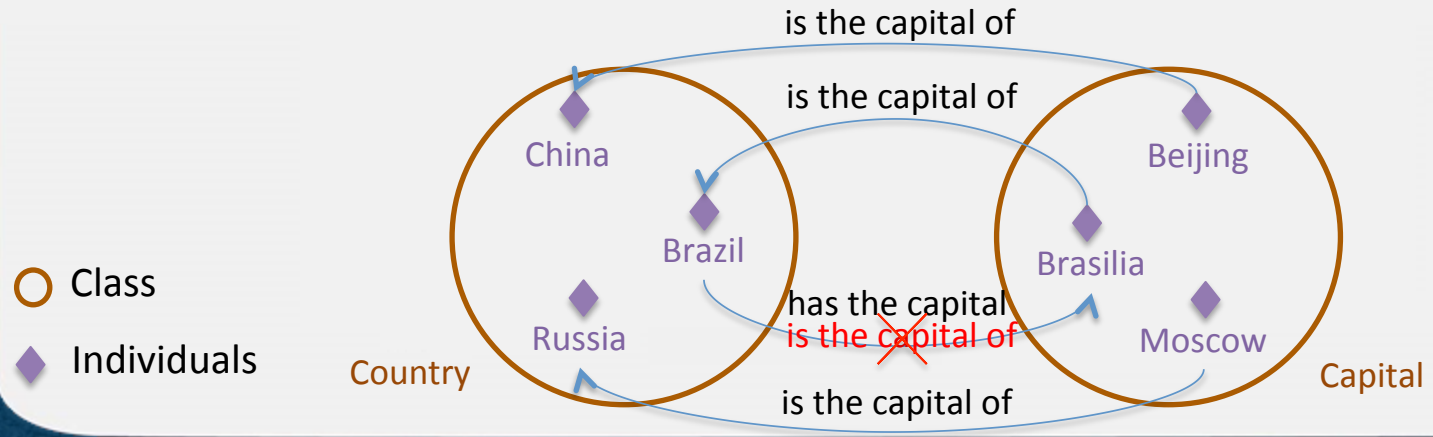
Part 6/6

PUCPR

Properties

$(x, y) \in A \times B$

- A property is a binary relation
  - A binary relation is a collection of **ordered** pairs.
    (1) Each relation is between two elements
    (2) The order of the relation contains meanings (semantics)
  - E.g. The relation between a country and a capital
    - "is the capital of" Order: from a capital to a country
    - "has the capital" Order: from a country to a capital



Properties

$(x, y) \in A \times B$

- A property is a binary relation

  Let *Country* be a set, say *Country*= {Brazil, China, Russia},

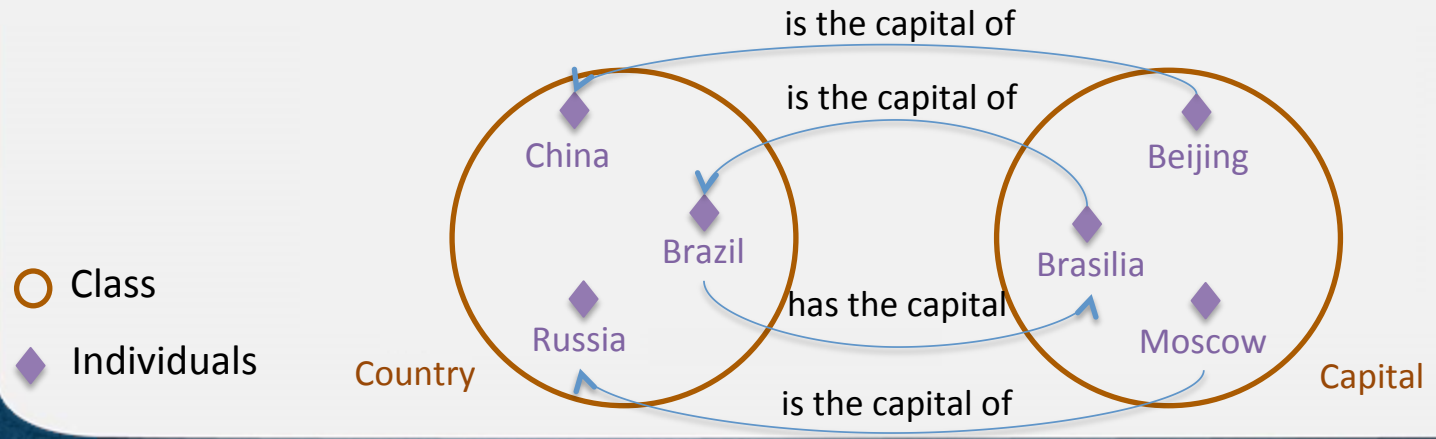
  and *Capital* be a set, say *Capital*= {Brasilia, Beijing, Moscow}.

  => A binary relation can be a collection of ordered pairs (x, y)

  for which " *x* is the capital of *y*" where x ∈ *Capital* and *y* ∈ *Country*

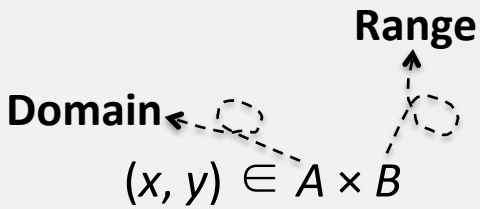
  => A binary relation can be a collection of ordered pairs (x, y)

  for which " *x* has the capital *y*" where x ∈ *Country* and *y* ∈ *Capital*



PUCPR

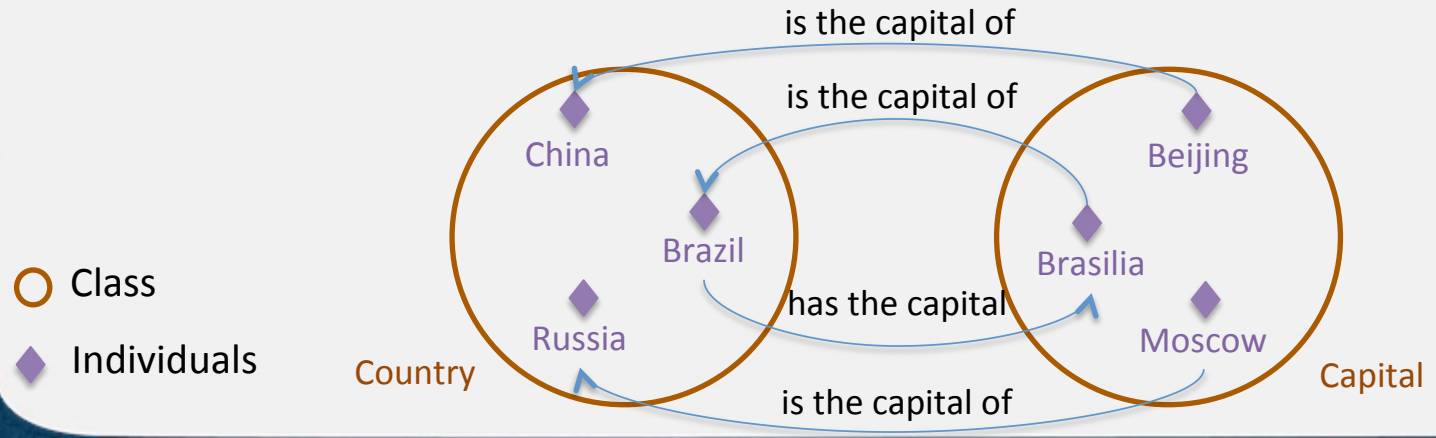
Properties



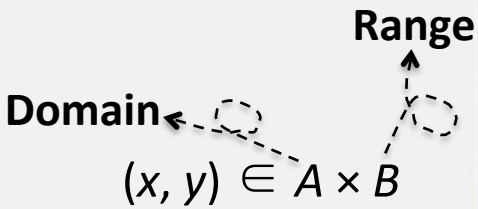
- A property is a binary relation
  - The **Domain** of a property is the set of all first coordinates of the ordered pairs.
  - The **Range** of a property is the set of all second coordinates of the ordered pairs

    E.g. The domain of "is the capital of" is the Class Capital

    The range of "is the capital of" is the Class Country



Properties

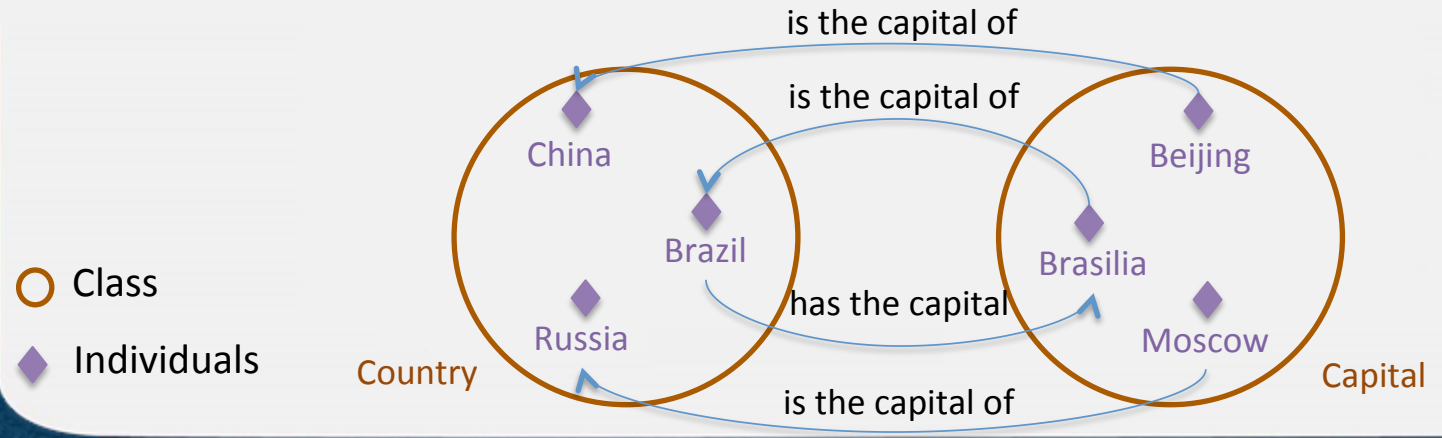


- A property is a binary relation
  - The **Domain** of a property is the set of all first coordinates of the ordered pairs.
  - The **Range** of a property is the set of all second coordinates of the ordered pairs

Question: What are the domain and range of "has the capital"?

