

Ontologies/Semantic Web Wrap-up

CS 431 – April 16, 2008

Carl Lagoze – Cornell University

Parts extracted with permission from:

A Practical Introduction to Ontologies & OWL

Session 1: Primitive Classes in OWL

Nick Drummond & Matthew Horridge



Parts extracted with permission from:

A Practical Introduction to Ontologies & OWL

Session 2: Defined Classes and Additional
Modelling Constructs in OWL

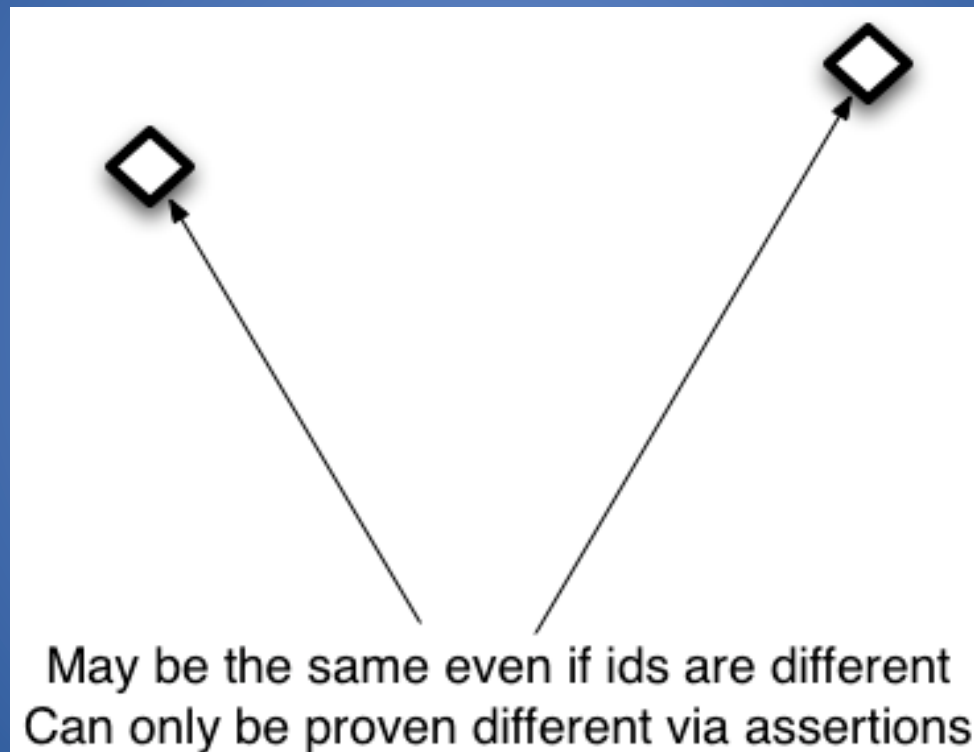
Nick Drummond & Matthew Horridge



Concepts

- Subsumption (is-a relationship)

