Representing and reasoning over relations in ontologies – Tutorial –

C. Maria Keet

Department of Computer Science, University of Cape Town
mkeet@cs.uct.ac.za

Forum on Artificial Intelligence Research,
Stellenbosch, South Africa, December 3-5, 2014
Outline

1. Introduction
2. Semantics of relations
   - Positionalism
   - Hierarchies of relations
3. Some common relations
   - Part-whole relations
   - Mereotopology
   - Beyond parts and space
4. Modelling and reasoning
   - Reasoner-mediated modelling
   - Performance considerations
   - Hands-on
5. Recap
<table>
<thead>
<tr>
<th>Introduction</th>
<th>Semantics of relations</th>
<th>Some common relations</th>
<th>Modelling and reasoning</th>
<th>Recap</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td>2 Semantics of relations</td>
<td>3 Some common relations</td>
<td>4 Modelling and reasoning</td>
<td>5 Recap</td>
</tr>
<tr>
<td></td>
<td>3 Positionalism</td>
<td>3 Part-whole relations</td>
<td>3 Reasoner-mediated modelling</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 Hierarchies of relations</td>
<td>3 Mereotopology</td>
<td>3 Performance considerations</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 Beyond parts and space</td>
<td>3 Hands-on</td>
<td></td>
</tr>
</tbody>
</table>
Setting

- Representing hierarchies of classes
  [/concepts/universals/entity types/...] typically received first/most/only attention

- Things become interesting from the viewpoint of automated reasoning only if there are other axioms, or: properties of those classes
Setting

- Representing hierarchies of classes
  [/concepts/universals/entity types/...] typically received first/most/only attention

- Things become interesting from the viewpoint of automated reasoning only if there are other axioms, or: properties of those classes

  ⇒ How to model those? (and have good quality)

  ⇒ What effect does that have on the deductions? (preferably desired ones)
Some problematic examples with relationships

A. Trans(partOf)
   Hand $\sqsubseteq \exists$partOf.Musician
   Musician $\sqsubseteq \exists$partOf.Orchestra
   Deducing that each Hand is part of an Orchestra is ‘wrong’
Some problematic examples with relationships

A. Trans(partOf)
   Hand ⊑ ∃partOf.Musician
   Musician ⊑ ∃partOf.Orchestra
   Deducing that each Hand is part of an Orchestra is ‘wrong’

B. hasMainTable ◦ hasFeature ⊑ hasFeature
   hasMainTable ⊑ DataSet × DataTable
   hasFeature ⊑ DataTable × Feature
Some problematic examples with relationships

A. Trans(partOf)
   Hand ⊆ ∃ partOf.Musician
   Musician ⊆ ∃ partOf.Orchestra
   Deducing that each Hand is part of an Orchestra is ‘wrong’

B. hasMainTable ◦ hasFeature ⊆ hasFeature
   hasMainTable ⊆ DataSet × DataTable
   hasFeature ⊆ DataTable × Feature
   Deduces DataSet ⊆ DataTable, which is ‘wrong’

<table>
<thead>
<tr>
<th></th>
<th>A. SubPropertyOf(PropertyChain(contains hasPart) contains)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>contains</td>
</tr>
<tr>
<td></td>
<td>haspart</td>
</tr>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>B. SubPropertyOf(PropertyChain(hasPart contains) hasPart)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>C.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mary’s mouth</td>
</tr>
<tr>
<td></td>
<td>contains</td>
</tr>
<tr>
<td></td>
<td>haspart</td>
</tr>
<tr>
<td></td>
<td>Legominifigure1</td>
</tr>
<tr>
<td></td>
<td>contains</td>
</tr>
<tr>
<td></td>
<td>haspart</td>
</tr>
<tr>
<td></td>
<td>Legominifigure1’s leg</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>D.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>haspart</td>
</tr>
<tr>
<td></td>
<td>Mary’s mouth</td>
</tr>
<tr>
<td></td>
<td>contains</td>
</tr>
<tr>
<td></td>
<td>haspart</td>
</tr>
<tr>
<td></td>
<td>Legominifigure1</td>
</tr>
</tbody>
</table>
And old issue

(A1) Class hierarchy with asserted conditions
(A2) Other class hierarchy with the same asserted conditions
(B) Correct role box (object properties)
(C) Wrong role box (object properties)

(Live with Protégé)
And old issue

- (Live with Protégé)
- A1+B: OK; A2+B: OK
- A1+C: Chassis inconsistent; A2+C: Chassis (re)classified as a PD
And old issue

- **A1.** Class hierarchy with asserted conditions
  - Assorted hierarchy:
    - `owl:Thing`
    - `PT`
    - `ED`
    - Chassis
    - Car
    - `PD`
  - Assorted Conditions:
    - `Necessary & Sufficient` `E`
    - `Necessary` `E`
    - `Part of some Car` `E`

- **B.** Correct role box (object properties)
  - Object properties:
    - Domain: `PT`
    - Range: `PT`

- **C.** Wrong role box (object properties)
  - Object properties:
    - Domain: `PD`
    - Range: `PD`

- **A2.** Other class hierarchy with the same asserted conditions
  - Assorted hierarchy:
    - `owl:Thing`
    - `PT`
    - `ED`
    - Chassis
    - Car
    - `PD`
  - Assorted Conditions:
    - `Necessary & Sufficient` `E`
    - `Necessary` `E`
    - `Part of some Car` `E`


- **(Live with Protégé)**
- **A1+B:** OK; **A2+B:** OK
- **A1+C:** Chassis inconsistent; **A2+C:** Chassis (re)classified as a PD

**C.** But actually, the property hierarchy is **wrong** (mostly ignored by the DL/OWL reasoner, so can’t find that mistake)
Other modelling and implementation issues

**Poll:** are teaches and taught by two relations?
Other modelling and implementation issues

**Poll**: are teaches and taught by two relations?

⇒ differentiate between *relation* between entities and *relational expression* describing that state
Other modelling and implementation issues

- **Poll**: are teaches and taught by two relations?
  \[\Rightarrow\] differentiate between relation between entities and relational expression describing that state

- **Poll**: How do you map UML’s association ends (or ORM’s roles) to an OWL object property (or vv.)?
Other modelling and implementation issues

- **Poll**: are teaches and taught by two relations?
  - ⇒ differentiate between *relation* between entities and *relational expression* describing that state

- **Poll**: How do you map UML’s association ends (or ORM’s roles) to an OWL object property (or vv.)?
  - ⇒ Bit tricky, you have to make a modelling decision...