Answers: The domain of "has the capital" is the Class Country

The range of "has the capital" is the Class Capital



Property Assertions

- Two main types of properties
  - Object Properties connect pairs of individuals
    - Object Property Assertion Axioms

      E.g. Curitiba isLocatedIn Brazil

      Brasilia isCapitalOf Brazil

  - Data Properties connect individuals with data values
    - Data Property Assertion Axioms

      E.g. Brasilia wasFoundOn "1960-04-21T00:00:00"^^xsd:dateTime

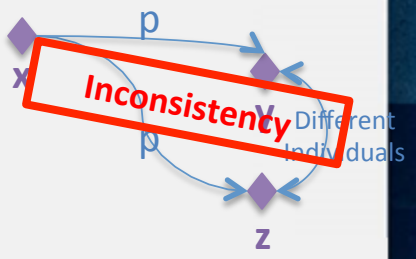
      Curitiba hasPopulation "1879355"^^xsd:integer

PUCPR

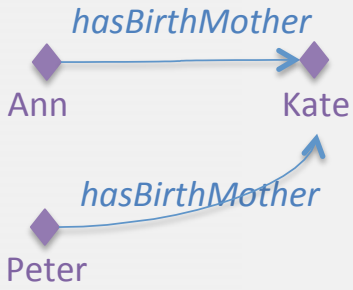
Object Properties

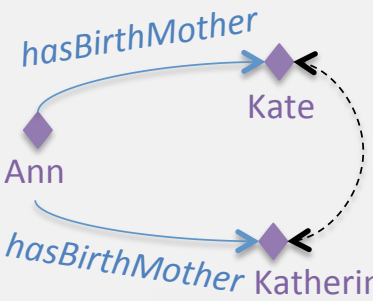
- Characteristics (1/7)
  - Functional Object Property Axioms
    - If a **functional** object property p is the connection from x to y, then for each individual x, there can be at most one distinct individual y.

Let p be a **functional** object property

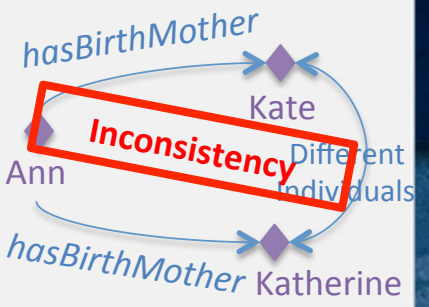


E.g. An example of functional Object properties: hasBirthMother





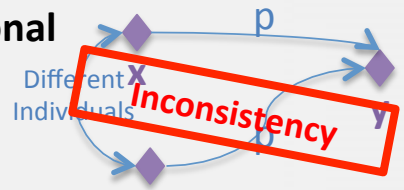
Implies Kate and Katherine are the same Individual



PUCPR

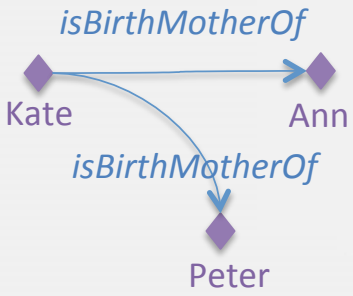
Object Properties

Let p be an **Inverse functional** object property

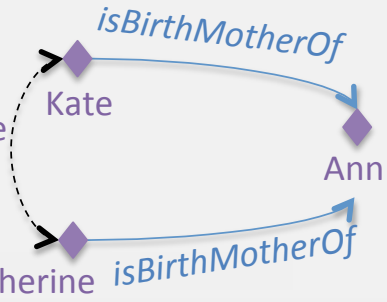


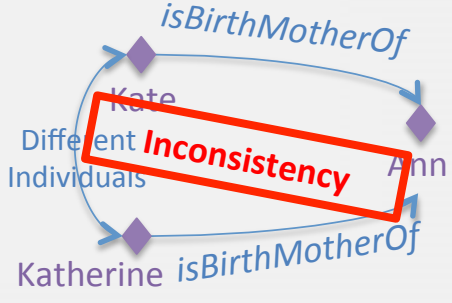
- Characteristics (2/7)
  - Inverse Functional Object Property Axioms
    - If an **inverse functional** object property p is the connection from x to y, then for each individual y, there can be at most one distinct individual x.

E.g. An example of inverse functional object properties: isBirthMotherOf



Implies Kate and Katherine are the same Individual

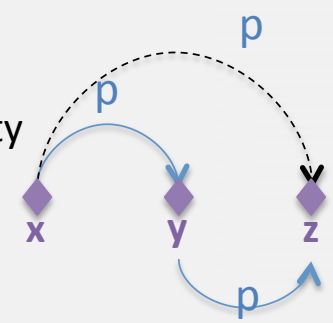




PUCPR

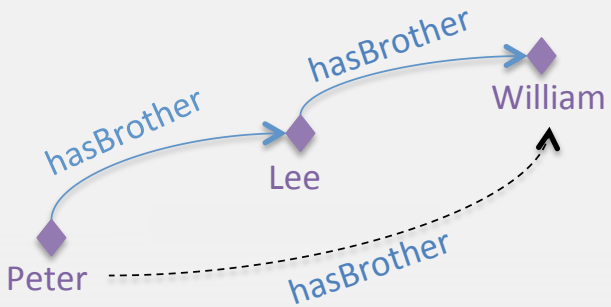
Object Properties

Let p be a **transitive** object property



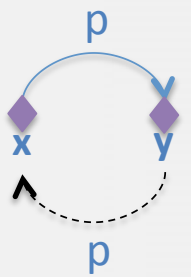
- Characteristics (3/7)
  - Transitive Object Property Axioms
    - If an Individual x is connected by a **transitive** object property p to an Individual y that is connected by p to an Individual z, then x is also connected by p to z

E.g. An example of transitive object properties: hasBrother



PUCPR

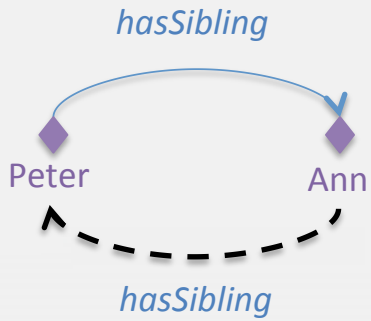
Object Properties



Let p be a **symmetric** object property

- Characteristics (4/7)
  - Symmetric Object Property Axioms
    - If an Individual x is connected by a **symmetric** object property p to an Individual y, then y is also connected by p to x.

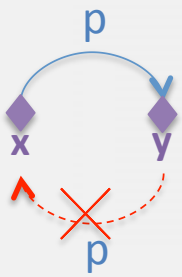
E.g. An example of symmetric object properties: hasSibling



PUCPR

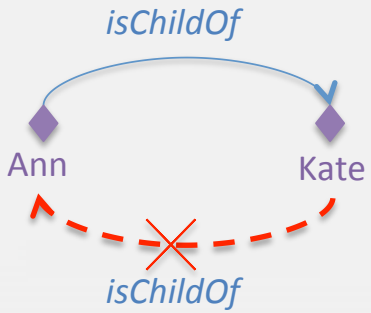
Object Properties

Let p be an **asymmetric** object property



- Characteristics (5/7)
  - Asymmetric Object Property Axioms
    - If an Individual x is connected by an **asymmetric** object property p to an Individual y, then y **cannot** be connected by p to x

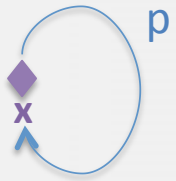
E.g. An example of asymmetric object properties: isChildOf



PUCPR

Object Properties

Let p be a **reflexive** object property



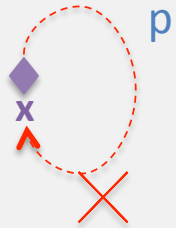
- Characteristics (6/7)
  - Reflexive Object Property Axioms
    - An object property is said to be **reflexive** when this object property must connect an individual to it self
    - Each individual in an ontology must connect by the reflexive object properties to itself

E.g. An example of reflexive object properties: knows

PUCPR

Object Properties

Let p be an **irreflexive** property



- Characteristics (7/7)
  - Irreflexive Object Property Axioms
    - An object property is said to be **irreflexive** when this object property must not connect an individual to it self
    - No Individual in an ontology is connected by the irreflexive object properties to itself

E.g. An example of irreflexive object properties: isBirthMotherOf

PUCPR

Object Properties

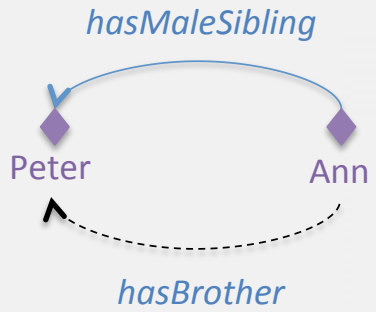
- Descriptions (1/6)
  - Equivalent Object Properties Axioms
    - It states that all the equivalent object properties are semantically equal to each other
    - They can be replaced by each other without affecting the meaning of the ontology

E.g. Individuals: Peter, Ann;

Object Properties: hasBorther, hasMaleSibling;

hasBorther is equivalent to hasMaleSibling;

Ann hasMaleSibling Peter.