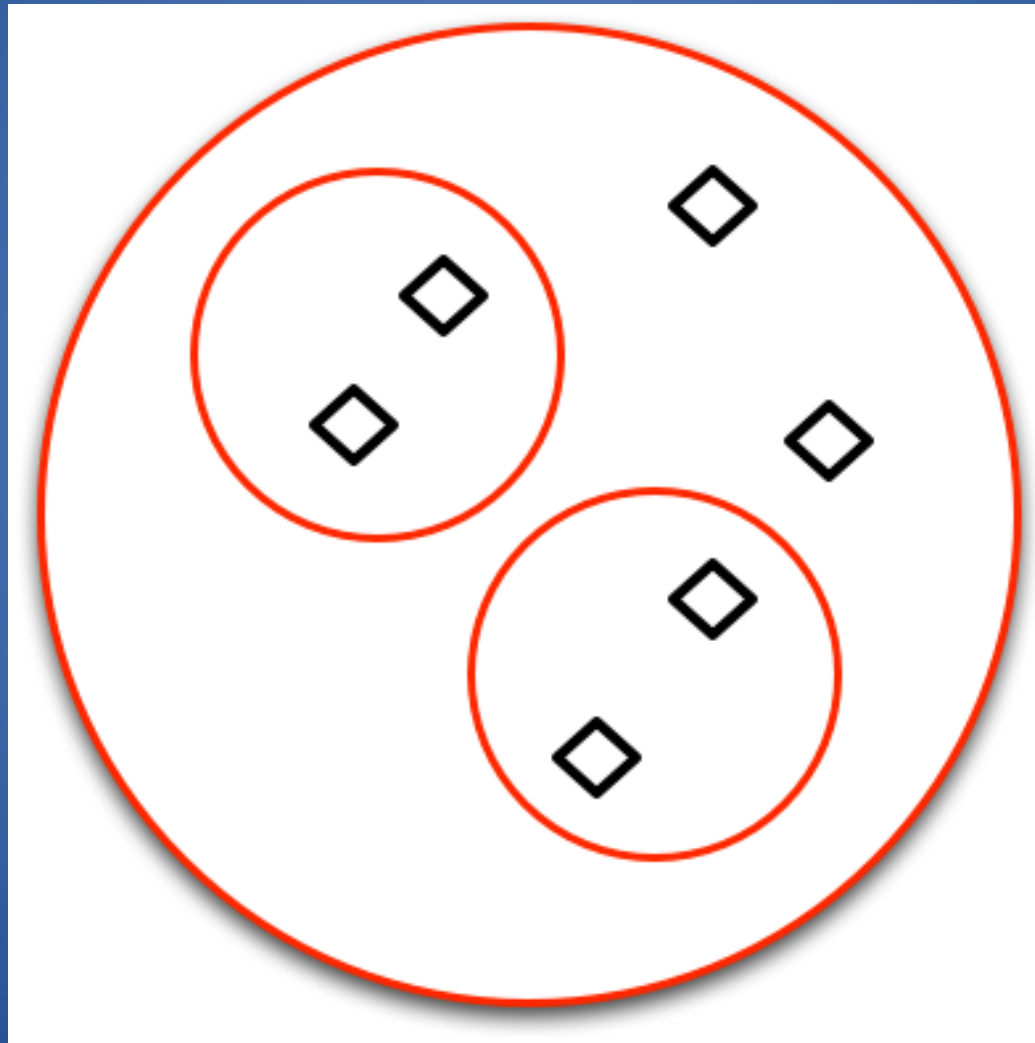
Uniqueness



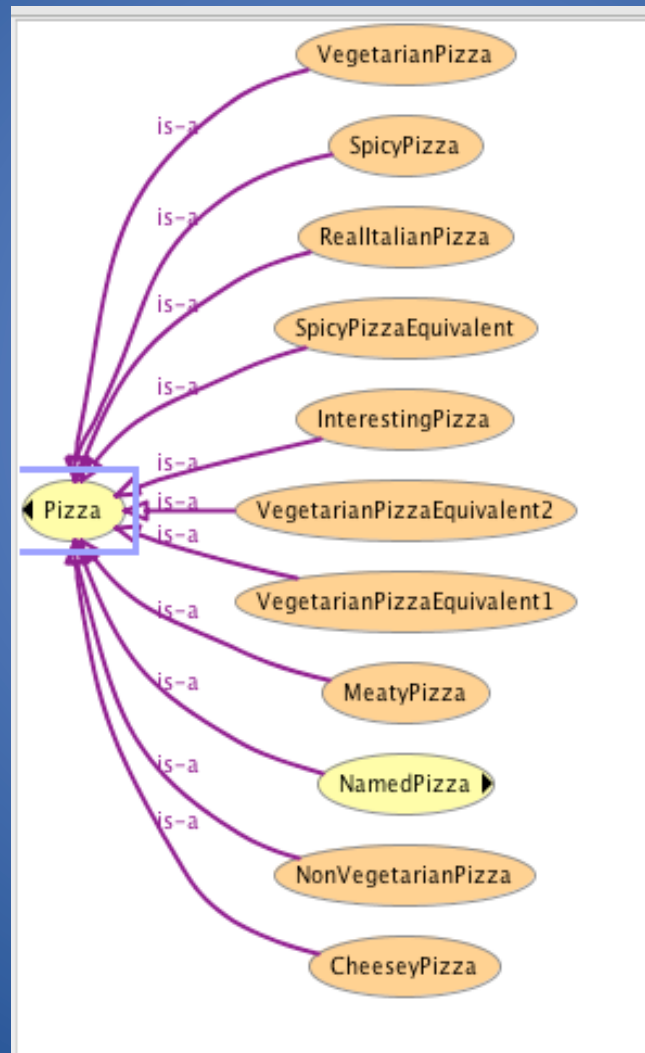
Subsumption – Class Hierarchy



