Building OWL Ontologies with Protege

CS 431 – April 9, 2007

Carl Lagoze – Cornell University

Protégé and RACER – tools for building, manipulating and reasoning over ontologies

- Protégé - http://protege.stanford.edu/
  - Use the 3.2 version (or the 3.3 beta)
  - Multiple plug-ins are available
- Protégé OWL plug-in
  - http://protege.stanford.edu/plugins/owl/
  - Packaged in the full Protégé download
- RacerPro
  - Description Logic based reasoning engine
  - Server-based
  - Integrates with Protégé-OWL
  - Apply for the academic license
  - http://www.racer-systems.com/

A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools
Edition 1.0

Matthew Horridge[1],
Holger Knublauch[2], Alan Rector[1], Robert Stevens[1], Chris Wroe[1]

[1] The University Of Manchester
[2] Stanford University

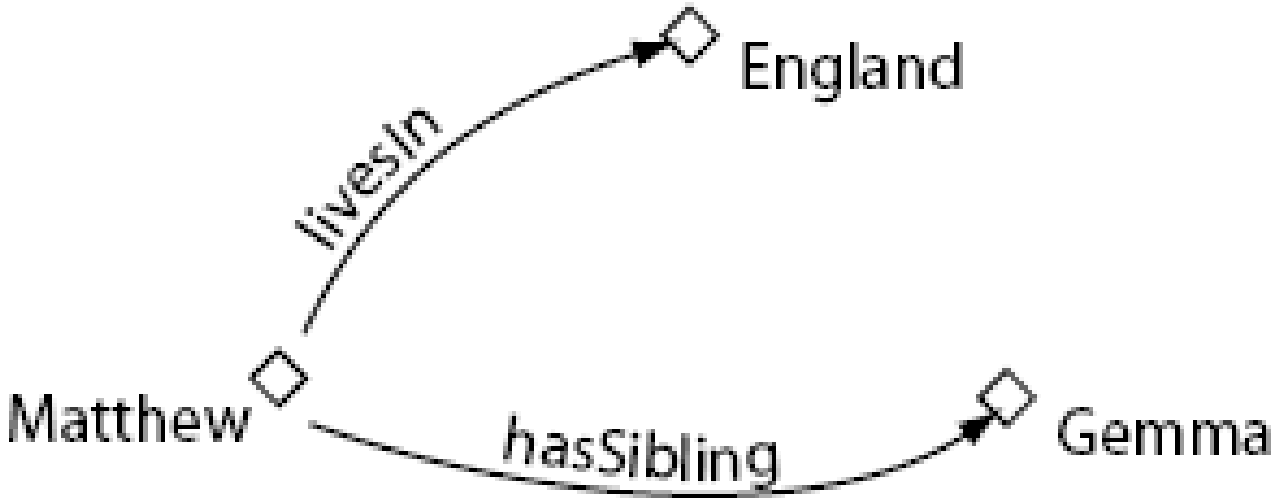
August 27, 2004

http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf

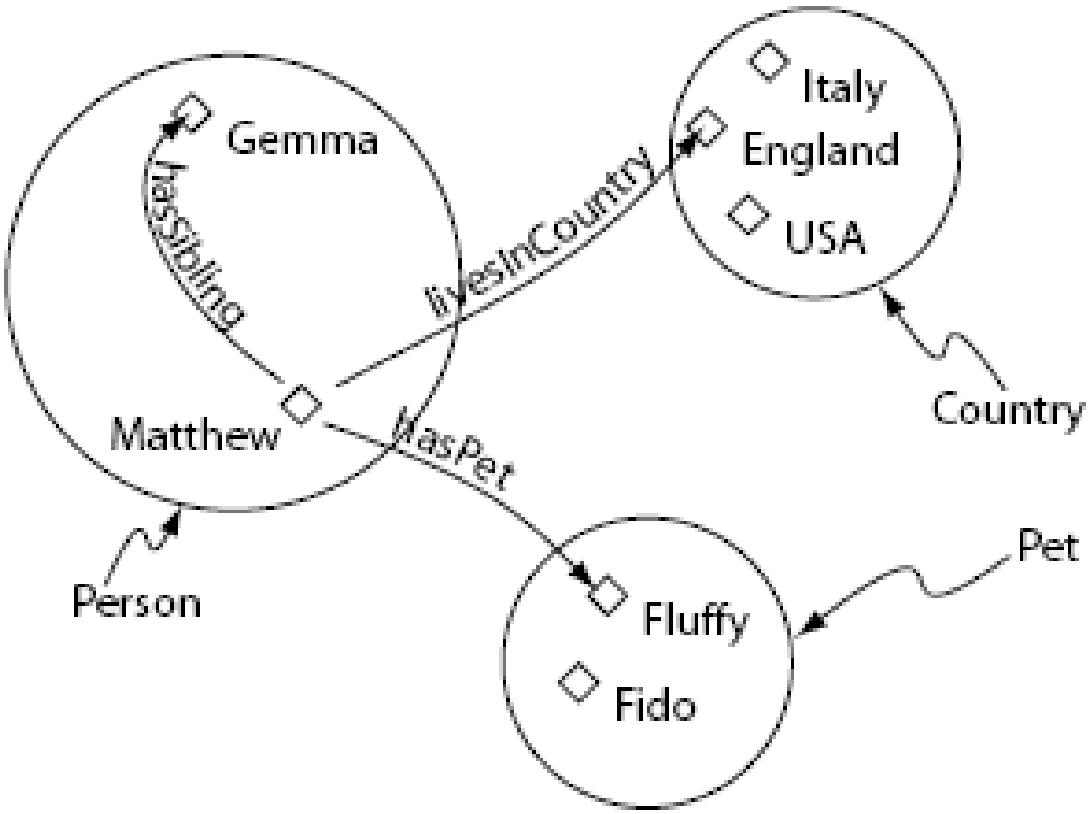
Components of OWL Ontologies: Individuals