- These two questions surface as a consequence of different ontological commitment as to what a relation really is (or what you’re convinced of it is)
A few other modelling questions

- Should you introduce a minimum amount of properties in your ontology, or many?
- Always (try to) declare domain and range axioms?
- Use explicit inverses (extending the vocabulary) or not?
- What about ternaries?
- How to find and fix mistakes and pitfalls?
- What if solution X is better modelling than option Y but computationally more costly than Y?
Outline

1. Introduction

2. Semantics of relations
   - Positionalism
   - Hierarchies of relations

3. Some common relations
   - Part-whole relations
   - Mereotopology
   - Beyond parts and space

4. Modelling and reasoning
   - Reasoner-mediated modelling
   - Performance considerations
   - Hands-on

5. Recap
Toward solving such issues

- Meaning of relations
  - Different modelling/representation languages have varying ‘ontological commitments’
  - When a relation(ship) is a specialisation of another
- Reuse relations that are already investigated widely cf. reinventing the wheel
- Methods and tools to avoid pitfalls
A note from philosophy

- Relations investigated in philosophy
  - Nature and properties of some specific relations (parthood, portions, participation, causation)
  - ‘Categories’ of relations (material, formal) (e.g., [Guizzardi and Wagner(2008)])
  - Nature of relation itself (standard, positionalist, anti-positionalist)

- Some results more useful for ontologies and conceptual modelling than others, some even for tool development
What relations are

- Applied to trying to resolve issues in ORM formalisations and tools [Keet(2009)]
- Not the arguments here, only present what they are
What relations are

- Applied to trying to resolve issues in ORM formalisations and tools [Keet(2009)]
- Not the arguments here, only present what they are
- Standard view relies on linguistics and the English language in particular
- Formalisation predicate-centred, order of entities important
Graphical depictions positionalist, anti-positionalist

A. Positionalist

B. Anti-positionalist

- Positionalist needs argument places in the “fundamental furniture of the universe”, anti-positionalist does not
Graphical depictions positionalist, anti-positionalist

- Positionalist needs argument places in the “fundamental furniture of the universe”, anti-positionalist does not.
- UML Class Diagrams, ORM, ER all positionalist [Keet and Fillottrani(2013)], most of DL and FOL take standard view.
Questions and Problems to address

- Modelling flaws in the RBox show up as unexpected or undesirable deductions regarding classes in the TBox, but current explanation algorithms (e.g., [Horridge et al. (2008), Parsia et al. (2005), Kalyanpur et al. (2006)]) mostly do not point to the actual flaw in the RBox

- What are the features of a ‘good’ RBox w.r.t. object property expressions?

- What type of flaws are being made?

- See [Keet (2014)]
Preliminaries (1/2)

- “basic form” for sub-properties, i.e., \( S \sqsubseteq R \),
- “complex form” with property chains
- \( R \sqsubseteq C_1 \times C_2 \) as shortcut for domain and range axioms
  \( \exists R \sqsubseteq C_1 \) and \( \exists R^- \sqsubseteq C_2 \) where \( C_1 \) and \( C_2 \) are generic classes;
  ObjectPropertyDomain(OPE CE) and
  ObjectPropertyRange(OPE CE) in OWL.
- \( R \sqsubseteq \top \times \top \) when no domain and range axiom has been declared

Definition (User-defined Domain and Range Classes)

Let \( R \) be an OWL object property and \( R \sqsubseteq C_1 \times C_2 \) its associated domain and range axiom. Then, with the symbol \( D_R \) we indicate the User-defined Domain of \( R \)—i.e., \( D_R = C_1 \)—and with the symbol \( R_R \) we indicate the User-defined Range of \( R \)—i.e., \( R_R = C_2 \).
Definition ((Regular) Role Inclusion Axioms ([Horrocks et al.(2006)]))

Let $\prec$ be a regular order on roles. A role inclusion axiom (RIA for short) is an expression of the form $w \sqsubseteq R$, where $w$ is a finite string of roles not including the universal role $U$, and $R \neq U$ is a role name. A role hierarchy $\mathcal{R}_h$ is a finite set of RIAs. An interpretation $\mathcal{I}$ satisfies a role inclusion axiom $w \sqsubseteq R$, written $\mathcal{I} \models w \sqsubseteq R$, if $w^\mathcal{I} \subseteq R^\mathcal{I}$. An interpretation is a model of a role hierarchy $\mathcal{R}_h$ if it satisfies all RIAs in $\mathcal{R}_h$, written $\mathcal{I} \models \mathcal{R}_h$. A RIA $w \sqsubseteq R$ is $\prec$-regular if $R$ is a role name, and

- $w = R \circ R$, or
- $w = R^-$, or
- $w = S_1 \circ \ldots \circ S_n$ and $S_i \prec R$, for all $1 \geq i \geq n$, or
- $w = R \circ S_1 \circ \ldots \circ S_n$ and $S_i \prec R$, for all $1 \geq i \geq n$, or
- $w = S_1 \circ \ldots \circ S_n \circ R$ and $S_i \prec R$, for all $1 \geq i \geq n$.

Finally, a role hierarchy $\mathcal{R}_h$ is regular if there exists a regular order $\prec$ such that each RIA in $\mathcal{R}_h$ is $\prec$-regular.
Object sub-properties

- Given $S \sqsubseteq R$, then all individuals in the property assertions involving property $S$ must also be related to each other through property $R$ (OWL 2 Spec.).
- Subsumption for OWL object properties (DL roles) holds if the subsumed property is more constrained such that in every model, the set of individual property assertions is a subset of those of its parent property.
Object sub-properties

- Given $S \sqsubseteq R$, then all individuals in the property assertions involving property $S$ must also be related to each other through property $R$ (OWL 2 Spec.).

- Subsumption for OWL object properties (DL roles) holds if the subsumed property is more constrained such that in every model, the set of individual property assertions is a subset of those of its parent property.

- Two ways to constrain a property, and either one suffices:
  - By specifying its domain or range
  - By declaring the property’s characteristics
Constraining a property

Figure: A: Example, alike the so-called ‘subsetting’ idea in UML; B: hierarchy of property characteristics (Based on Halpin 2001, 2011)
Constraining a property

Figure: A: Example, alike the so-called ‘subsetting’ idea in UML; B: hierarchy of property characteristics relevant for OWL 2.
Outline Sub-Property compatibility Service (*SubProS*)

- First part extends the basic notions from the *RBox compatibility* [Keet and Artale(2008)] (defined for *ALCQI*)
- Informally, it first checks the ‘compatibility’ of domain and range axioms w.r.t the object property hierarchy and the class hierarchy.
Outline Sub-Property compatibility Service (*SubProS*)

- First part extends the basic notions from the *RBox compatibility* [Keet and Artale(2008)] (defined for *ALCQI*)

- Informally, it first checks the ‘compatibility’ of domain and range axioms w.r.t the object property hierarchy and the class hierarchy.

- After that, *SubProS* checks whether the object property characteristic(s) conform to specification, provided there is such an expression in the ontology.