We can infer that Ann hasBorther Peter

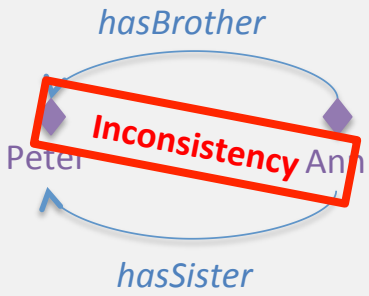
Object Properties

- Descriptions (2/6)
  - Disjoint Object Properties Axioms
    - It states that all the disjoint object properties are pairwise disjoint
    - No Individual x can be connected to an Individual y by two disjoint object properties at the same time.

E.g. Individuals: Peter, Ann;

Object Properties: hasBorther, hasSister;

hasBorther is disjoint with hasSister.

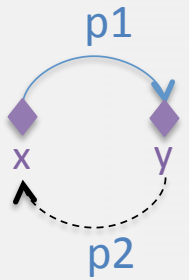


If Ann hasBorther Peter and Ann hasSister Peter both exist, then the ontology would become inconsistent.

Object Properties

- Descriptions (3/6)
  - Inverse Object Property Axioms
    - Let the object property p1 and p2 be two inverse object properties. If an individual x is connected by p1 to an individual y, then y is also connected by p2 to x.

Let p1 and p2 be two inverse object property



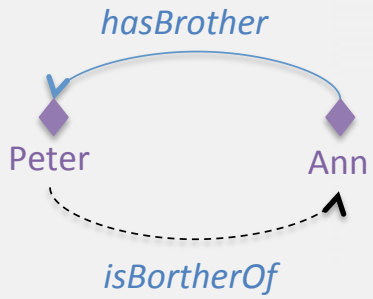
E.g. Individuals: Peter, Ann;

Object Properties: hasBorther, isBrotherOf;

hasBorther and isBrotherOf are inverse properties

Ann hasBorther Peter.

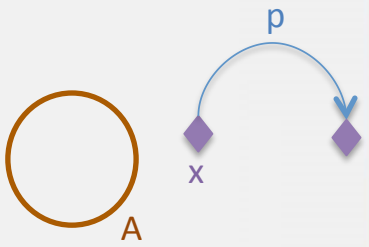
We can infer that Peter isBrotherOf Ann



PUCPR

Object Properties

Let the set A be specified as
the domain of an object property p



- Descriptions (4/6)
  - Object Property Domain Axioms
    - Let the set A be specified as the domain of the object property p. If an individual x is connected by p with some individual, then x must be a member of A.

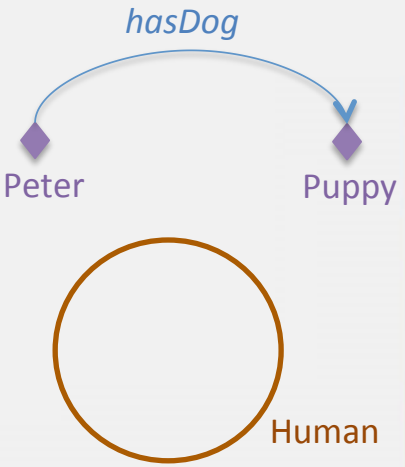
E.g. Class: Human;

Individuals: Peter, Puppy;

Object Property: hasDog;

Human is specified as the domain of hasDog;

Peter hasDog Puppy.

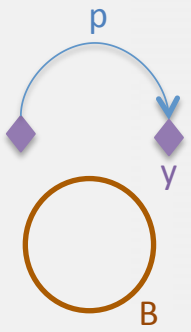


We can infer that Peter is a member of Human.

Object Properties

- Descriptions (5/6)
  - Object Property Range Axioms
    - Let the set B be specified as the range of the object property p. If some individual is connected by p with an individual y, then y must be a member of B.

Let the set B be specified as the range of an object property p



E.g. Class: Dog;

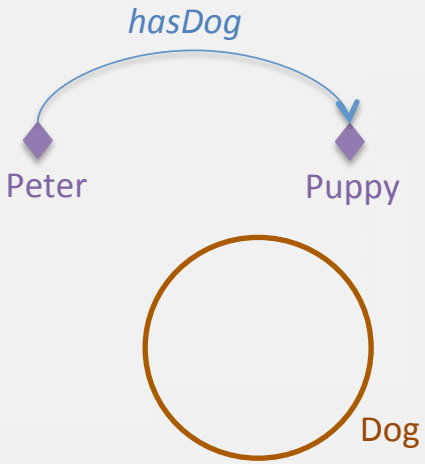
Individuals: Peter, Puppy;

Object Property: hasDog;

Dog is specified as the range of hasDog;

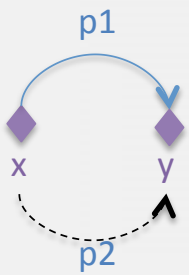
Peter hasDog Puppy.

We can infer that Puppy is a member of Dog



Object Properties

Let the p1 be a sub object property of p2



- Descriptions (6/6)
  - Sub Object Property Axioms
    - Let the object property p1 be a sub property of the object property p2. If an Individual x is connected by p1 to and an Individual y, then x is also connected by p2 to y
    - They cannot be replaced by each other

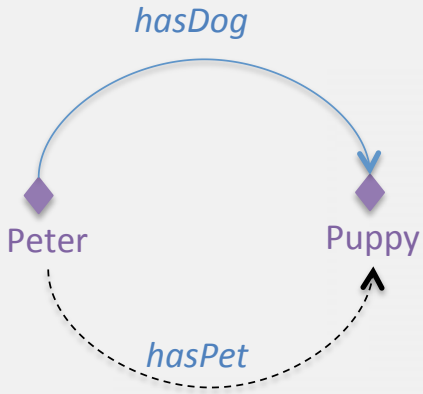
E.g. Individuals: Petter, Puppy;

Object Properties: hasDog, hasPet;

hasDog is a sub property of hasPet;

Peter hasDog Puppy.

We can infer that Peter hasPet Puppy



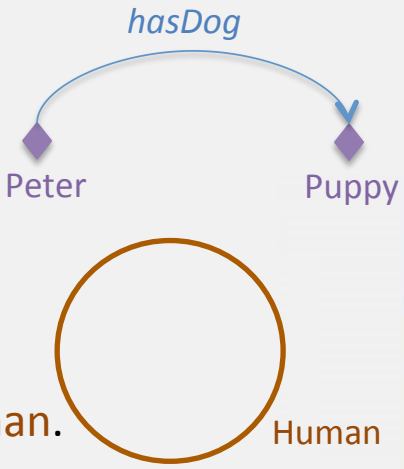
PUCPR

Reasoners (1/2)

- Classification of Individuals
  - Based on the descriptions (conditions) of individuals, properties, and classes to check whether or not an individual is a member of existing classes.

E.g. Class: Human;
Individuals: Peter, Puppy;
Object Property: hasDog;
Human is specified as the domain of hasDog;
Peter hasDog Puppy.

A reasoner can infer that Peter is a member of Human.



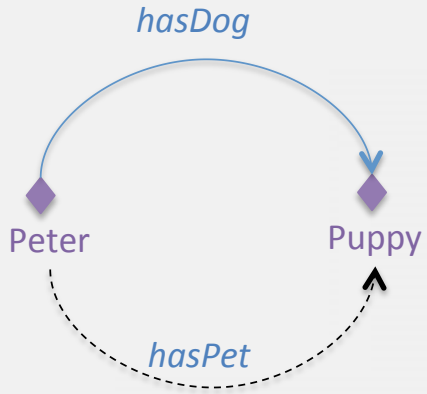
PUCPR

Reasoners (2/2)

- Discover New Property Assertions
  - Based on the descriptions (conditions) of individuals, properties, and classes to check whether or not this individual is connected to another individual (or itself) through an existing property.

E.g. Individuals: Petter, Puppy;
Object Properties: hasDog, hasPet;
hasDog is a sub property of hasPet;
Peter hasDog Puppy.