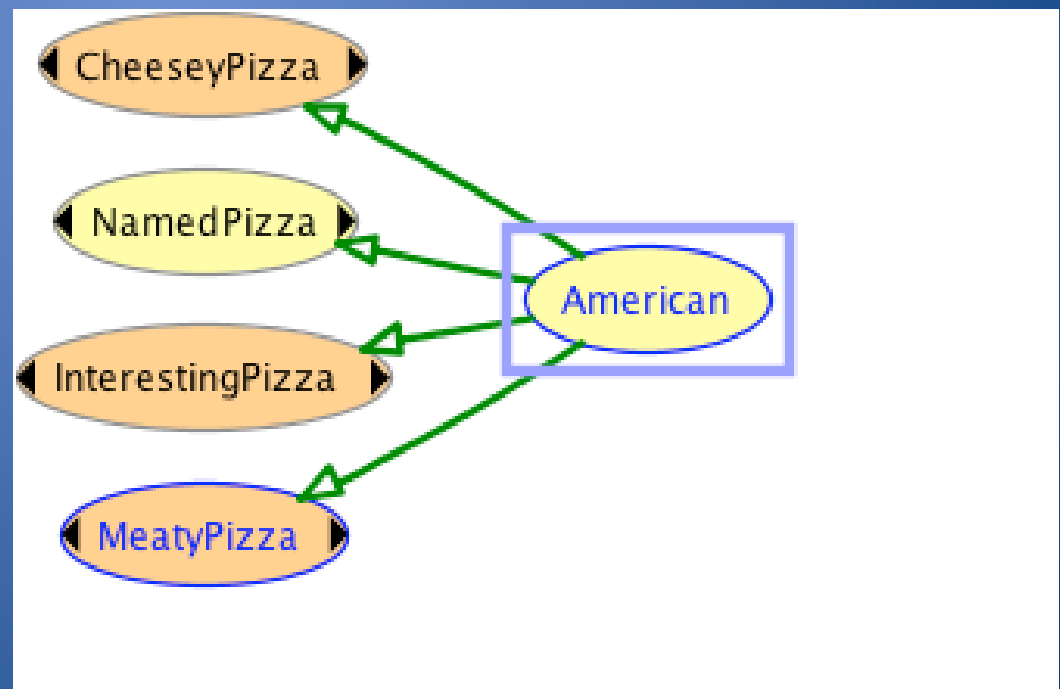
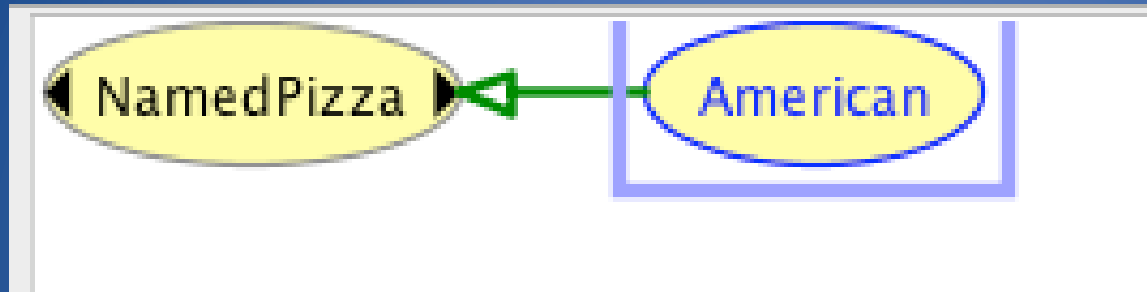
Subsumption – Class Hierarchy