- It exhaustively checks each permutation of domain and range and then of the characteristic of the parent and child property in the object property hierarchy
Definition (Sub-Property compatibility Service (SubProS))

For each pair of object properties, $R, S \in \mathcal{O}$ such that $\mathcal{O} \models S \sqsubseteq R$, and $\mathcal{O}$ an OWL ontology adhering to the syntax and semantics as specified in OWL 2 Standard, check whether:

Test 1. $\mathcal{O} \models D_S \sqsubseteq D_R$ and $\mathcal{O} \models R_S \sqsubseteq R_R$;
Test 2. $\mathcal{O} \not\models D_R \sqsubseteq D_S$;
Test 3. $\mathcal{O} \not\models R_R \sqsubseteq R_S$;
Test 4. If $\mathcal{O} \models \text{Asym}(R)$ then $\mathcal{O} \models \text{Asym}(S)$;
Test 5. If $\mathcal{O} \models \text{Sym}(R)$ then $\mathcal{O} \models \text{Sym}(S)$ or $\mathcal{O} \models \text{Asym}(S)$;
Test 6. If $\mathcal{O} \models \text{Trans}(R)$ then $\mathcal{O} \models \text{Trans}(S)$;
Test 7. If $\mathcal{O} \models \text{Ref}(R)$ then $\mathcal{O} \models \text{Ref}(S)$ or $\mathcal{O} \not\models \text{Irr}(S)$;
Test 8. If $\mathcal{O} \models \text{Irr}(R)$ then $\mathcal{O} \models \text{Irr}(S)$ or $\mathcal{O} \models \text{Asym}(S)$;
Test 9. If $\mathcal{O} \models \text{Asym}(R)$ then $\mathcal{O} \not\models \text{Sym}(S)$;
Test 10. If $\mathcal{O} \models \text{Irr}(R)$ then $\mathcal{O} \not\models \text{Ref}(S)$; continues....
Definition (Sub-Property compatibility Service (SubProS))

... continued from previous page

Test 10. If $O \models \text{Irr}(R)$ then $O \not\models \text{Ref}(S)$;

Test 11. If $O \models \text{Trans}(R)$ then $O \not\models \text{Irr}(R)$, $O \not\models \text{Asym}(R)$, $O \not\models \text{Irr}(S)$, and $O \not\models \text{Asym}(S)$;

An OWL object property hierarchy is said to be compatible iff

- Test 1 and (2 or 3) hold for all pairs of property-subproperty in $O$, and
- Tests 4–11 hold for all pairs of property-subproperty in $O$. 

...continued on next page
Recall the three cases for property chains, with \( w \sqsubseteq R \):

- **Case S:** \( w = S_1 \circ \ldots \circ S_n \) and \( S_i \prec R \), for all \( 1 \geq i \geq n \), or

- **Case RS:** \( w = R \circ S_1 \circ \ldots \circ S_n \) and \( S_i \prec R \), for all \( 1 \geq i \geq n \), or

- **Case SR:** \( w = S_1 \circ \ldots \circ S_n \circ R \) and \( S_i \prec R \), for all \( 1 \geq i \geq n \).
Property chains

Recall the three cases for property chains, with $w \sqsubseteq R$:

- Case S: $w = S_1 \circ \ldots \circ S_n$ and $S_i \prec R$, for all $1 \geq i \geq n$, or
- Case RS: $w = R \circ S_1 \circ \ldots \circ S_n$ and $S_i \prec R$, for all $1 \geq i \geq n$, or
- Case SR: $w = S_1 \circ \ldots \circ S_n \circ R$ and $S_i \prec R$, for all $1 \geq i \geq n$.

To ensure avoidance of undesirable classifications or inconsistencies, informally:

- The domain/range class from left to right has to be equal or a superclass, on the lhs of the inclusion
- Similarly for the outer domain and range on the lhs and domain and range of the object property on the rhs
Definition (Property Chain Compatibility Service (ProChainS))

For each set of object properties, \( R, S_1, \ldots, S_n \in \mathcal{R} \), \( \mathcal{R} \) the set of OWL object properties (\( V_{OP} \) in OWL 2) in OWL ontology \( \mathcal{O} \), and \( S_i \prec R \) with \( 1 \leq i \leq n \), \( \mathcal{O} \) adheres to the constraints of Definition 2 (and, more generally, the OWL 2 specification), and user-defined domain and range axioms as defined in Definition 1, for each of the property chain expression, select either one of the three cases:

**Case S.** Property chain pattern as \( S_1 \circ S_2 \circ \ldots \circ S_n \subseteq R \). Test whether:

\[
\text{Test S-a. } \mathcal{O} \models R_{S_1} \subseteq D_{S_2}, \ldots, R_{S_{n-1}} \subseteq D_{S_n}; \\
\text{Test S-b. } \mathcal{O} \models D_{S_1} \subseteq D_R; \\
\text{Test S-c. } \mathcal{O} \models R_{S_n} \subseteq R_R;
\]

**Case RS.** Property chain pattern as \( R \circ S_1 \circ \ldots \circ S_n \subseteq R \). Test whether:

\[
\text{Test RS-a. } \mathcal{O} \models R_{S_1} \subseteq D_{S_2}, \ldots, R_{S_{n-1}} \subseteq D_{S_n}; \\
\text{Test RS-b. } \mathcal{O} \models R_R \subseteq D_{S_1}; \\
\text{Test RS-c. } \mathcal{O} \models R_{S_n} \subseteq R_R;
\]

**Case SR.** Property chain pattern as \( S_1 \circ \ldots \circ S_n \circ R \subseteq R \). Test whether:

\[
\text{Test SR-a. } \mathcal{O} \models R_{S_1} \subseteq D_{S_2}, \ldots, R_{S_{n-1}} \subseteq D_{S_n}; \\
\text{Test SR-b. } \mathcal{O} \models D_{S_1} \subseteq R_R; \\
\text{Test SR-c. } \mathcal{O} \models R_{S_n} \subseteq R_R;
\]

An OWL property chain expression is said to be compatible iff the OWL 2 syntactic constraints hold and either Case S, or Case RS, or Case SR holds.
Definition (Property Chain Compatibility Service (ProChainS))

.... continued from previous page

Case SR. Property chain pattern as \( S_1 \circ \ldots \circ S_n \circ R \sqsubseteq R \). Test whether:

Test SR-a. \( \mathcal{O} \models R_{S_1} \sqsubseteq D_{S_2}, \ldots, R_{S_{n-1}} \sqsubseteq D_{S_n} \);
Test SR-b. \( \mathcal{O} \models D_{S_1} \sqsubseteq D_R \);
Test SR-c. \( \mathcal{O} \models R_{S_n} \sqsubseteq D_R \);

An OWL property chain expression is said to be compatible iff the OWL 2 syntactic constraints hold and either Case S, or Case RS, or Case SR holds.
Does it matter?

or: How common are violations? which violations appear in ontologies ‘in the wild’?
Does it matter?