Unique name assumption: Two individuals are not equivalent (even if ids are different) unless explicitly stated so (open world)

Components of OWL Ontologies: Properties among Individuals

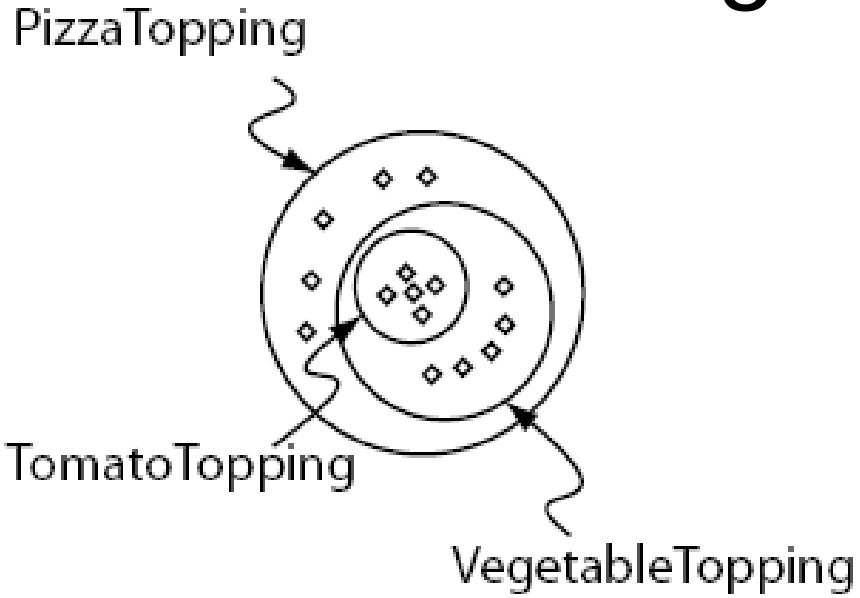


Components of OWL Ontologies: Classes, Properties, and Individuals



- All individuals must be in a class
- Define sets of individuals

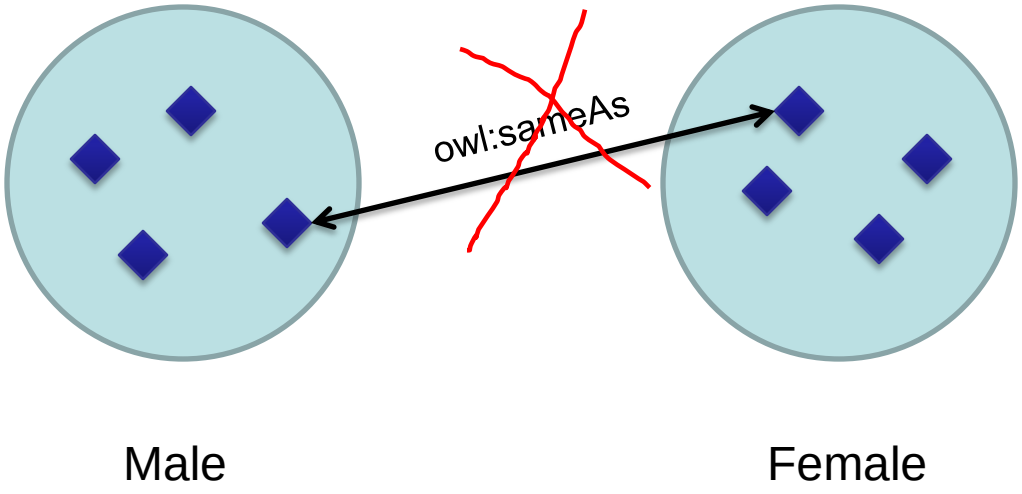
Relationships among classes: Sub-Classing