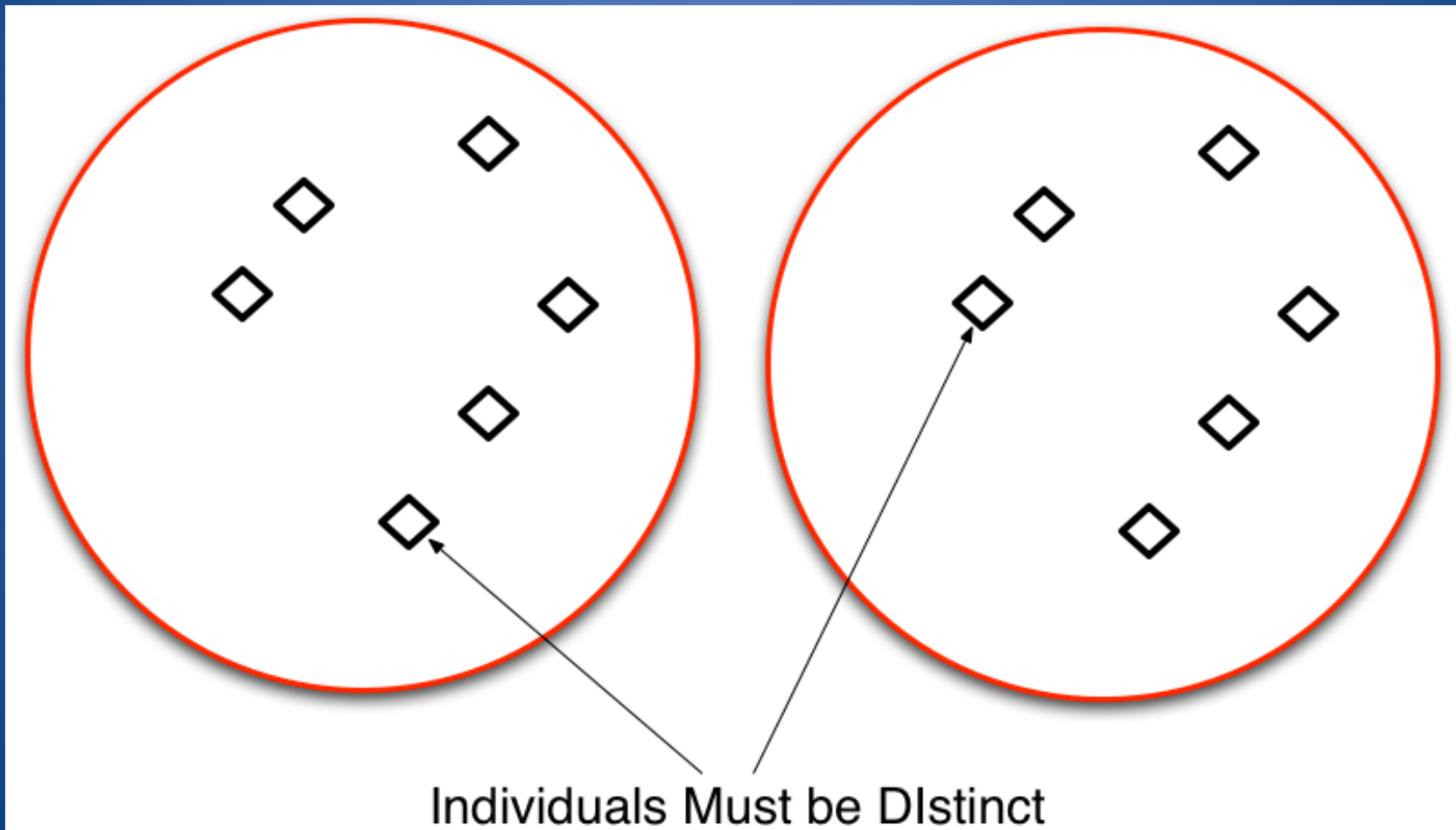
Subsumption – Class Hierarchy



Polyhierarchies (Inferred)