- or: How common are violations? which violations appear in ontologies ‘in the wild’?
- Evaluated against 15 ontologies that have many OPs
- Two examples on next slide
- Then a summary of selection of TONES Repository ontologies (d.d. 12-3-2012) on next slide
BioTop’s inconsistent ‘has process role’

‘has process role’ in BioTop [Beisswanger et al. (2008)] (v. June 17, 2010) is inconsistent. Relevant axioms are:

- ‘has process role’ ⊑ ‘temporally related to’  \(\text{(E.1)}\)
- ‘has process role’ ⊑ ‘processual entity’ × role \(\text{(E.2)}\)
- ‘temporally related to’ ⊑
- ‘processual entity’ ⊓ quality ×
- ‘processual entity’ ⊓ quality \(\text{(E.3)}\)
- role ⊑ ¬quality \(\text{(E.4)}\)
- role ⊑ ¬‘processual entity’ \(\text{(E.5)}\)
- Sym(‘temporally related to’) \(\text{(E.6)}\)
BioTop’s inconsistent ‘has process role’

Use *SubProS* to isolate the flaw:

- **Test 1**: fail, because $R_{\text{hasprocessrole}} \subseteq R_{\text{temporallyrelatedto}}$ is false, as the ranges (see E.2 cf. E.3) are disjoint (see E.4, E.5) and therewith ‘has process role’ is inconsistent;

- **Test 2** and **3**: pass.

- **Test 4**: not applicable.

- **Test 5**: fail, because $\emptyset$ does not contain Sym(‘has process role’).

- **Test 6–11**: not applicable.
Of type Case S. Test S–c (for corrections) failed because \( \mathcal{O} \not\models R_{\text{DM-Task} \sqsubseteq \text{OptimizationProblem}} \sqsubseteq R_{\text{DM-Task}} \). Considering the suggestions for revision, step B’s first option to revise the ontology was chosen, i.e., removing OptimizationProblem from the range axiom of addresses.
<table>
<thead>
<tr>
<th>Ontology</th>
<th>No. of OPs</th>
<th>No. of SubOPs axioms</th>
<th>No. more constrained by char.</th>
<th>Comments (partial)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOLCE-lite</td>
<td>70</td>
<td>46</td>
<td>13</td>
<td>transitivity added</td>
</tr>
<tr>
<td>SAO 1.2</td>
<td>36</td>
<td>25</td>
<td>21</td>
<td>2 x transitivity; Test 6 fails on has Vesicle Component</td>
</tr>
<tr>
<td>airsystem</td>
<td>111</td>
<td>56</td>
<td>43</td>
<td>imports DUL. ProChainS fails</td>
</tr>
<tr>
<td>family-tree</td>
<td>52</td>
<td>25</td>
<td>14</td>
<td>fails Test 6 of SubProS</td>
</tr>
<tr>
<td>propreo</td>
<td>32</td>
<td>20</td>
<td>17</td>
<td>beyond OWL 2 DL (non-simple prop. in max card.)</td>
</tr>
<tr>
<td>heart</td>
<td>29</td>
<td>18</td>
<td>9</td>
<td>many inconsistencies</td>
</tr>
<tr>
<td>mygrid-unclassified</td>
<td>69</td>
<td>39</td>
<td>0</td>
<td>1 x transitive added</td>
</tr>
<tr>
<td>building Architecture</td>
<td>28</td>
<td>24</td>
<td>0</td>
<td>imports rcc, fails Test 5 of SubProS (omission Asym on properPartOf)</td>
</tr>
<tr>
<td>biotop</td>
<td>89</td>
<td>84</td>
<td>45</td>
<td>with transitivity; ‘has process role’ is inconsistent (disjoint ranges), see Evaluation 1</td>
</tr>
</tbody>
</table>
Outline

1. Introduction

2. Semantics of relations
   - Positionalism
   - Hierarchies of relations

3. Some common relations
   - Part-whole relations
   - Mereotopology
   - Beyond parts and space

4. Modelling and reasoning
   - Reasoner-mediated modelling
   - Performance considerations
   - Hands-on

5. Recap
Don’t reinvent the wheel

- Part-whole relations, probably received most attention in ontologies
- Spatial relations, and its interaction with parthood
- Participation, constitution, causation, ...
- Similarity: important for combination machine learning with ontologies
Taxonomy of part-whole relations

- Hierarchy of part-whole relations common in ontologies and conceptual data models
- Uses DOLCE foundational ontology [Masolo et al.(2003)] for domain and range of a relation
- Main distinction between transitive (parthood) vs non-transitive (just meronymic) part-whole relations
- Formally defined
- Details in [Keet and Artale(2008)]
Part-whole relations

Part-whole relation

\[
\text{part\textsubscript{of}} \quad \text{mpart\textsubscript{of}}
\]

(\textit{Mereological part-of relation})

(\textit{(Meronymic) part-whole relation})

\[
\begin{align*}
\text{s-part-of} & \quad \text{spatial-part-of} & \quad \text{involved-in} & \quad \text{member-of} & \quad \text{constitutes} & \quad \text{sub-quantity-of} & \quad \text{participates-in} \\
\text{f-part-of} & \quad \text{contained-in} & \quad \text{located-in} & \quad \text{member-of'} & \quad & \quad & \\
\end{align*}
\]
Part-whole relations

“member-bunch”, collective nouns (e.g. Herd, Orchestra) with their members (Sheep, Musician)

$$\forall x, y (\text{member}_n(x, y) \triangleq \text{mpart}_n(x, y) \land (\text{POB}(x) \lor \text{SOB}(x))$$

“material-object”, that what something is made of (e.g., Vase and Clay)

$$\forall x, y (\text{constitutes}_i(x, y) \equiv \text{constituted}_i(y, x) \triangleq \text{mpart}_i(x, y) \land \text{POB}(y) \land M(x))$$
Part-whole relations

“quantity-mass”, “portion-object”, relating a smaller (or sub) part of an amount of matter to the whole. Two issues (glass of wine & bottle of wine vs. Salt as subquantity of SeaWater)

\[ \forall x, y (\text{sub\_quantity\_of}_n(x, y) \triangleq \text{mpart\_of}(x, y) \land M(x) \land M(y)) \]

“noun-feature/activity”, entity participates in a process, like Enzyme that participates in CatalyticReaction

\[ \forall x, y (\text{participates\_in}_it(x, y) \triangleq \text{mpart\_of}(x, y) \land ED(x) \land PD(y)) \]
Part-whole relations

processes and sub-processes (e.g. Chewing is involved in the grander process of Eating)

\[ \forall x, y (\text{involved in}(x, y) \triangleq \text{part of}(x, y) \land \text{PD}(x) \land \text{PD}(y)) \]

Object and its 2D or 3D region, such as contained in(John’s address book, John’s bag) and located in(Pretoria, South Africa)

\[ \forall x, y (\text{contained in}(x, y) \triangleq \text{part of}(x, y) \land \text{R}(x) \land \text{R}(y) \land \exists z, w (\text{has 3D}(z, x) \land \text{has 3D}(w, y) \land \text{ED}(z) \land \text{ED}(w))) \]

\[ \forall x, y (\text{located in}(x, y) \triangleq \text{part of}(x, y) \land \text{R}(x) \land \text{R}(y) \land \exists z, w (\text{has 2D}(z, x) \land \text{has 2D}(w, y) \land \text{ED}(z) \land \text{ED}(w))) \]

\[ \forall x, y (\text{s part of}(x, y) \triangleq \text{part of}(x, y) \land \text{ED}(x) \land \text{ED}(y)) \]
Knowledge and Google & AfriGIS
Knowledge and Google & AfriGIS

- How can we represent
  - The Kruger Park *overlaps* with South Africa
  - Durban is a *tangential proper part* of South Africa
  - Gauteng is a *non-tangential proper part* of South Africa
  - Botswana is *connected to* South Africa (do they *share* a border?)
  - Lesotho is *spatially located within* the area of South Africa (but not part of)?
Knowledge and Google & AfriGIS

- How can we represent
  - The Kruger Park overlaps with South Africa
  - Durban is a tangential proper part of South Africa
  - Gauteng is a non-tangential proper part of South Africa
  - Botswana is connected to South Africa (do they share a border?)
  - Lesotho is spatially located within the area of South Africa (but not part of)?