A reasoner can infer that Peter hasPet Puppy



PUCPR

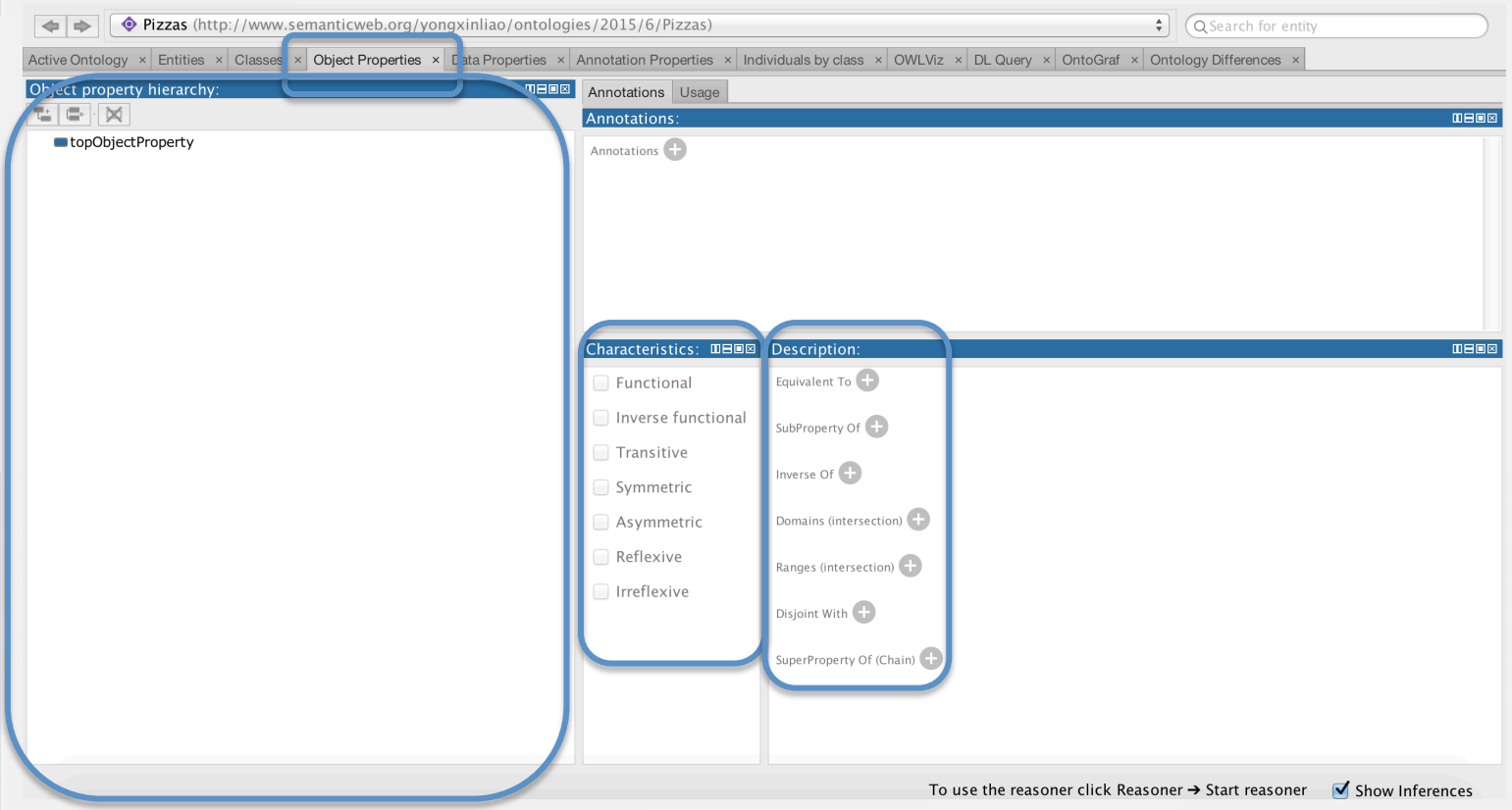
Protégé Practices

- The “Object Properties” Tab
  - Object Properties
  - Object Property Characteristics
  - Object Property Description
  - Object Property Assertion and Inference
- The “Data Properties” Tab
  - Data Properties
  - Data Property Characteristics
  - Data Property Description
  - Data Property Assertion and Inference

PUCPR

Protégé Practices

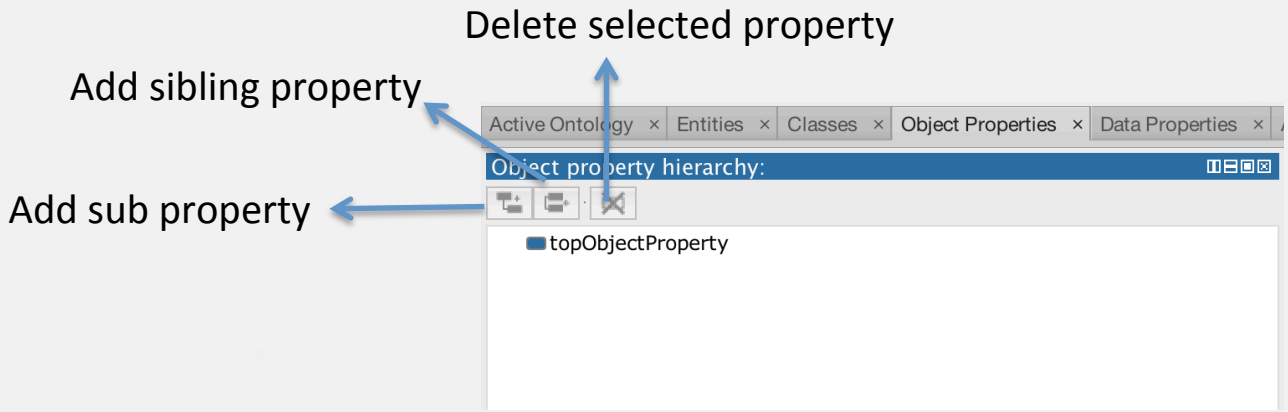
- The “Object Properties” Tab



PUCPR

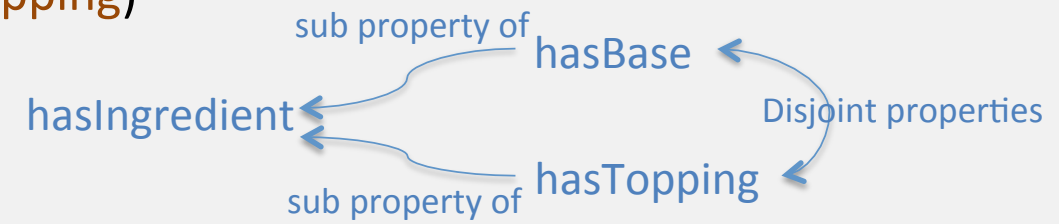
Protégé Practices

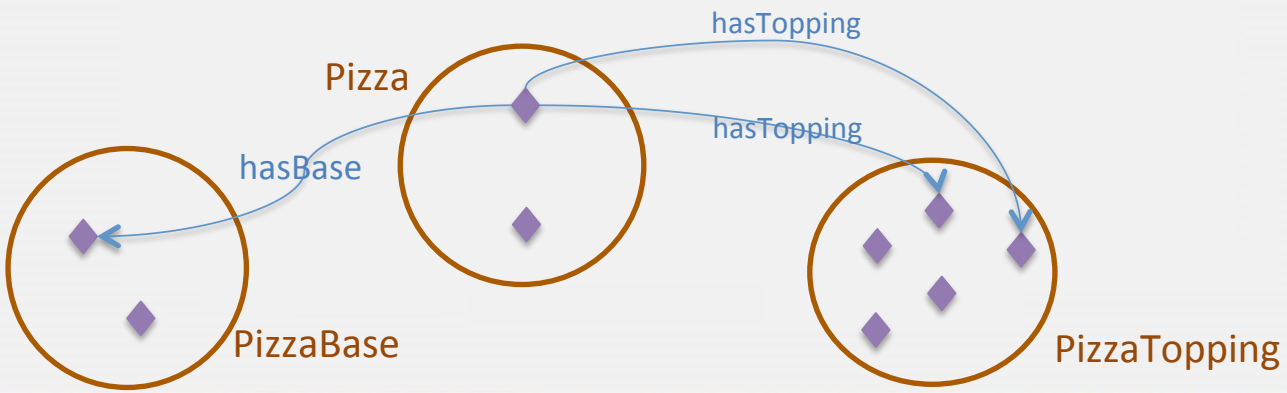
- The TopObjectProperty
  - It is a pre-defined property as the most general object property in an ontology.
  - It is the root property of all object properties
  - It represents the property that can connect all possible pairs of individuals



Protégé Practices

- Object Properties
  - A pizza (an individual in Class: Pizza) is composed of one pizza base (an individual in Class: PizzaBase) and one or more pizza toppings (one ore more individuals in Class: PizzaTopping)