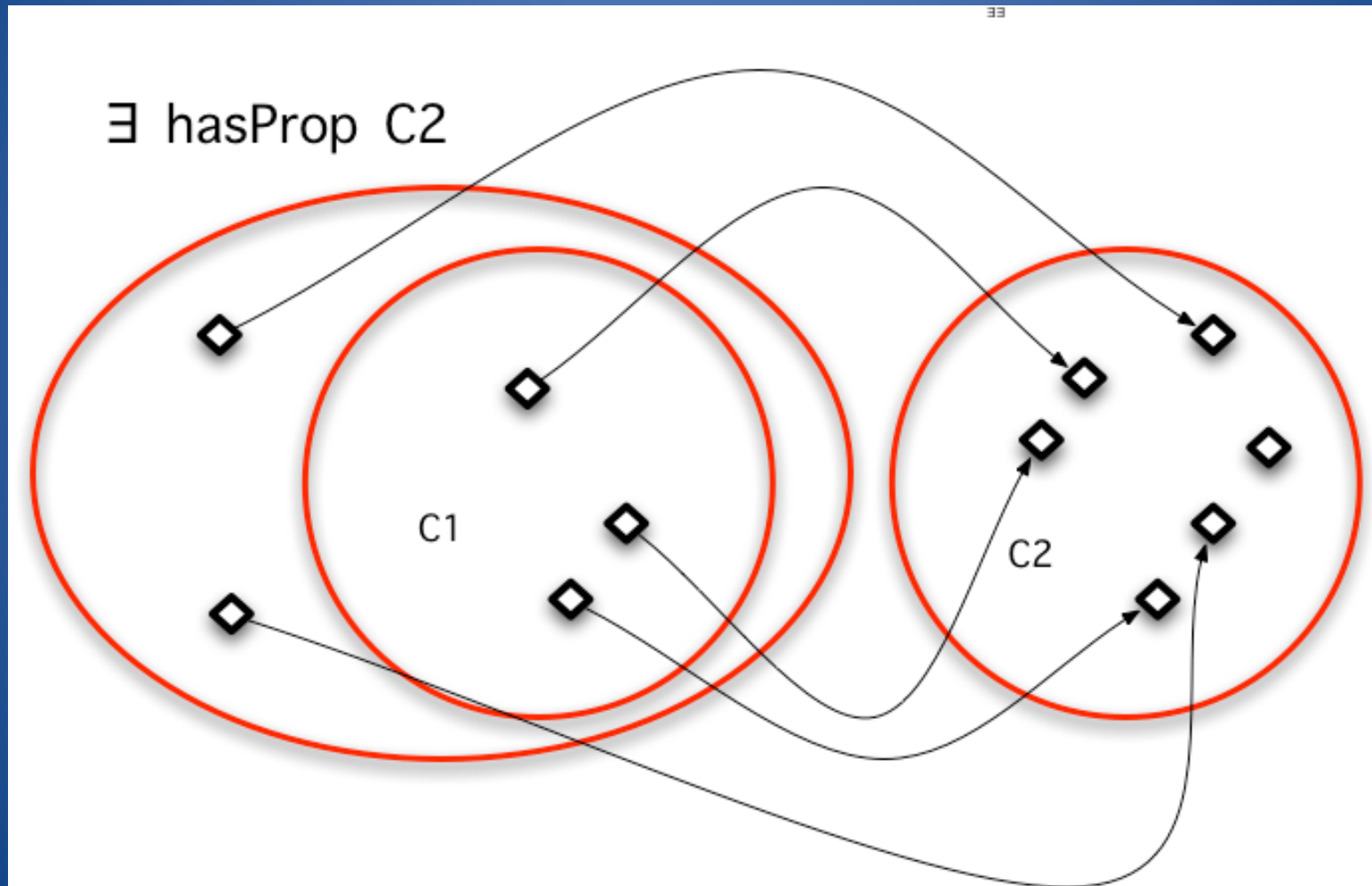


Disjoint Classes



Necessary Existential Restriction

Primitive Class



CLASS EDITOR + - F T

For Class: ● Pizza (instance of owl:Class) ☐ Inferred View

 Annotations

Property	Value	Lang
 rdfs:comment		
 rdfs:label	Pizza	en

 Asserted Conditions

DomainConcept

hasBase **some** PizzaBase

NECESSARY & SUFFICIENT

NECESSARY

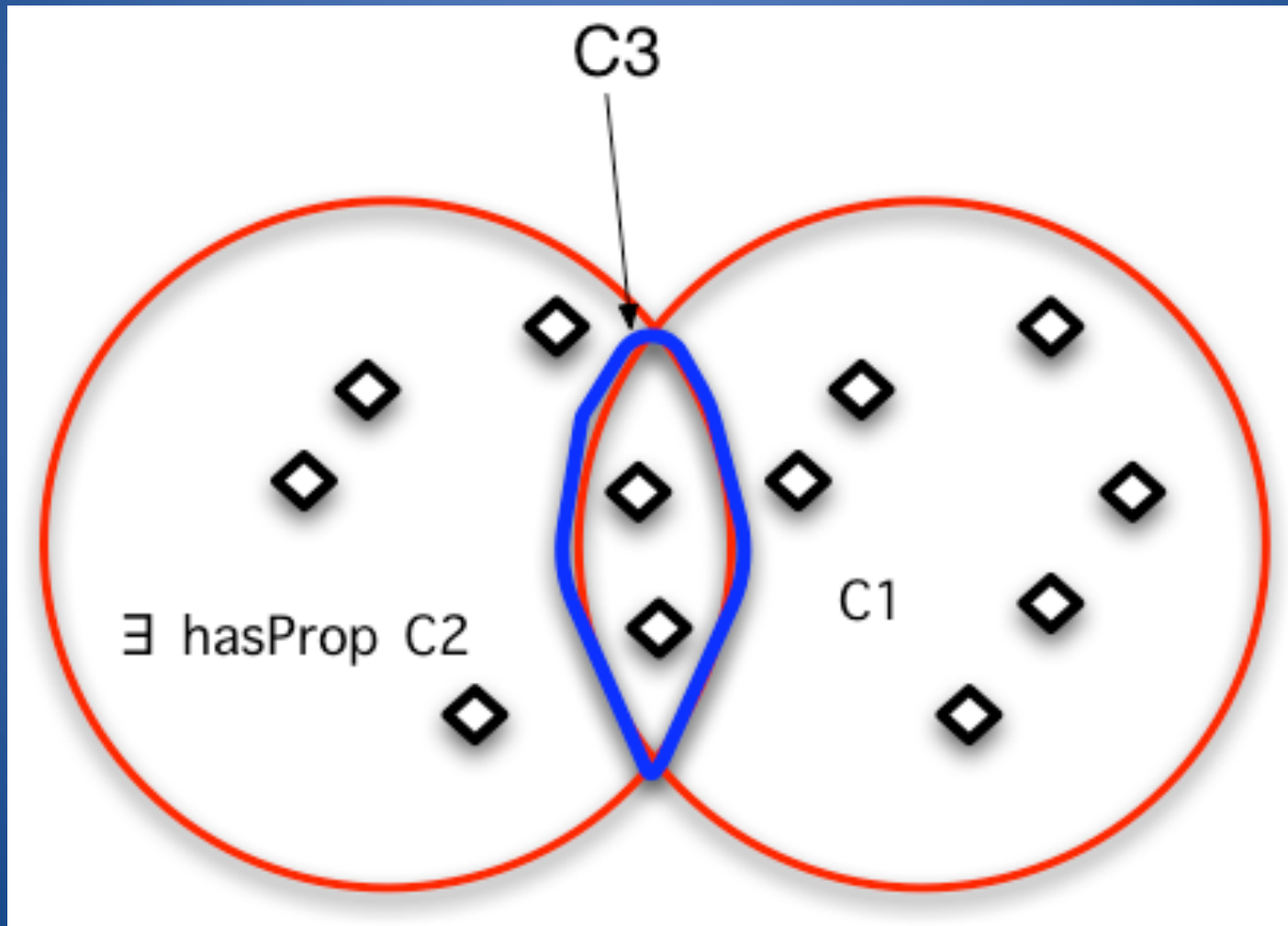
 Disjoints

● PizzaTopping

● PizzaBase

- Can say: Any pizza **MUST** have a pizza base
- Can't say: If it has a pizza base, it is a pizza

Necessary and Sufficient Existential Condition – Defined Class



CLASS EDITOR ⊕ ⊖ ⌂

For Class: (instance of owl:Class) ☐ Inferred View

⊕ ⊖ ⊕ ⊗ Annotations

Property	Value	Lang
 <code>rdfs:comment</code>	Any pizza that has at least one meat topping	en
 <code>rdfs:label</code>	PizzaDeCarne	pt