- Can we do all that with mereology? Use only spatial relations? Combining mereo+spatial?
Parts and space

- Could not represent all of parthood in OWL or any DL, worse for mereotopology, but tried anyway [Keet et al. (2012)]
- Example:
  - Let NTPLI be a ‘non-tangential proper located in’ relation
  - EnclosedCountry ≡ Country \( \sqcap \exists \text{NTPLI}.\text{Country} \)
  - NTPLI(Lesotho, South Africa), Country(Lesotho), Country(South Africa),
  - then it will correctly deduce EnclosedCountry(Lesotho).
  - with merely ‘part-of’, one would not have been able to obtain this result
• 9-Intersection Method (9IM), based on point-set topology [Egenhofer and Herring(1990)]

• Region Connection Calculus (RCC), based on the reflexive and symmetric connection [Randell et al.(1992)]

• Neither one considers the combination of the space region with the object that occupies it

• This interaction is addressed by mereotopology, which focuses on spatial entities, not just regions.
Options to merging parts and locations

How to combine them? Concerning primitive relations [Cohn and Renz(2008), Varzi(2007)], one can

- define parthood, $P$, in terms of connection, $C$, (i.e., $P(x, y) =_{def} \forall z (C(z, x) \rightarrow C(z, y))$) so that topology is principal and mereology a subtheory

- introduce topology as a sub-domain of mereology by introducing a sorted predicate to denote region ($R$) and define $C$ in terms of overlapping regions
  $$C(x, y) =_{def} O(x, y) \land R(x) \land R(y)$$
  [Eschenbach and Heydrich(1995)]
Options to merging parts and locations

How to combine them? Concerning primitive relations [Cohn and Renz(2008), Varzi(2007)], one can

- define parthood, $P$, in terms of connection, $C$, (i.e., $P(x, y) = \text{def } \forall z (C(z, x) \rightarrow C(z, y))$) so that topology is principal and mereology a subtheory

- introduce topology as a sub-domain of mereology by introducing a sorted predicate to denote region ($R$) and define $C$ in terms of overlapping regions $(C(x, y) = \text{def } O(x, y) \land R(x) \land R(y))$ [Eschenbach and Heydrich(1995)]

- consider both $P$ and $C$ as primitive

- introduce a ternary relation $CP(x, y, z)$, so that $P(x, y) = \text{def } \exists z CP(x, z, y)$ and $C(x, y) = \text{def } \exists z CP(x, y, z)$
Kuratowski extension of GEMT (KGEMT)

Kuratowski axioms for topological closure (inclusion, idempotence, and additivity), therewith a full account of intended interpretation of connection [Varzi(2007)].
Ground Topology

<table>
<thead>
<tr>
<th>Core axioms and definitions</th>
<th>(P(x, x)) (t1)</th>
<th>(P(x, y) \land P(y, z) \rightarrow P(x, z)) (t2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P(x, y) \land P(y, x) \rightarrow x = y) (t3)</td>
<td>(\neg P(y, x) \rightarrow \exists z (P(z, y) \land \neg O(z, x))) (t4)</td>
<td></td>
</tr>
<tr>
<td>(\exists w \phi(w) \rightarrow \exists z \forall w (O(w, z) \leftrightarrow \exists v (\phi(v) \land O(w, v)))) (t5)</td>
<td>(\exists w \phi(w) \rightarrow \exists z \forall w (O(w, z) \leftrightarrow \exists v (\phi(v) \land O(w, v)))) (t5)</td>
<td></td>
</tr>
<tr>
<td>(C(x, x)) (t6)</td>
<td>(C(x, y) \rightarrow C(y, x)) (t7)</td>
<td></td>
</tr>
<tr>
<td>(P(x, y) \rightarrow E(x, y)) (t8)</td>
<td>(E(x, y) =_{df} \forall z (C(z, x) \rightarrow C(z, y))) (t9)</td>
<td></td>
</tr>
<tr>
<td>(E(x, y) \rightarrow P(x, y)) (t10)</td>
<td>(\neg \exists w \phi(w) \rightarrow \exists w (\phi(w) \land \neg O(w, x))) (t11)</td>
<td></td>
</tr>
<tr>
<td>(O(x, y) =_{df} \exists z (P(z, x) \land O(z, y) \land \forall w (P(w, z) \rightarrow (O(w, x) \lor O(w, y)))) \rightarrow C(x, y)) (t12)</td>
<td>(O(x, y) =_{df} \exists z (P(z, x) \land O(z, y) \land \forall w (P(w, z) \rightarrow (O(w, x) \lor O(w, y)))) \rightarrow C(x, y)) (t12)</td>
<td></td>
</tr>
<tr>
<td>(z = \sum x \phi x \rightarrow \forall y (C(y, z) \rightarrow \exists x (\phi x \land C(y, x)))) (t13)</td>
<td>(z = \sum x \phi x \rightarrow \forall y (C(y, z) \rightarrow \exists x (\phi x \land C(y, x)))) (t13)</td>
<td></td>
</tr>
<tr>
<td>(P(x, cx)) (t14)</td>
<td>(c(cx) = cx) (t15)</td>
<td></td>
</tr>
<tr>
<td>(c(x + y) = cx + cy) (t16)</td>
<td>(cx =_{df} \sim (ex)) (t17)</td>
<td></td>
</tr>
<tr>
<td>(ex =_{df} i(\sim x)) (t18)</td>
<td>(ix =_{df} \sum z \forall y (C(z, y) \rightarrow O(x, y))) (t19)</td>
<td></td>
</tr>
</tbody>
</table>

Additional axioms, definitions, and theorems

| PP(x, y) =_{df} P(x, y) \land \neg P(y, x) (t20) | O(x, y) =_{df} \exists z (P(z, x) \land P(z, y)) (t21) |
| EQ(x, y) =_{df} P(x, y) \land P(y, x) (t22) | TP(x, y) =_{df} PP(x, y) \land \neg IPP(x, y) (t23) |
| IPP(x, y) =_{df} PP(x, y) \land \forall z (C(z, x) \rightarrow O(z, y)) (t24) | \(\neg PP(x, x)\) (t25) |
| \(PP(x, y) \rightarrow \neg PP(y, x)\) (t27) | \(PP(x, y) \land PP(y, z) \rightarrow PP(x, z)\) (t26) |
### Minimal (mereo) Topology