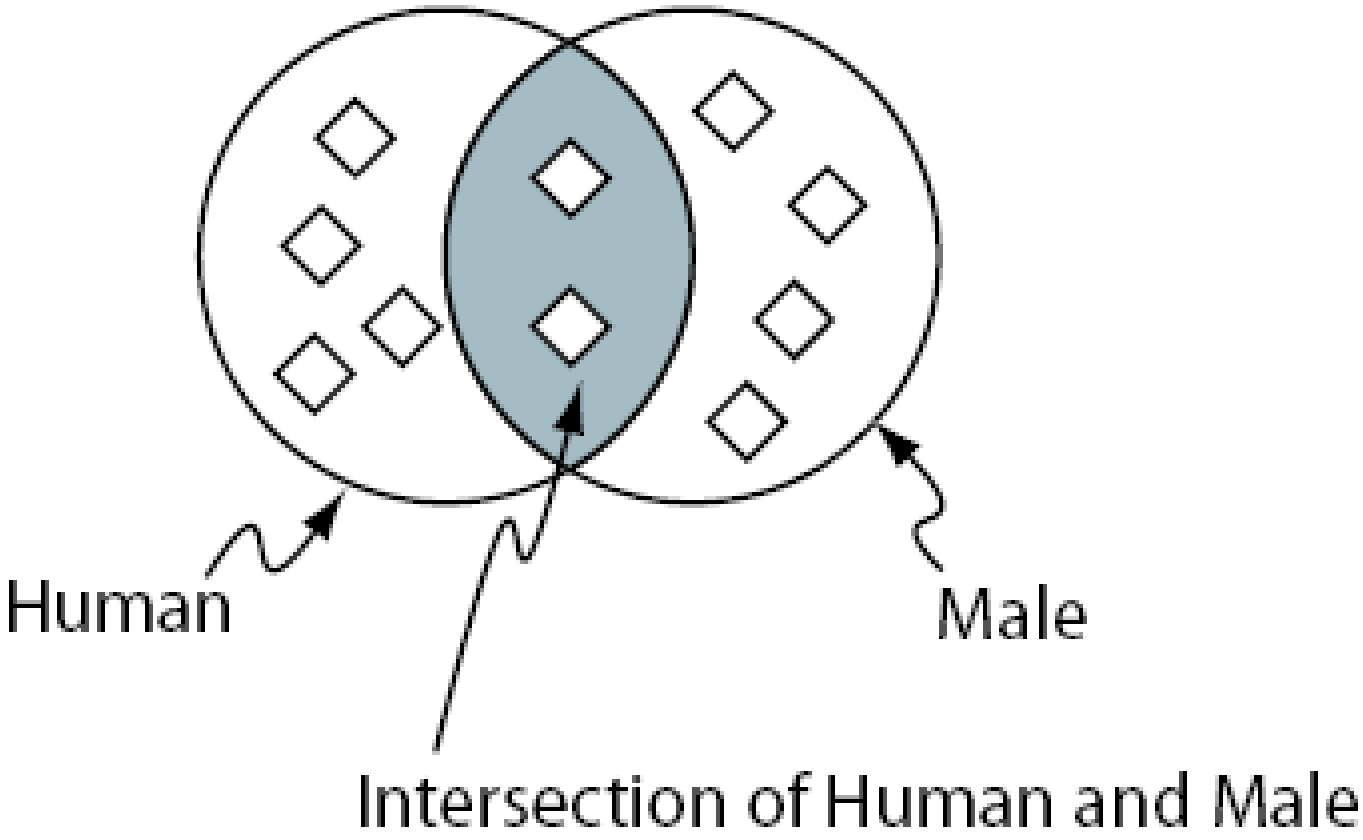
- Can be defined explicitly
- Or derived by a reasoner