⊕ ⊖ ⊕ ⊗ **Asserted Conditions**

 Pizza — NECESSARY & SUFFICIENT ⊕

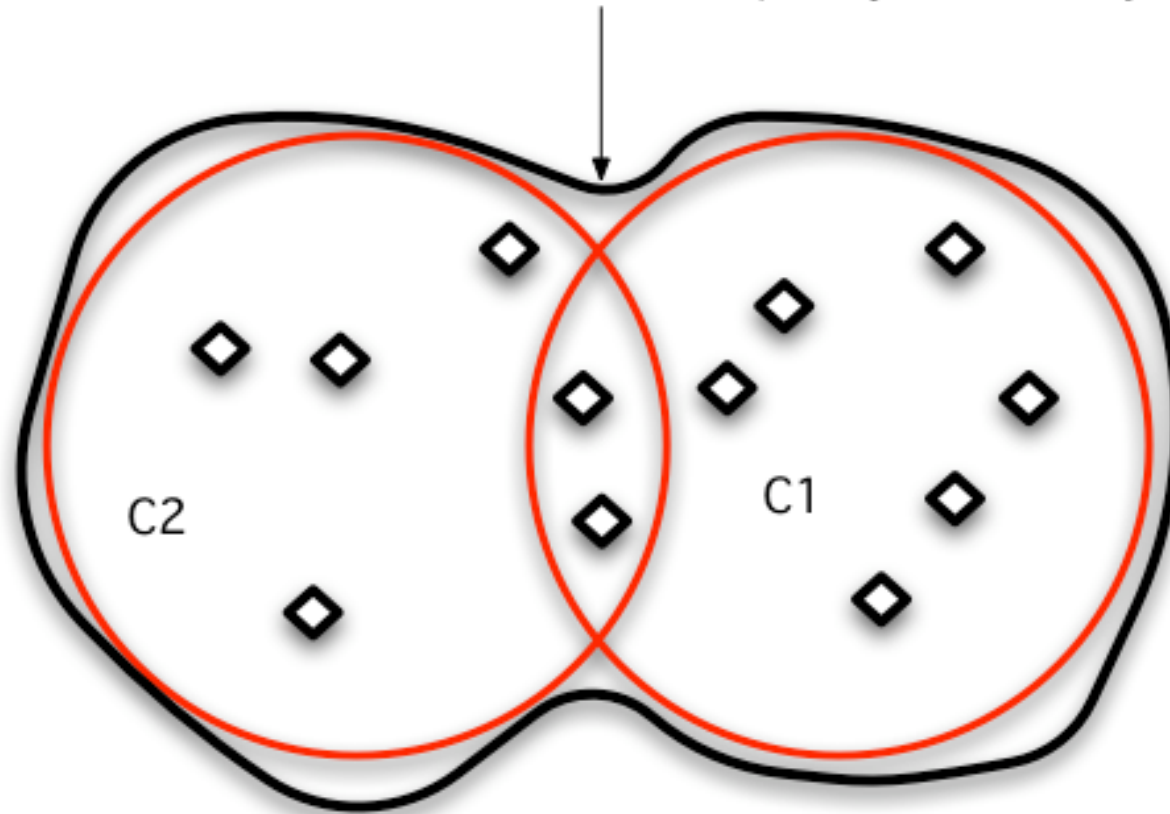
 hasTopping **some** MeatTopping — NECESSARY

 hasBase **some** PizzaBase — INHERITED [from Pizza] ⊖

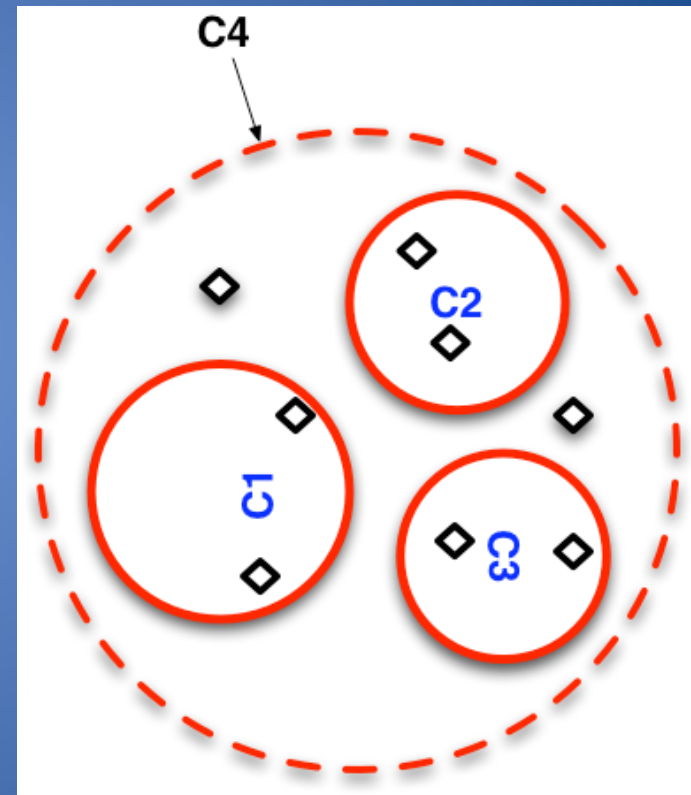
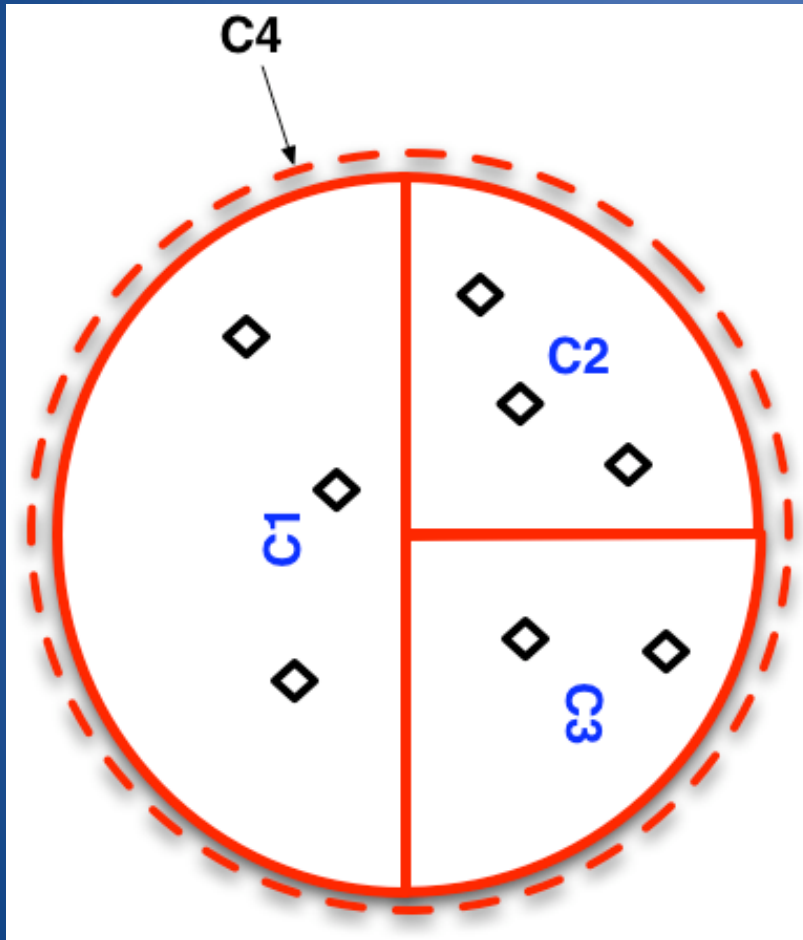
- Can say: Any Meaty Pizza MUST have a have a Meat Topping
- Can't say: If it has a Meat Topping, it is a Meaty Pizza