#### Core axioms and definitions

<table>
<thead>
<tr>
<th>Formula</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P(x, x) )</td>
<td>( P ) is reflexive</td>
<td>(t1)</td>
</tr>
<tr>
<td>( P(x, y) \land P(y, x) \rightarrow x = y )</td>
<td>( P ) is symmetric</td>
<td>(t3)</td>
</tr>
<tr>
<td>( \exists w \phi(w) \rightarrow \exists z \forall w \left( O(w, z) \leftrightarrow \exists v (\phi(v) \land O(w, v)) \right) )</td>
<td></td>
<td>(t5)</td>
</tr>
<tr>
<td>( C(x, x) )</td>
<td>( C ) is reflexive</td>
<td>(t6)</td>
</tr>
<tr>
<td>( P(x, y) \rightarrow E(x, y) )</td>
<td>( E ) is reflexive</td>
<td>(t8)</td>
</tr>
<tr>
<td>( E(x, y) \rightarrow P(x, y) )</td>
<td></td>
<td>(t10)</td>
</tr>
<tr>
<td>( \exists z \left( SC(z) \land O(z, x) \land O(z, y) \land \forall w (P(w, z) \rightarrow (O(w, x) \lor O(w, y))) \right) \rightarrow C(x, y) )</td>
<td></td>
<td>(t12)</td>
</tr>
<tr>
<td>( z = \sum x \phi(x) \rightarrow \forall y \left( C(y, z) \rightarrow \exists x (\phi(x) \land C(y, x)) \right) )</td>
<td></td>
<td>(t13)</td>
</tr>
<tr>
<td>( c(cx) = cx )</td>
<td>( c ) is distributive over ( P )</td>
<td>(t15)</td>
</tr>
<tr>
<td>( \exists z (SC(z) \land O(z, x) \land O(z, y) \land \forall w (P(w, z) \rightarrow (O(w, x) \lor O(w, y))) \rightarrow C(x, y) )</td>
<td></td>
<td>(t21)</td>
</tr>
<tr>
<td>( TPP(x, y) = df \ P(x, y) \land \neg IPP(x, y) )</td>
<td>( TPP ) is transitive</td>
<td>(t23)</td>
</tr>
<tr>
<td>( IPP(x, y) = df \ P(x, y) \land \forall z (C(z, x) \rightarrow O(z, y)) )</td>
<td></td>
<td>(t24)</td>
</tr>
</tbody>
</table>

#### Additional axioms, definitions, and theorems

<table>
<thead>
<tr>
<th>Formula</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( PP(x, y) = df \ P(x, y) \land \neg P(y, x) )</td>
<td>( PP ) is irreflexive</td>
<td>(t20)</td>
</tr>
<tr>
<td>( EQ(x, y) = df \ P(x, y) \land P(y, x) )</td>
<td>( EQ ) is irreflexive</td>
<td>(t22)</td>
</tr>
<tr>
<td>( IPP(x, y) = df \ PP(x, y) \land \forall z (C(z, x) \rightarrow O(z, y)) )</td>
<td></td>
<td>(t24)</td>
</tr>
<tr>
<td>( \neg PP(x, x) )</td>
<td></td>
<td>(t25)</td>
</tr>
<tr>
<td>( PP(x, y) \rightarrow \neg PP(y, x) )</td>
<td></td>
<td>(t27)</td>
</tr>
</tbody>
</table>
## Minimal (mereo) Topology; Ground Mereology

### Core axioms and definitions

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(x, x)$</td>
<td>(t1)</td>
</tr>
<tr>
<td>$P(x, y) \land P(y, x) \rightarrow x = y$</td>
<td>(t3)</td>
</tr>
<tr>
<td>$P(x, y) \land P(y, z) \rightarrow P(x, z)$</td>
<td>(t2)</td>
</tr>
<tr>
<td>$\exists w \phi(w) \rightarrow \exists z \forall w(\neg O(w, z) \leftrightarrow \exists \nu(\phi(\nu) \land O(w, \nu)))$</td>
<td>(t5)</td>
</tr>
<tr>
<td>$C(x, x)$</td>
<td>(t6)</td>
</tr>
<tr>
<td>$C(x, y) \rightarrow C(y, x)$</td>
<td>(t7)</td>
</tr>
<tr>
<td>$P(x, y) \rightarrow E(x, y)$</td>
<td>(t8)</td>
</tr>
<tr>
<td>$E(x, y) = \exists z(C(z, x) \rightarrow C(z, y))$</td>
<td>(t9)</td>
</tr>
<tr>
<td>$E(x, y) \rightarrow P(x, y)$</td>
<td>(t10)</td>
</tr>
<tr>
<td>$\exists z(SC(z) \land O(z, x) \land O(z, y) \land \forall w(P(w, z) \rightarrow (O(w, x) \lor O(w, y)))) \rightarrow C(x, y)$</td>
<td>(t12)</td>
</tr>
<tr>
<td>$\exists z(\neg O(z, x) \land \forall w(P(w, z) \rightarrow (O(w, x) \lor O(w, y)))) \rightarrow C(x, y)$</td>
<td>(t13)</td>
</tr>
<tr>
<td>$P(x, cx)$</td>
<td>(t14)</td>
</tr>
<tr>
<td>$c(x + y) = cx + cy$</td>
<td>(t16)</td>
</tr>
<tr>
<td>$ex = df \sum x \phi x$</td>
<td>(t18)</td>
</tr>
<tr>
<td>$c(cx) = cx$</td>
<td>(t15)</td>
</tr>
<tr>
<td>$cx = df \exists z(\phi x \land C(z, x))$</td>
<td>(t17)</td>
</tr>
</tbody>
</table>

### Additional axioms, definitions, and theorems

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PP(x, y) = df P(x, y) \land \neg P(y, x)$</td>
<td>(t20)</td>
</tr>
<tr>
<td>$EQ(x, y) = df P(x, y) \land P(y, x)$</td>
<td>(t22)</td>
</tr>
<tr>
<td>$IPP(x, y) = df PP(x, y) \land \forall z(C(z, x) \rightarrow O(z, y))$</td>
<td>(t24)</td>
</tr>
<tr>
<td>$\neg PP(x, x)$</td>
<td>(t25)</td>
</tr>
<tr>
<td>$PP(x, y) \rightarrow \neg PP(y, x)$</td>
<td>(t27)</td>
</tr>
<tr>
<td>$O(x, y) = df \exists z(P(z, x) \land P(z, y))$</td>
<td>(t21)</td>
</tr>
<tr>
<td>$TPP(x, y) = df PP(x, y) \land \neg IPP(x, y)$</td>
<td>(t23)</td>
</tr>
<tr>
<td>$PP(x, y) \land PP(y, z) \rightarrow PP(x, z)$</td>
<td>(t26)</td>
</tr>
</tbody>
</table>
**Minimal (mereo) Topology; General Extensional Mereology**