Relationships among classes: Disjoint Classes

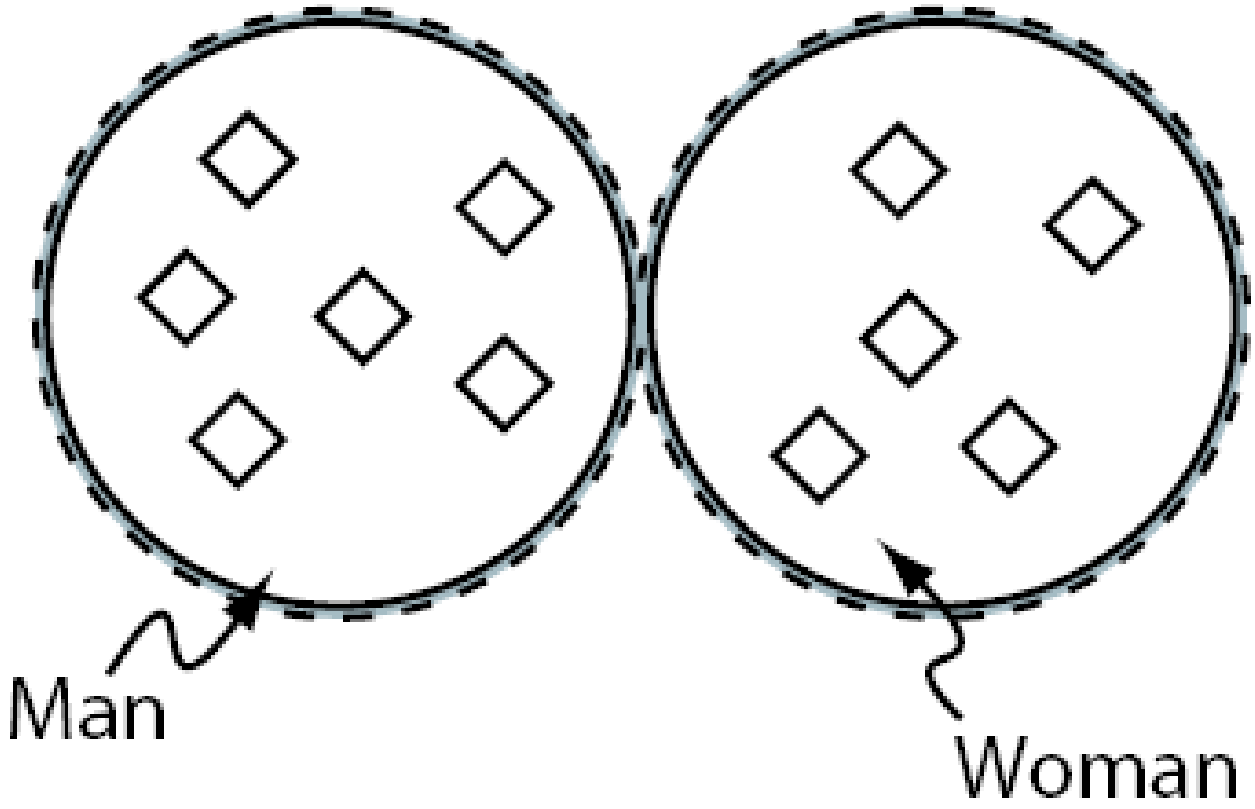
- Asserts that individual can be a member of both classes
- Default assumption that classes overlap



Relationships among classes: Class Intersection



Relationships among classes: Class Union



Properties

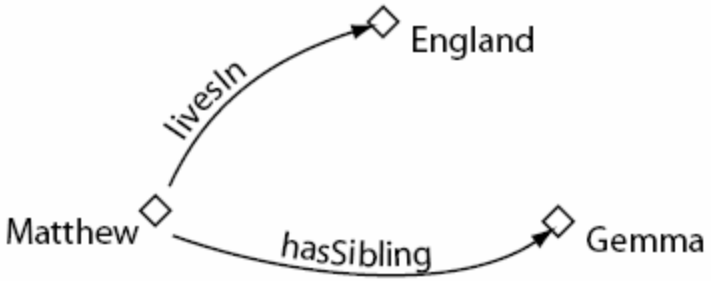
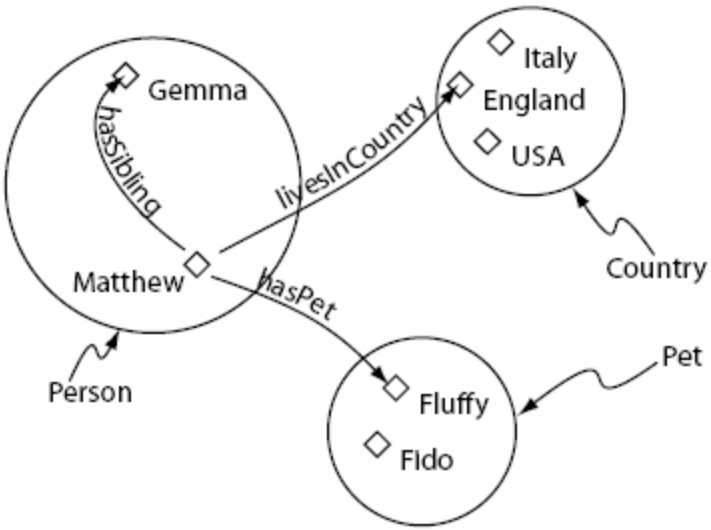
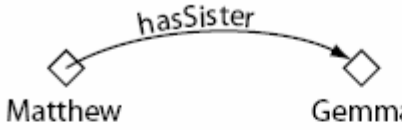