Union Class - Disjunction

All instances in $C1$, $C2$, and $C1 \cap C2$ (if they are not disjoint)



Union Class for Covering Axiom



Union Class for Covering Axiom

CLASS EDITOR + - F T

For Class: (instance of owl:Class) ☐ Inferred View

 Annotations

Property	Value	Lang
 <code>rdfs:comment</code>	A ValuePartition that describes only values from Hot, Medium or Mild. NB Subclasses can themselves be divided up into further partitions.	en
 <code>rdfs:label</code>	Tempero	pt

 Asserted Conditions

 Hot **or** Medium **or** Mild NECESSARY & SUFFICIENT 

 ValuePartition NECESSARY 

Spiciness must be either Hot, Medium, or Mild

Effect of the Open World Assumption

The screenshot shows the 'CLASS EDITOR' window for the 'Margherita' class. The interface includes a toolbar with icons for adding, removing, and editing properties. Below the toolbar is a table of properties and their values. The 'Property' column lists 'rdfs:comment' and 'rdfs:label'. The 'Value' column shows an empty field for 'rdfs:comment' and 'Margherita' for 'rdfs:label'. The 'Lang' column shows an empty field for 'rdfs:comment' and 'pt' for 'rdfs:label'. Below the table is a section for 'Asserted Conditions' with a toolbar for adding, removing, and editing conditions. The conditions are listed in a table with columns for the condition name, the property, the value, and the condition type. The conditions are: 'NamedPizza' (NECESSARY & SUFFICIENT), 'hasTopping some MozzarellaTopping' (NECESSARY), 'hasTopping some TomatoTopping' (NECESSARY), and 'hasBase some PizzaBase' (INHERITED). The 'hasTopping some MozzarellaTopping' and 'hasTopping some TomatoTopping' conditions are highlighted in yellow.

Property	Value	Lang
rdfs:comment		
rdfs:label	Margherita	pt

Condition	Property	Value	Condition Type
NamedPizza			NECESSARY & SUFFICIENT
hasTopping some MozzarellaTopping	hasTopping	some MozzarellaTopping	NECESSARY
hasTopping some TomatoTopping	hasTopping	some TomatoTopping	NECESSARY
hasBase some PizzaBase	hasBase	some PizzaBase	INHERITED [from Pizza]

- Margherita pizza **MUST** have Mozzarella and Tomato
 - But it **MAY** have anything else
- We need **closure** for the given property

Closure

- This is in the form of a **Universal Restriction** with a filler that is the **Union** of the other fillers for that property
- Closure works along a single property

Closure example: MargheritaPizza

All **MargheritaPizzas** must have:

- at least 1 topping from **MozzarellaTopping** and
- at least 1 topping from **TomatoTopping** and
- only toppings from **MozzarellaTopping** or **TomatoTopping**

The screenshot shows a software interface with two tabs: 'Asserted' and 'Inferred'. The 'Asserted' tab is active. Below the tabs is a section titled 'Asserted Conditions' with four icons: a yellow circle with a plus, a yellow circle with a minus, a yellow circle with a plus, and a yellow circle with a minus. Below this is a table with four rows. The first row is 'NamedPizza' with a yellow circle icon and a 'NECESSARY & SUFFICIENT' label. The second row is '∀ hasTopping (MozzarellaTopping ⊔ TomatoTopping)' with a yellow circle icon and a 'NECESSARY' label. The third row is '∃ hasTopping MozzarellaTopping' with a yellow circle icon. The fourth row is '∃ hasTopping TomatoTopping' with a yellow circle icon. Each row has a yellow button with a minus sign on the right.