### Core axioms and definitions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Annotation</th>
<th>Expression</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(x, x)$</td>
<td>(t1)</td>
<td>$P(x, y) \land P(y, z) \rightarrow P(x, z)$</td>
<td>(t2)</td>
</tr>
<tr>
<td>$P(x, y) \land P(y, x) \rightarrow x = y$</td>
<td>(t3)</td>
<td>$\neg P(y, x) \rightarrow \exists z (P(z, y) \land \neg O(z, x))$</td>
<td>(t4)</td>
</tr>
<tr>
<td>$\exists w \phi(w) \rightarrow \exists z\forall w (O(w, z) \leftrightarrow \exists v (\phi(v) \land O(w, v)))$</td>
<td>(t5)</td>
<td>$C(x, y) \rightarrow C(y, x)$</td>
<td>(t7)</td>
</tr>
<tr>
<td>$P(x, y) \rightarrow E(x, y)$</td>
<td>(t8)</td>
<td>$E(x, y) =_{df} \forall z (C(z, x) \rightarrow C(z, y))$</td>
<td>(t9)</td>
</tr>
<tr>
<td>$E(x, y) \rightarrow P(x, y)$</td>
<td>(t10)</td>
<td>$SC(x) = \forall y, z(x = y + z \rightarrow C(y, z))$</td>
<td>(t11)</td>
</tr>
<tr>
<td>$\exists z (SC(z) \land O(z, x) \land O(z, y) \land \forall w (P(w, z) \rightarrow (O(w, x) \lor O(w, y)))) \rightarrow C(x, y)$</td>
<td>(t12)</td>
<td>$z = \sum x \phi x \rightarrow \forall y (C(y, z) \rightarrow \exists x (\phi x \land C(y, x)))$</td>
<td>(t13)</td>
</tr>
<tr>
<td>$P(x, cx)$</td>
<td>(t14)</td>
<td>$c(cx) = cx$</td>
<td>(t15)</td>
</tr>
<tr>
<td>$c(x + y) = cx + cy$</td>
<td>(t16)</td>
<td>$cx =_{df} (ex)$</td>
<td>(t17)</td>
</tr>
<tr>
<td>$ex =_{df} i(\sim x)$</td>
<td>(t18)</td>
<td>$ix =_{df} \sum z \forall y (C(z, y) \rightarrow O(x, y))$</td>
<td>(t19)</td>
</tr>
</tbody>
</table>

### Additional axioms, definitions, and theorems

<table>
<thead>
<tr>
<th>Expression</th>
<th>Annotation</th>
<th>Expression</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PP(x, y) =_{df} P(x, y) \land \neg P(y, x)$</td>
<td>(t20)</td>
<td>$O(x, y) =_{df} \exists z (P(z, x) \land P(z, y))$</td>
<td>(t21)</td>
</tr>
<tr>
<td>$EQ(x, y) =_{df} P(x, y) \land P(y, x)$</td>
<td>(t22)</td>
<td>$TPP(x, y) =_{df} PP(x, y) \land \neg IPP(x, y)$</td>
<td>(t23)</td>
</tr>
<tr>
<td>$IPP(x, y) =_{df} PP(x, y) \land \forall z (C(z, x) \rightarrow O(z, y))$</td>
<td>(t24)</td>
<td>$\neg PP(x, x)$</td>
<td>(t25)</td>
</tr>
<tr>
<td>$PP(x, y) \rightarrow \neg PP(y, x)$</td>
<td>(t27)</td>
<td>$PP(x, y) \land PP(y, z) \rightarrow PP(x, z)$</td>
<td>(t26)</td>
</tr>
</tbody>
</table>
### General Extensional Mereotopology

#### Core axioms and definitions

<table>
<thead>
<tr>
<th>Axiom/Definition</th>
<th>(t1)</th>
<th>Axiom/Definition</th>
<th>(t2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P(x, x) )</td>
<td></td>
<td>( P(x, y) \land P(y, z) \rightarrow P(x, z) )</td>
<td></td>
</tr>
<tr>
<td>( P(x, y) \land P(y, x) \rightarrow x = y )</td>
<td>(t3)</td>
<td>( \neg P(y, x) \rightarrow \exists z(P(z, y) \land \neg O(z, x)) )</td>
<td>(t4)</td>
</tr>
<tr>
<td>( \exists w \phi(w) \rightarrow \exists z \forall w(O(w, z) \leftrightarrow \exists v(\phi(v) \land O(w, v))) )</td>
<td>(t5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( C(x, x) )</td>
<td>(t6)</td>
<td>( C(x, y) \rightarrow C(y, x) )</td>
<td>(t7)</td>
</tr>
<tr>
<td>( P(x, y) \rightarrow E(x, y) )</td>
<td>(t8)</td>
<td>( E(x, y) =_{df} \forall z(C(z, x) \rightarrow C(z, y)) )</td>
<td>(t9)</td>
</tr>
<tr>
<td>( E(x, y) \rightarrow P(x, y) )</td>
<td>(t10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \exists z(SC(z) \land O(z, x) \land O(z, y) \land \forall w(P(w, z) \rightarrow (O(w, x) \lor O(w, y))) \rightarrow C(x, y) )</td>
<td>(t12)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( z = \sum x \phi(x) \rightarrow \forall y(C(y, z) \rightarrow \exists x(\phi(x) \land C(y, x))) )</td>
<td>(t13)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( P(x, cx) \) \hspace{1cm} \( c(cx) = cx \) \hspace{1cm} (t14)

\( c(x + y) = cx + cy \) \hspace{1cm} \( cx =_{df} \sim (ex) \) \hspace{1cm} (t16)

\( ex =_{df} i(\sim x) \) \hspace{1cm} \( ix =_{df} \sum z \forall y(C(z, y) \rightarrow O(x, y)) \) \hspace{1cm} (t18)

#### Additional axioms, definitions, and theorems

<table>
<thead>
<tr>
<th>Axiom/Definition</th>
<th>(t20)</th>
<th>Axiom/Definition</th>
<th>(t21)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( PP(x, y) =_{df} P(x, y) \land \neg P(y, x) )</td>
<td></td>
<td>( O(x, y) =_{df} \exists z(P(z, x) \land P(z, y)) )</td>
<td></td>
</tr>
<tr>
<td>( EQ(x, y) =_{df} P(x, y) \land P(y, x) )</td>
<td>(t22)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( IPP(x, y) =_{df} PP(x, y) \land \forall z(C(z, x) \rightarrow O(z, y)) )</td>
<td></td>
<td>( TPP(x, y) =_{df} PP(x, y) \land \neg IPP(x, y) )</td>
<td></td>
</tr>
<tr>
<td>( \neg PP(x, x) )</td>
<td>(t25)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( PP(x, y) \rightarrow \neg PP(y, x) )</td>
<td>(t27)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Kuratowski General Extensional Mereotopology