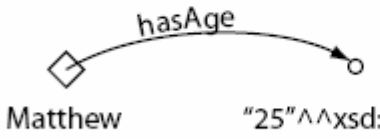
Figure 3.2: Representation Of Properties



Object vs. Data Properties



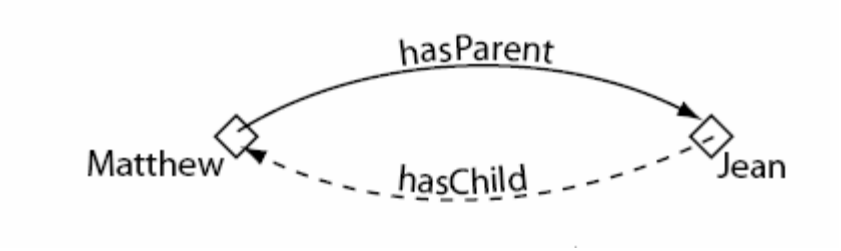
An object property linking the individual Matthew to the individual Gemma



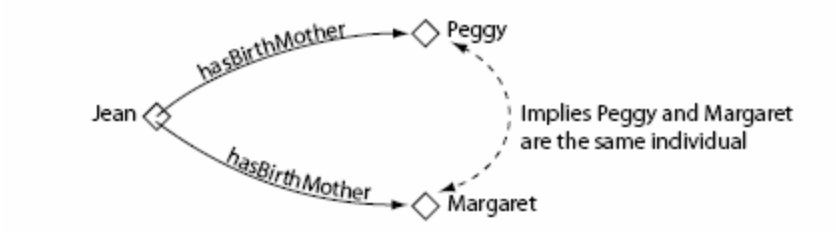
A datatype property linking the individual Matthew to the data literal '25', which has a type of an xml:integer.