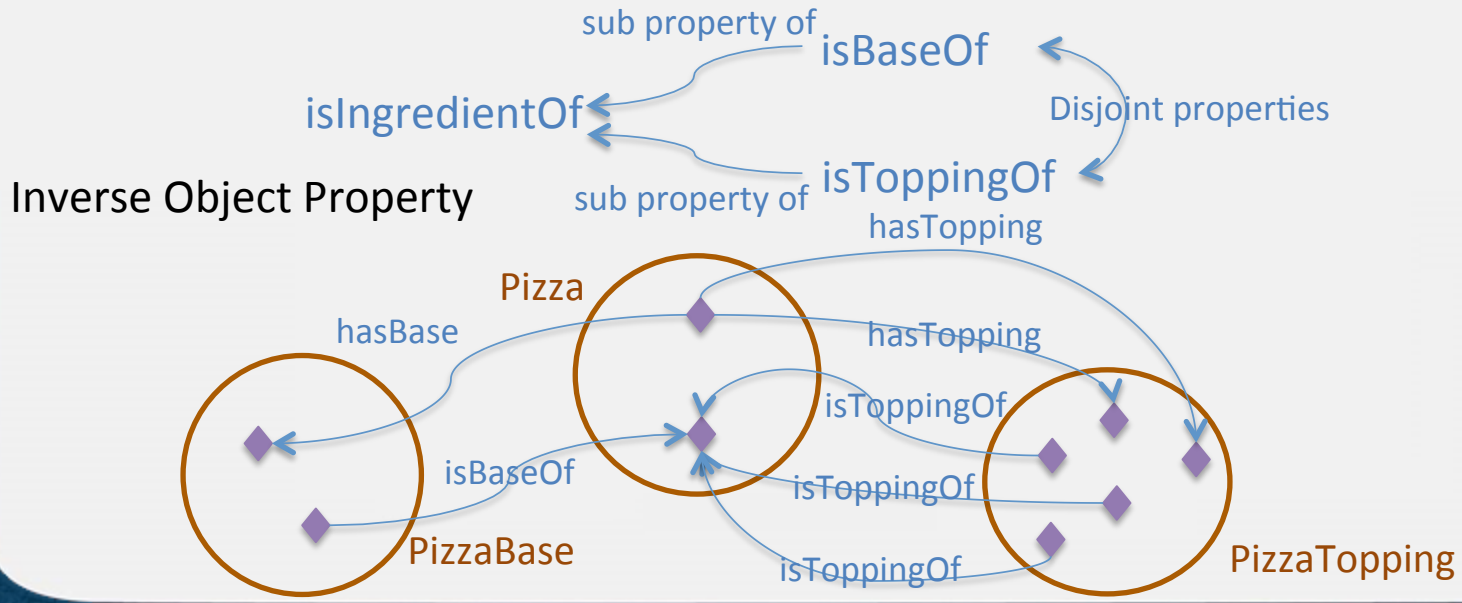




PUCPR

Protégé Practices

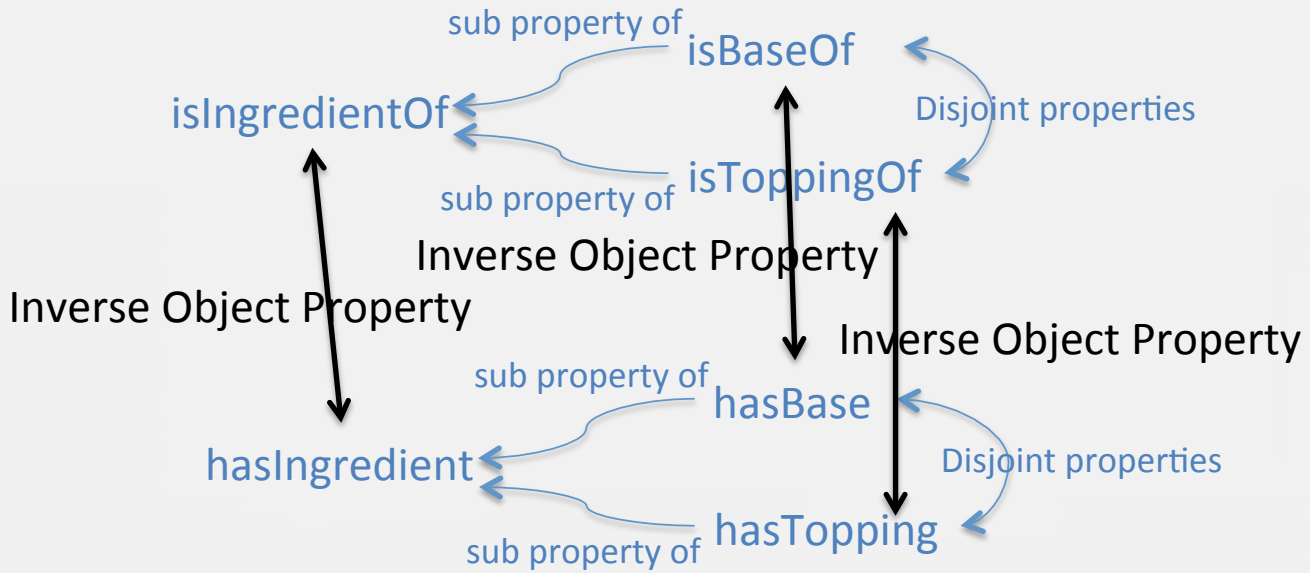
- Inverse Object Properties
  - A pizza (an individual in Class: Pizza) is composed of one pizza base (an individual in Class: PizzaBase) and one or more pizza toppings (one ore more individuals in Class: PizzaTopping)



PUCPR

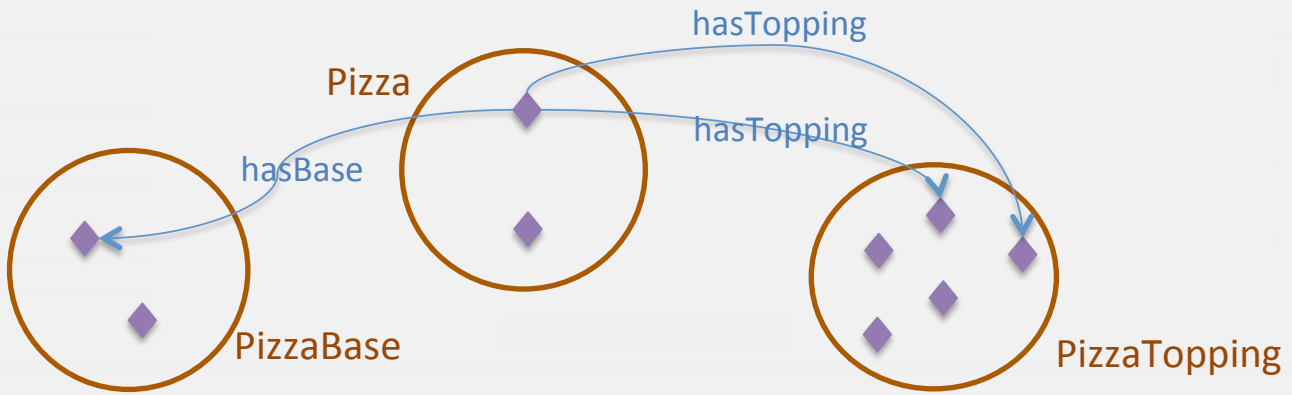
Protégé Practices

- Inverse Object Properties
  - A pizza (an individual in Class: Pizza) is composed of one pizza base (an individual in Class: PizzaBase) and one or more pizza toppings (one ore more individuals in Class: PizzaTopping)



Protégé Practices

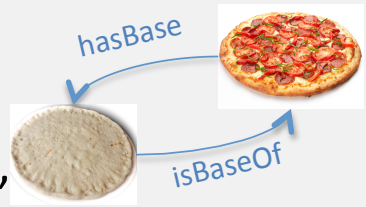
- Object Property Domains and Ranges
  - The domain of hasTopping is Class Pizza
  - The range of hasTopping is Class PizzaTopping
  - The domain of hasBase is Class Pizza
  - The range of hasBase is Class PizzaBase

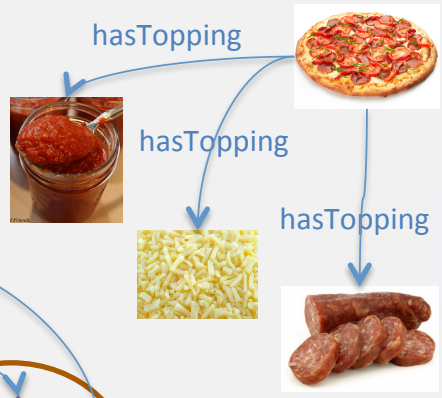


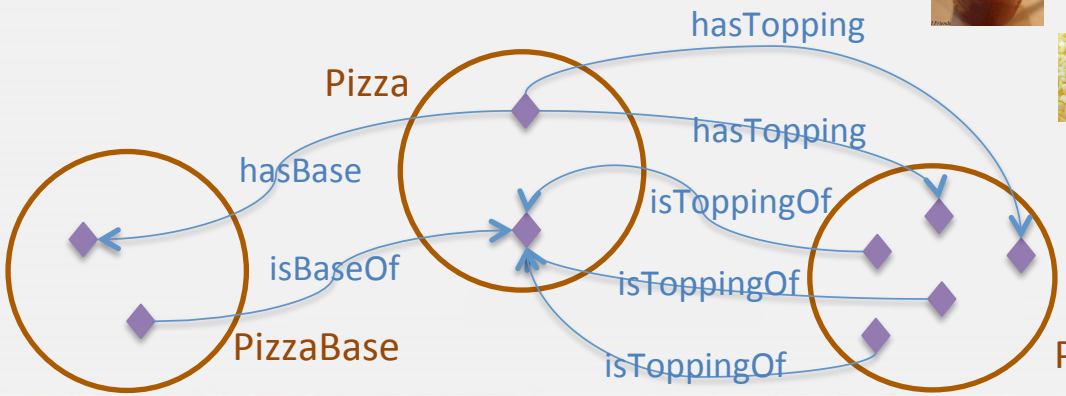
PUCPR

Protégé Practices

- Property Characteristics
  - hasIngredient => “Transitive”
  - isIngredientOf => “Transitive”
  - hasBase => “Functional” and “Inverse functional”
  - isBaseOf => “Functional” and “Inverse Functional”
  - hasTopping => “Inverse functional”
  - isToppingOf =>“Functional”







Protégé Practices

- Object Property Assertions

Pieces of Knowledge:

(1) hasBase has the following characteristics:
- Functional Object Property
- Inverse functional Object Property

hasBase has the following descriptions:
- Disjoint with hasTopping
- Domain: Pizza. Range: PizzaBase
- Inverse of isBaseOf
- Sub property of hasIngredient

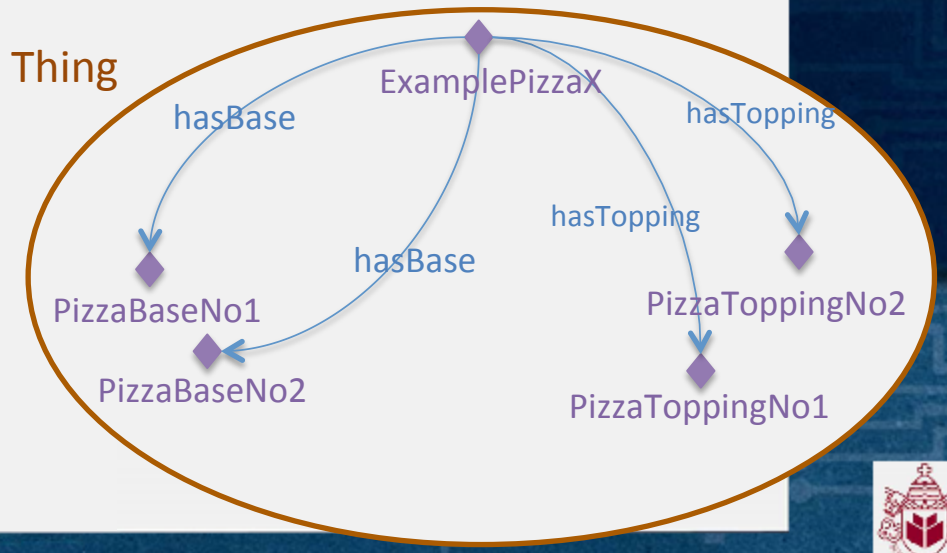
(2) hasTopping has the following characteristics:
- Inverse functional Object Property

hasTopping has the following descriptions:
- Disjoint with hasBase
- Domain: Pizza. Range: PizzaTopping
- Inverse of isToppingOf
- Sub property of hasIngredient

(3) ExamplePizzaX, PizzaBaseNo1, PizzaBaseNo2, PizzaToppingNo1, PizzaToppingNo2 are members of Class Thing

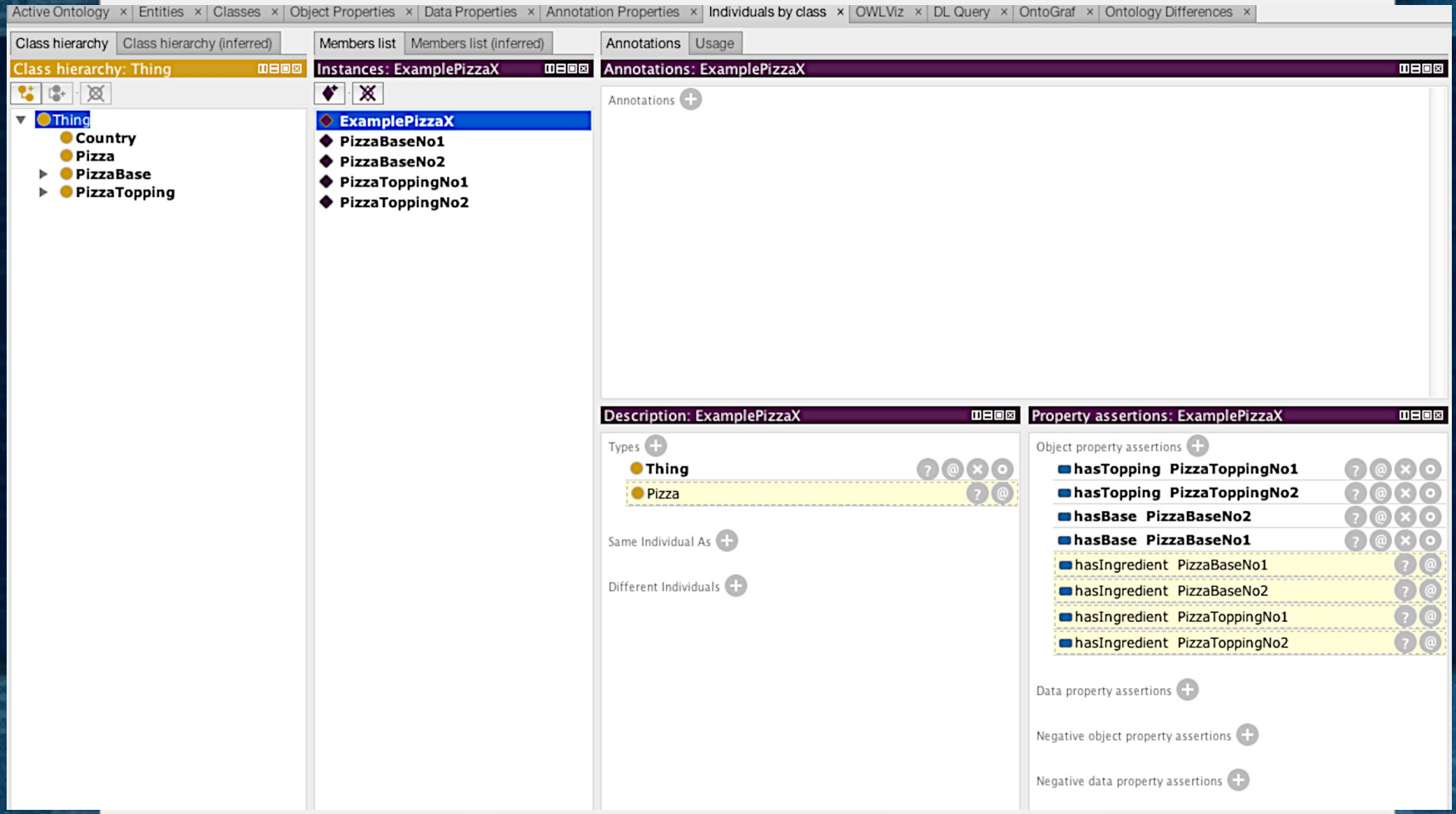
(4) ExamplePizzaX hasBase PizzaBaseNo1
ExamplePizzaX hasBase PizzaBaseNo2.

(5) ExamplePizzaX hasTopping PizzaToppingNo1
ExamplePizzaX hasTopping PizzaToppingNo2



PUCPR

Protégé Practices



PUCPR

Literals

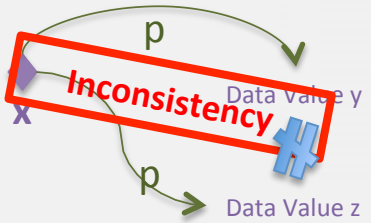
- Data Values
  - Data values can be understood as individuals denoting values through their literal expressions.
    - They can not be members of Classes
    - They are members of Data Types (integer, string, dateTime)
  - Each data value consists of a lexical form
    - A string + a data type
    - E.g. “22”^^xsd:integer “Peter Young”^^xsd:string
      “1997-01-01T00:00:00”^^xsd:dateTime
  - Two data value are structurally equivalent if and only if both the lexical form are equivalent.
    - No need to specify the equivalent among data values
    - E.g. “22”^^xsd:integer = “22”^^xsd:integer
      “22”^^xsd:integer ≠ “22”^^xsd:string
      “Peter Young”^^xsd:string ≠ “Peter_Young”^^xsd:string

PUCPR

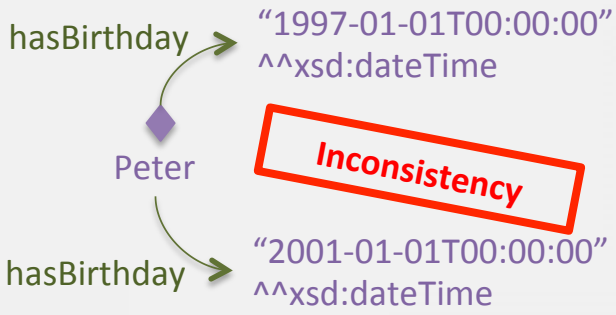
Relationships in an Ontology

- Properties
  - Data Property Characteristics (1/1)
    - Functional Data Property Axioms
      - For each individual x there can be at most one distinct data value y such that x is connected by a **functional** data property p with y.

Let p be a **functional** data property

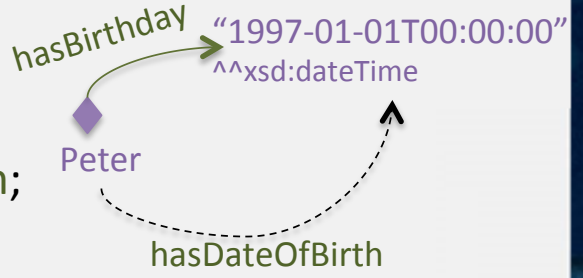


E.g. An example of functional data properties: hasBirthday



Relationships in an Ontology

- Properties
  - Data Property Descriptions (1/5)
    - Equivalent Data Properties Axiom
      - It states that all the equivalent data properties are semantically equal to each other
      - They can be replaced by each other without affecting the meaning of the ontology



E.g. Individual: Peter;

Data Properties: hasBirthday, hasDateOfBirth;

hasBirthday is equivalent to hasDateOfBirth;

Peter hasBirthday "1997-01-01T00:00:00"^^xsd:dateTime.