Condition	Label
NamedPizza	NECESSARY & SUFFICIENT
∀ hasTopping (MozzarellaTopping ⊔ TomatoTopping)	NECESSARY
∃ hasTopping MozzarellaTopping	
∃ hasTopping TomatoTopping	

- The last part is paraphrased into “no other toppings”
- The union **closes** the hasTopping property on **MargheritaPizza**

Interesting ontologies (1)

- XFN (XHTML Friends Network)
 - <http://gmpg.org/xfn/>
- Simple list of rel tags to include in hrefs to represent human relationships

```
<a href="http://jane-blog.example.org/" rel="sweetheart date met">Jane</a>  
<a href="http://dave-blog.example.org/" rel="friend met">Dave</a>  
<a href="http://darryl-blog.example.org/" rel="friend met">Darryl</a>  
<a href="http://www.metafilter.com/">MetaFilter</a>  
<a href="http://james-blog.example.com/" rel="met">James Expert</a>
```

Interesting ontologies (1)

- FOAF (Friend of a Friend)
 - <http://www.foaf-project.org/>
- Creating a Web of machine-readable pages describing people, the links between them and the things they create and do.

```
<foaf:Person rdf:about="#me" xmlns:foaf="http://xmlns.com/foaf/0.1/">  
  <foaf:name>Dan Brickley</foaf:name>  
  <foaf:mbox_sha1sum>241021fb0e6289f92815fc210f9e9137262c252e</foaf:mbox_sha1sum>  
  <foaf:homepage rdf:resource="http://danbri.org/" />  
  <foaf:img rdf:resource="/images/me.jpg" />  
</foaf:Person>
```

Applications of Social Network Ontologies

- Google Social Graph API
 - <http://code.google.com/apis/socialgraph/docs/>
- SIOC initiative (Semantically-Interlinked Online Communities)
 - <http://sioc-project.org/>

Embedding Semantic Information in HTML: Microformats and RDF-a

CS 431 – April 16, 2008

Carl Lagoze – Cornell University

The general idea

- Almost all content on the web is in HTML
- People are not going to produce parallel semantic information (e.g., RDF/XML) for web pages
- So why not define a way that machine-readable semantic information can be embedded in HTML

Microformats

- Use Class, Rel, and Rev attributes in HTML to embed semantic information
- Define microformat standards for specific purposes
- Result: machine parsing can extract semantics while maintaining presentation purpose of HTML
- <http://microformats.org/>

Example 1: hcard

```
BEGIN:VCARD  
VERSION:3.0  
N:Lawless;Derek  
FN:Derek Lawless  
EMAIL:hi@dereklawless.net  
URL:http://dereklawless.net/  
ORG:TransGloboCorp  
END:VCARD
```

```
<div class="vcard">  
  <a class="url fn" href="http://  
dereklawless.net/">Derek Lawless</a>  
  <a class="email"  
href="mailto:hi@dereklawless.net" />  
  <div class="org">TransGloboCorp</div>  
</div>
```

Presentation with embedded semantics

```
<div id="hcard-Carl-Lagoze" class="vcard">
  <span class="fn">Carl Lagoze</span>
  <div class="org">Cornell University</div>
  <a class="email"
href="mailto:lagoze@cs.cornell.edu">lagoze@cs.cornell.edu</a>
  <div class="adr">
    <div class="street-address">301 College Ave.</div>
    <span class="locality">Ithaca</span>
    '
    <span class="region">NY</span>
    '
    <span class="postal-code">14850</span>
```

Carl Lagoze
Cornell University
lagoze@cs.cornell.edu
301 College Ave.
Ithaca , NY , 14850
607-255-6046

Example 2: hcalendar

```
BEGIN:VCALENDAR
PRODID:-//XYZproduct//EN
VERSION:2.0
BEGIN:VEVENT
URL:http://www.web2con.com/
DTSTART:20071005
DTEND:20071020
SUMMARY:Web 2.0 Conference
LOCATION:Argent Hotel\, San Francisco\, CA
END:VEVENT
END:VCALENDAR
```

```
<div class="vevent">
  <a class="url" href="http://www.web2con.com/">http://www.web2con.com/</a>
  <span class="summary">Web 2.0 Conference</span>:
  <abbr class="dtstart" title="2007-10-05">October 5</abbr>-
  <abbr class="dtend" title="2007-10-20">19</abbr>,
  at the <span class="location">Argent Hotel, San Francisco, CA</span>
</div>
```

Web 2.0 Conference: October 5- 19, at the Argent Hotel, San Francisco, CA

RDFa

- Extend XHTML to embed RDF
 - XHTML can be rendered by browser
 - Triples can be extracted via programs (GRIDDL)
- In proposal stage since it is not currently valid html

GRDDL

- Gleaning Resource Descriptions from Dialects of Languages
- A technique for obtaining RDF data from XML documents and in particular XHTML pages.
- Mechanisms
 - Authors may explicitly associate documents with transformation algorithms, typically represented in XSLT, using a link element in the head of the document.
 - the information needed to obtain the transformation may be held in an associated metadata profile document or namespace document.
 - Mappings exist for common microformats
 - Slide content borrowed from <http://www.w3.org/2001/sw/grddl-wg/tut7/gtut.html>

GRDDL Goal

- Provide common mechanism to exploit RDF for mashups
- By translating to RDF makes it possible for storage in triple stores (common representation) and query with SPARQL (common query language)

Example Problem

When can Jane, David, and Robin meet?

- Jane keeps **her calendar** in one format (RDFa)
- David uses another (eRDF) for **his calendar**
- Robin keeps **his schedule** in a 3rd format (hCalendar)