Relationships among Properties

- subPropertyOf
- Inverse

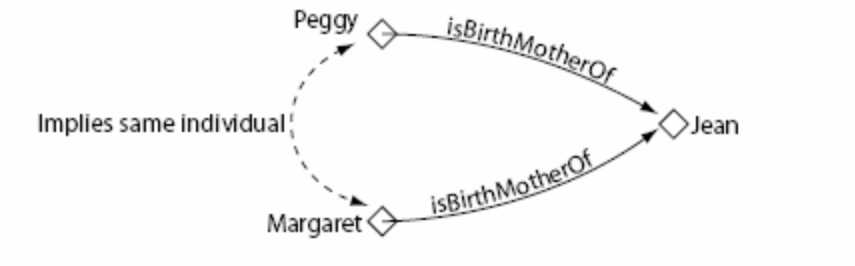


- functional

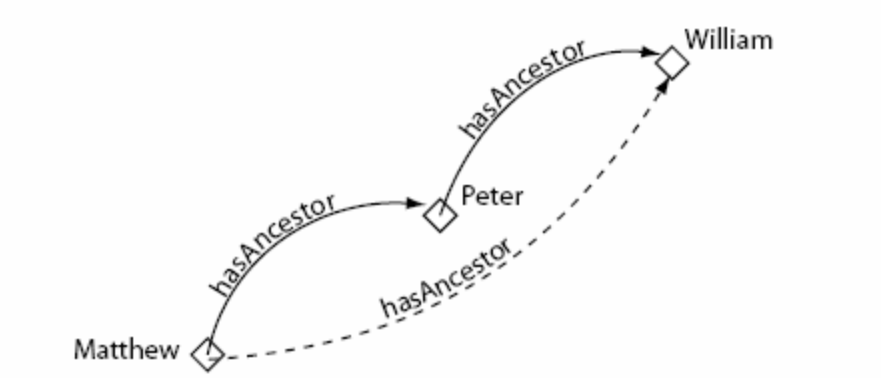


Relationships among Properties

- Inverse Functional

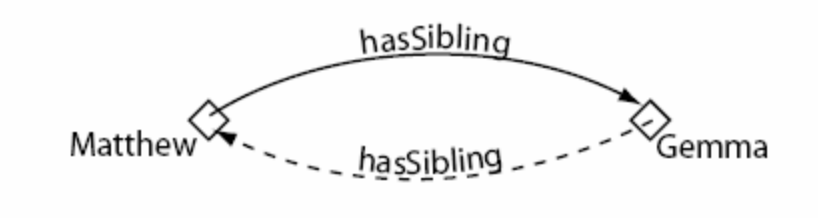


- Transitive

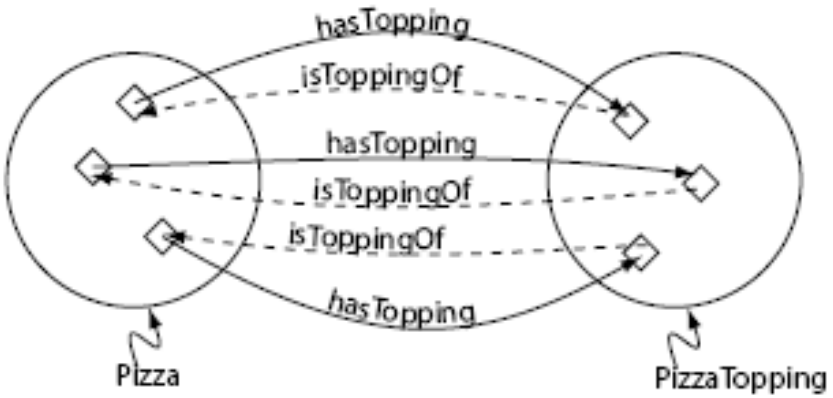


Relationships among Properties

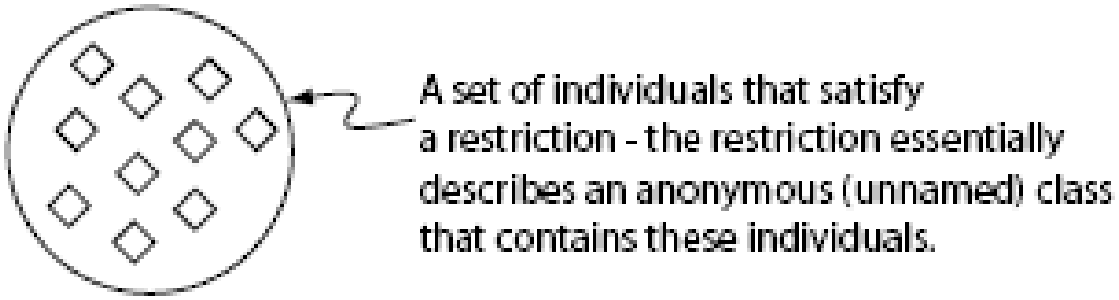
- Symmetric



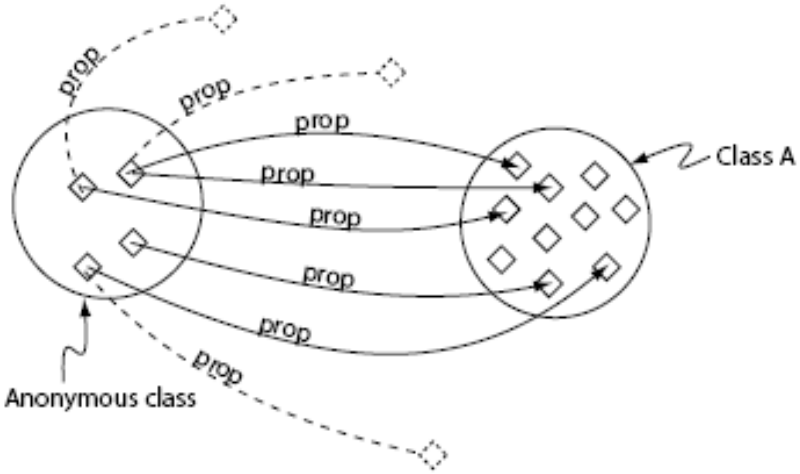
Domain and Range Constraints

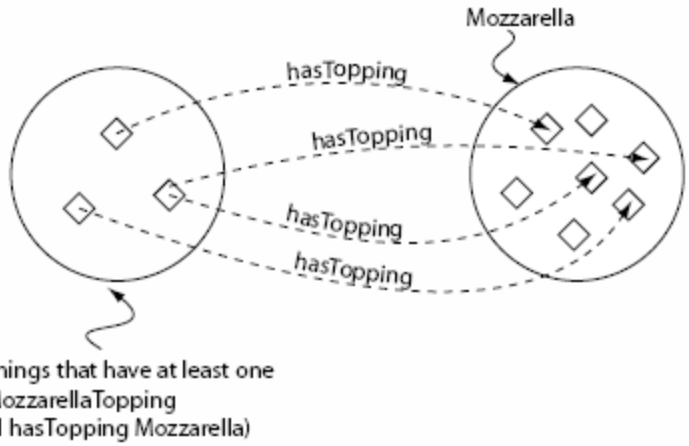


Quantifier Restrictions