We can infer that Peter hasDateOfBirth "1997-01-01T00:00:00"^^xsd:dateTime

PUCPR

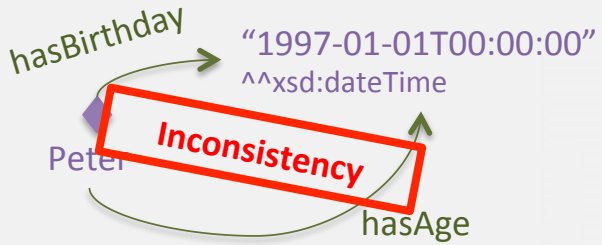
Relationships in an Ontology

- Properties
  - Data Property Descriptions (2/5)
    - Disjoint Data Properties Axiom
      - It states that all the disjoint data properties are pairwise disjoint
      - No Individual x can be connected to a data value y by two disjoint data properties at the same time

E.g. Individual: Peter;

Data Properties: hasBirthday, hasAge;

hasBirthday is disjoint with hasAge.



If Peter hasBirthday "1997-01-01T00:00:00"^^xsd:dateTime, and Peter hasAge "1997-01-01T00:00:00"^^xsd:dateTime both exist,

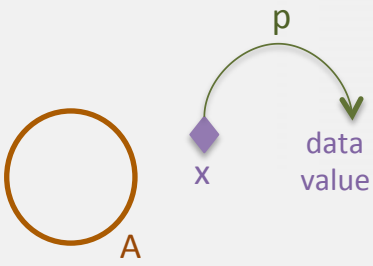
then the ontology would become inconsistent.

PUCPR

Relationships in an Ontology

- Properties
  - Data Property Descriptions (3/5)
    - Data Property Domain Axiom
      - Let the set A be specified as the domain of the data property p. If an individual x is connected by p with some data value, then the x is a member of A.

Let the set A be specified as the domain of a data property p



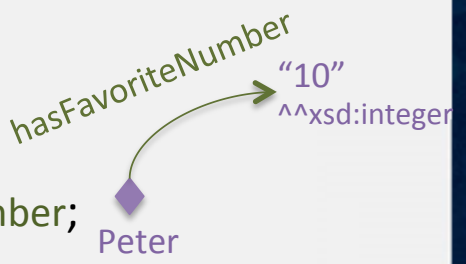
E.g. Class: Human;

Individual: Peter;

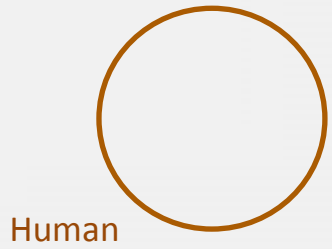
Data Property: hasFavoriteNumber;

Human is specified as the domain of hasFavoriteNumber;

Peter hasFavoriteNumber "10"^^xsd:integer.



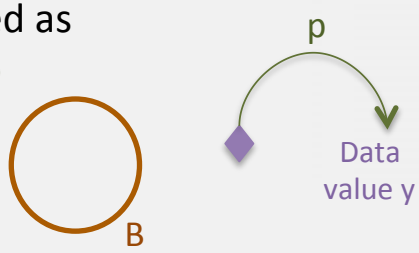
We can infer that Peter is a member of Human.



Relationships in an Ontology

- Properties
  - Data Property Descriptions (4/5)
    - Data Property Range Axiom
      - Let the data range B be specified as the range of the data property p. If some individual is connected by p with a data value y, the y must be in the data range B.

Let the data rang B be specified as the range of a data property p

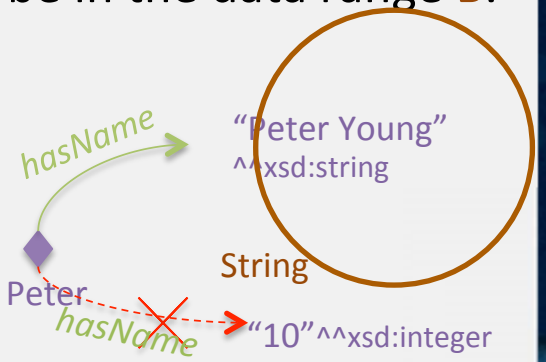


E.g. Data Range: String;

Individual: Peter;

Object Property: hasName;

String is specified as the range of hasName.



If Peter hasName "Peter Young"^^xsd:string, then the range condition is satisfied

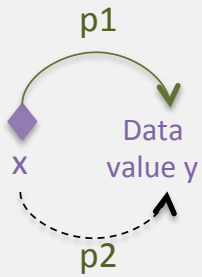
If Peter hasName "10"^^xsd:integer, then the ontology would become inconsistent

PUCPR

Relationships in an Ontology

- Properties
  - Data Property Descriptions (5/5)
    - Sub Data Property Axiom
      - Let the data property p1 be a sub property of the data property p2. If an Individual x is connected by p1 to and a data value y, then x is also connected by p2 to y
      - They cannot be replaced by each other

Let the p1 be a sub data property of p2

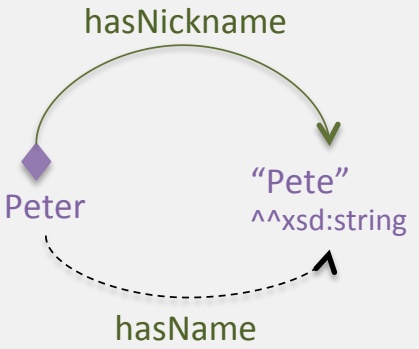


E.g. Individual: Petter;

Object Properties: hasNickname, hasName;

hasNickname is a sub property of hasName;

Peter hasNickname "Pete"^^xsd:string.



We can infer that Peter hasName "Pete"^^xsd:string

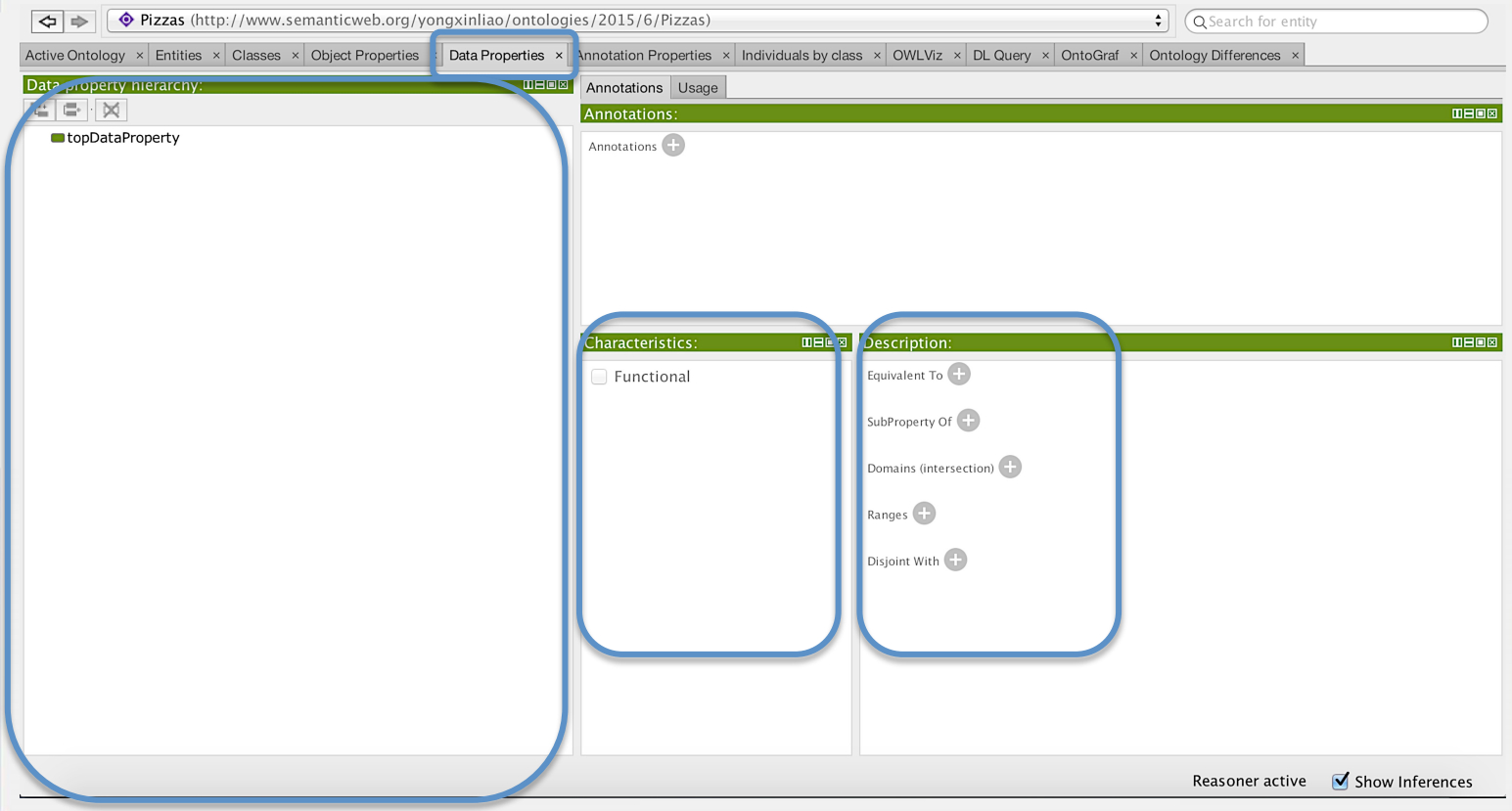
Protégé Practices

- The "Object Properties" Tab
  - Object Properties
  - Object Property Characteristics
  - Object Property Description
  - Object Property Assertion and Inference
- The "Data Properties" Tab
  - Data Properties
  - Data Property Characteristics
  - Data Property Description
  - Data Property Assertion and Inference