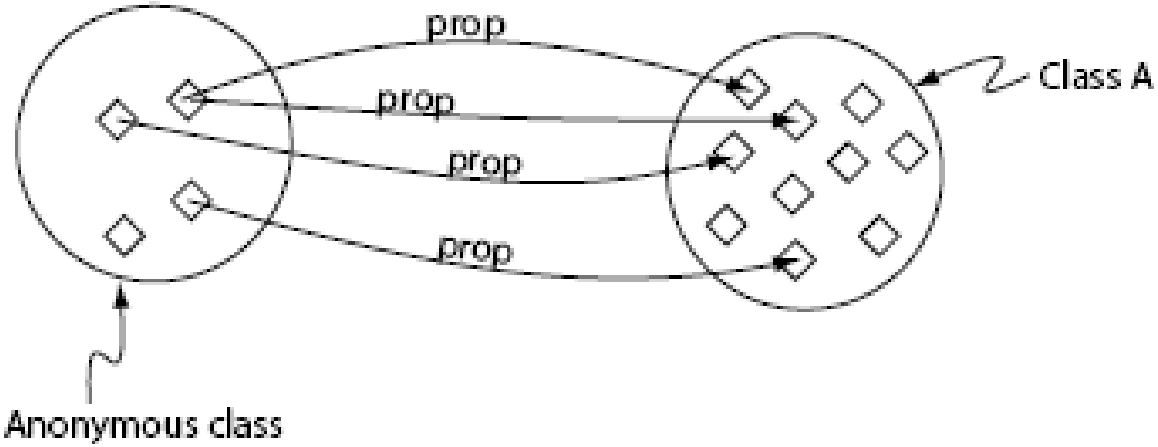


Existential Restriction

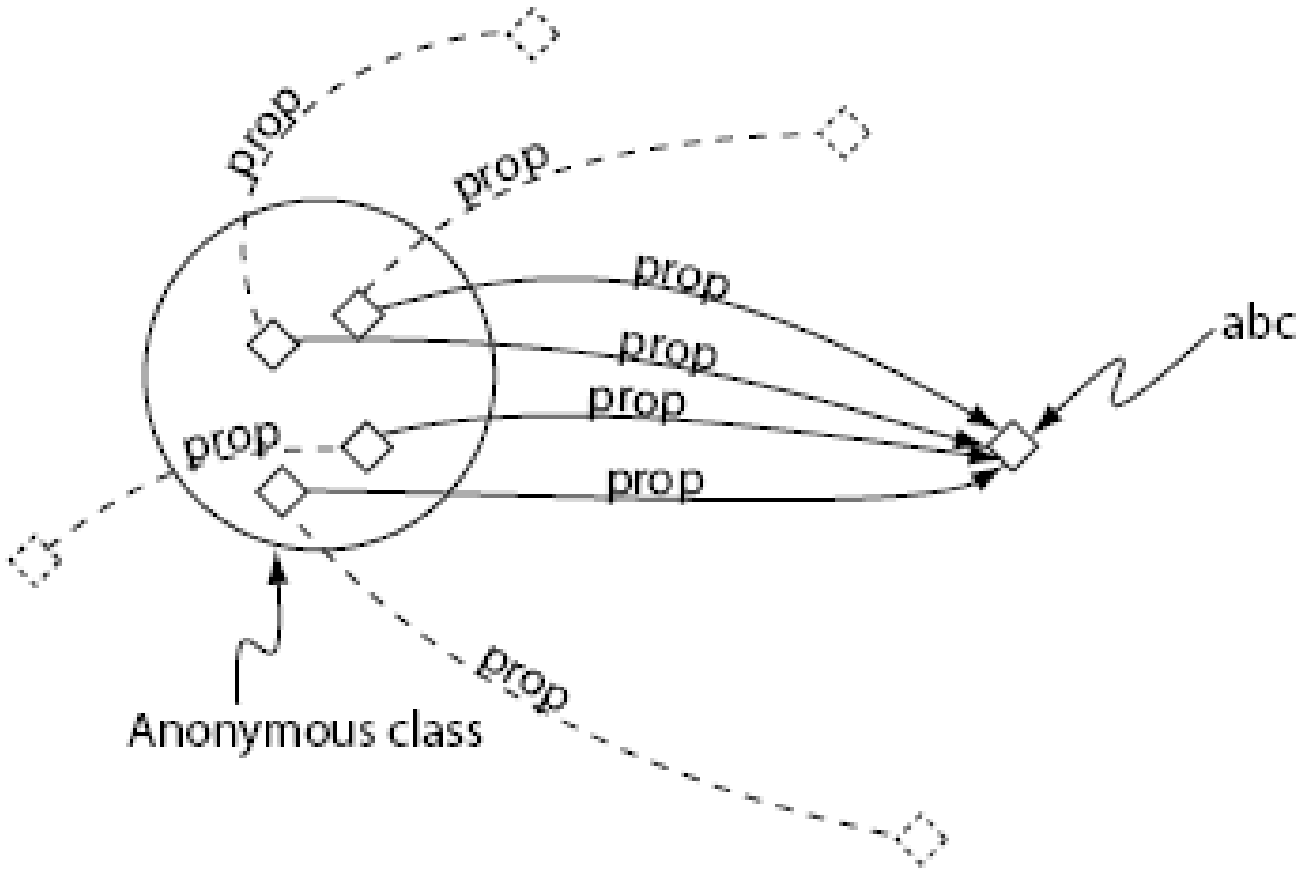




Universal Restriction

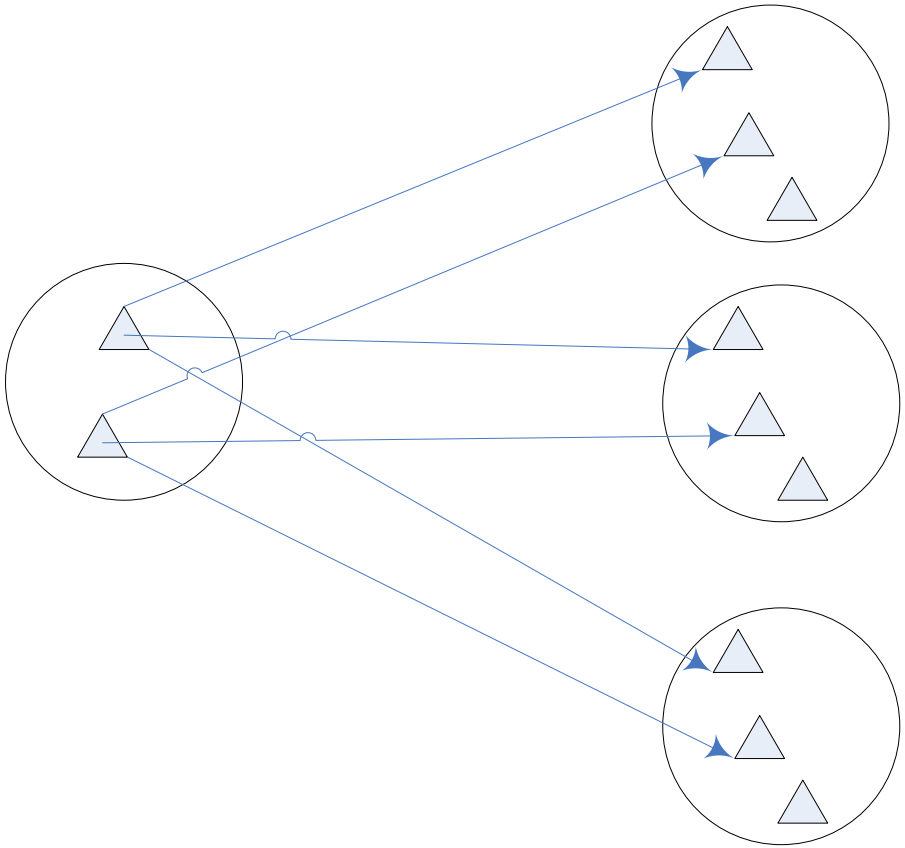


Has Value Restriction

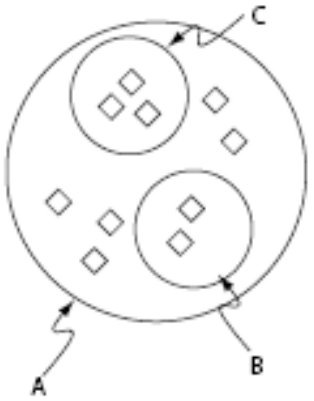


Closure Axiom on Properties

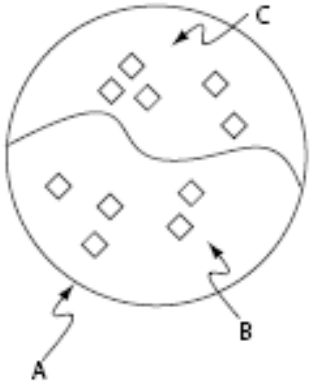
All values of a property must be a union of certain values



Covering Axiom



Without a covering axiom
(B and C are subclasses of A)



With a covering axiom
(B and C are subclasses of A
and A is a subclass of B union C)

Necessary and Sufficient Conditions

- *Necessary Conditions:* If something is a member of this class it is necessary to fulfill these conditions
  - if class member then meets condition
- *Necessary and Sufficient Conditions:* If something fulfills these conditions then it *must* be a member of this class
  - if class member then meets condition
  - if meets condition then class member
- *Primitive Class:* only has *necessary* conditions.
- *Defined Class*: has at least one set of *necessary and sufficient* conditions.