PUCPR

Protégé Practices

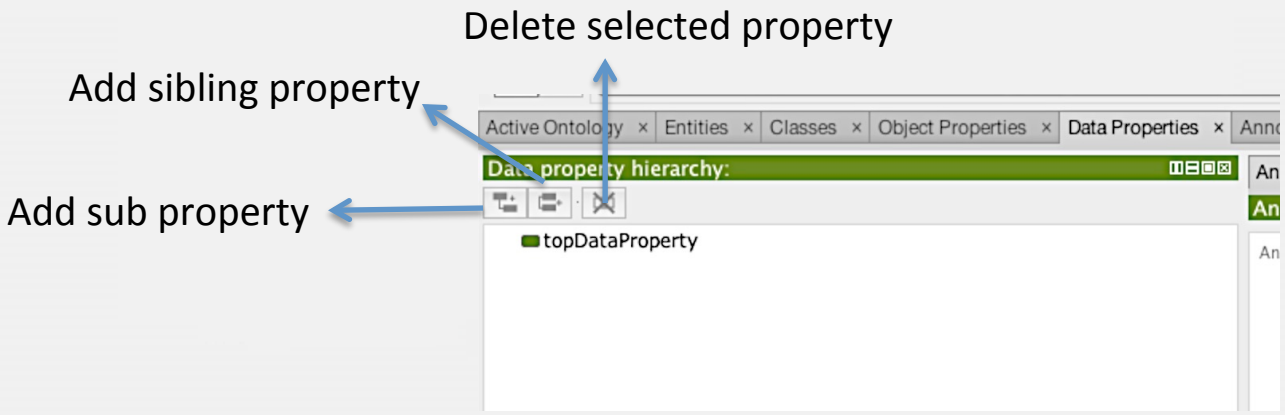
- The “Data Properties” Tab



PUCPR

Protégé Practices

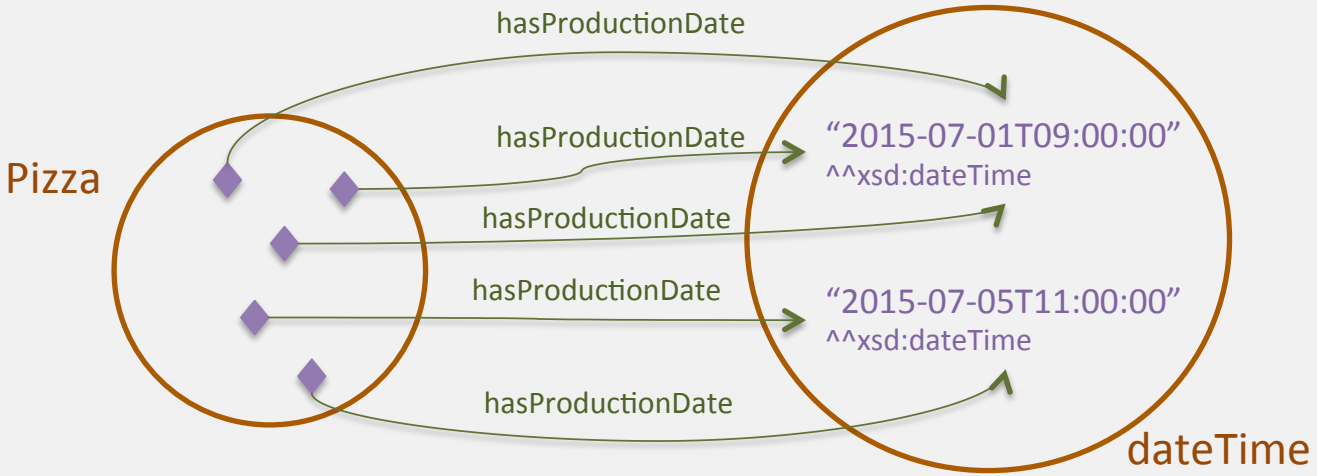
- The TopDataProperty
  - It is a pre-defined property as the most general data property in an ontology.
  - It is the root property of all data properties
  - It represents the property that can connect all possible pairs individuals with all data values



PUCPR

Protégé Practices

- Data Properties
  - Each pizza (an individual in Class: Pizza) has only one production date.
  - The hasProductionDate is a Functional Data Property.
  - The Range of hasProductionDate is dateTime
    - It is a built-in data type, which represents instants of time.
    - It is specified in the form of "YYYY-MM-DDThh:mm:ss"



Protégé Practices

- Object Property Assertions

Pieces of Knowledge:

(1) hasProductionDate has the following characteristics:

- Functional Data Property

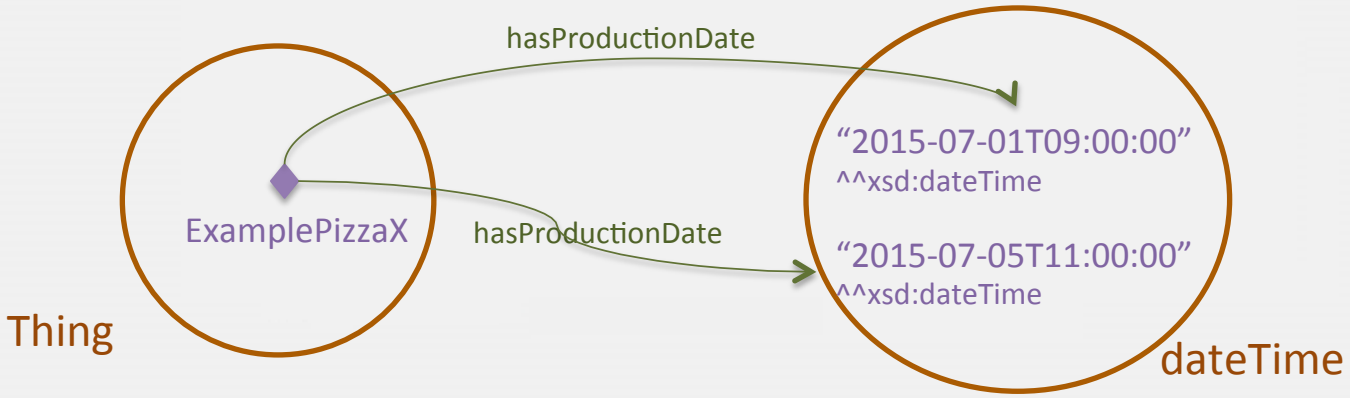
hasProductionDate has the following descriptions:

- Range: dateTime

(2) ExamplePizzaX is a member of Class Thing

(3) ExamplePizzaX hasProductionDate "2015-07-01T09:00:00"^^xsd:dateTime

ExamplePizzaX hasProductionDate "2015-07-05T09:00:00"^^xsd:dateTime



Protégé Practices

- Object Property Assertions

Pieces of Knowledge:

(1) hasProductionDate has the following characteristics:

- Functional Data Property

hasProductionDate has the following descriptions:

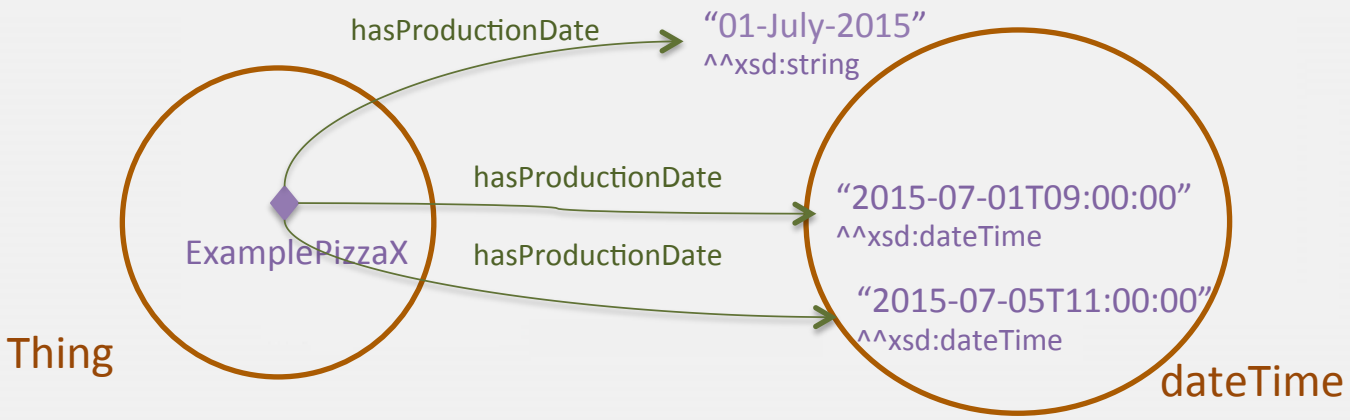
- Range: dateTime

(2) ExamplePizzaX is a member of Class Thing

(3) ExamplePizzaX hasProductionDate "2015-07-01T09:00:00"^^xsd:dateTime

ExamplePizzaX hasProductionDate "2015-07-05T09:00:00"^^xsd:dateTime

ExamplePizzaX hasProductionDate "01-July-2015"^^xsd:string



PUCPR

Thank you
for your attention!

Any Questions?

PUCPR