<table>
<thead>
<tr>
<th>Core axioms and definitions</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( P(x, x) )</td>
<td>(t1)</td>
<td>( P(x, y) \land P(y, z) \to P(x, z) )</td>
</tr>
<tr>
<td>( P(x, y) \land P(y, x) \to x = y )</td>
<td>(t3)</td>
<td>( \neg P(y, x) \to \exists z(P(z, y) \land \neg O(z, x)) )</td>
</tr>
<tr>
<td>( \exists w \phi(w) \to \exists z \forall w (O(w, z) \leftrightarrow \exists v(\phi(v) \land O(w, v))) )</td>
<td>(t5)</td>
<td></td>
</tr>
<tr>
<td>( C(x, x) )</td>
<td>(t6)</td>
<td>( C(x, y) \to C(y, x) )</td>
</tr>
<tr>
<td>( P(x, y) \to E(x, y) )</td>
<td>(t8)</td>
<td>( E(x, y) =_{df} \forall z(C(z, x) \to C(z, y)) )</td>
</tr>
<tr>
<td>( E(x, y) \to P(x, y) )</td>
<td>(t10)</td>
<td>( SC(x) \leftrightarrow \forall y, z(x = y + z \to C(y, z)) )</td>
</tr>
<tr>
<td>( \exists z(SC(z) \land O(z, x) \land O(z, y) \land \forall w(P(w, z) \to (O(w, x) \lor O(w, y)))) \to C(x, y) )</td>
<td>(t12)</td>
<td></td>
</tr>
<tr>
<td>( z = \sum x \phi x \to \forall y(C(y, z) \rightarrow \exists x(\phi x \land C(y, x))) )</td>
<td>(t13)</td>
<td></td>
</tr>
<tr>
<td>( P(x, cx) )</td>
<td>(t14)</td>
<td>( c(cx) = cx )</td>
</tr>
<tr>
<td>( c(x + y) = cx + cy )</td>
<td>(t16)</td>
<td>( cx =_{df} \sim (ex) )</td>
</tr>
<tr>
<td>( ex =_{df} i(\sim x) )</td>
<td>(t18)</td>
<td>( ix =_{df} \sum z \forall y(C(z, y) \to O(x, y)) )</td>
</tr>
</tbody>
</table>

### Additional axioms, definitions, and theorems

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( PP(x, y) =_{df} P(x, y) \land \neg P(y, x) )</td>
<td>(t20)</td>
<td>( O(x, y) =_{df} \exists z(P(z, x) \land P(z, y)) )</td>
</tr>
<tr>
<td>( EQ(x, y) =_{df} P(x, y) \land P(y, x) )</td>
<td>(t22)</td>
<td>( TPP(x, y) =_{df} PP(x, y) \land \neg IPP(x, y) )</td>
</tr>
<tr>
<td>( IPP(x, y) =_{df} PP(x, y) \land \forall z(C(z, x) \rightarrow O(z, y)) )</td>
<td>(t24)</td>
<td></td>
</tr>
<tr>
<td>( \neg PP(x, x) )</td>
<td>(t25)</td>
<td>( PP(x, y) \land PP(y, z) \rightarrow PP(x, z) )</td>
</tr>
<tr>
<td>( PP(x, y) \to \neg PP(y, x) )</td>
<td>(t27)</td>
<td></td>
</tr>
</tbody>
</table>
\[ \forall x, y (ECI(x, y) \equiv CI(x, y) \land P(y, x)) \]
\[ \forall x, y (PCI(x, y) \equiv PPO(x, y) \land R(x) \land R(y) \land \exists z, w (has\_3D(z, x) \land has\_3D(w, y) \land ED(z) \land ED(w))) \]
\[ \forall x, y (NTPCI(x, y) \equiv PCI(x, y) \land \forall z (C(z, x) \rightarrow O(z, y))) \]
\[ \forall x, y (TPCI(x, y) \equiv PCI(x, y) \land \neg NTPCI(x, y)) \]
\[ \forall x, y (ELI(x, y) \equiv LI(x, y) \land P(y, x)) \]
\[ \forall x, y (PLI(x, y) \equiv PPO(x, y) \land R(x) \land R(y) \land \exists z, w (has\_2D(z, x) \land has\_2D(w, y) \land ED(z) \land ED(w))) \]
\[ \forall x, y (NTPLI(x, y) \equiv PLI(x, y) \land \forall z (C(z, x) \rightarrow O(z, y))) \]
\[ \forall x, y (TPLI(x, y) \equiv PLI(x, y) \land \neg NTPLI(x, y)) \]
<table>
<thead>
<tr>
<th>Introduction</th>
<th>Semantics of relations</th>
<th>Some common relations</th>
<th>Modelling and reasoning</th>
<th>Recap</th>
</tr>
</thead>
</table>

\[
\forall x, y(ECI(x, y) \equiv CI(x, y) \land P(y, x)
\]

\[
\forall x, y(PCI(x, y) \equiv PPO(x, y) \land R(x) \land R(y) \land \exists z, w(has\_3D(z, x) \land has\_3D(w, y) \land ED(z) \land ED(w)))
\]

\[
\forall x, y(NTPCI(x, y) \equiv PCI(x, y) \land \forall z(C(z, x) \rightarrow O(z, y)))
\]

\[
\forall x, y(TPCI(x, y) \equiv PCI(x, y) \land \neg NTPCI(x, y))
\]

\[
\forall x, y(ELI(x, y) \equiv LI(x, y) \land P(y, x)
\]

\[
\forall x, y(PLI(x, y) \equiv PPO(x, y) \land R(x) \land R(y) \land \exists z, w(has\_2D(z, x) \land has\_2D(w, y) \land ED(z) \land ED(w)))
\]

\[
\forall x, y(NTPLI(x, y) \equiv PLI(x, y) \land \forall z(C(z, x) \rightarrow O(z, y)))
\]

\[
\forall x, y(TPLI(x, y) \equiv PLI(x, y) \land \neg NTPLI(x, y))
\]
∀x, y(\(ECI(x, y)\)) \equiv CI(x, y) \land P(y, x)

∀x, y(\(PCI(x, y)\)) \equiv PPO(x, y) \land R(x) \land R(y) \land \exists z, w(\text{has}_3D(z, x) \land \text{has}_3D(w, y) \land ED(z) \land ED(w))

∀x, y(\(NTPCI(x, y)\)) \equiv PCI(x, y) \land \forall z(C(z, x) \rightarrow O(z, y))

∀x, y(\(TPCI(x, y)\)) \equiv PCI(x, y) \land \neg NTPCI(x, y)

∀x, y(\(ELI(x, y)\)) \equiv LI(x, y) \land P(y, x)

∀x, y(\(PLI(x, y)\)) \equiv PPO(x, y) \land R(x) \land R(y) \land \exists z, w(\text{has}_2D(z, x) \land \text{has}_2D(w, y) \land ED(z) \land ED(w))

∀x, y(\(NTPLI(x, y)\)) \equiv PLI(x, y) \land \forall z(C(z, x) \rightarrow O(z, y))

∀x, y(\(TPLI(x, y)\)) \equiv PLI(x, y) \land \neg NTPLI(x, y)
Integrate the extension

Part-whole relation

proper-part-of

proper-s-part-of

s-part-of

proper-s-part-of

spatial-part-of

Spatial relation

involved-in

proper-involved-in

Located-in

proper-located-in

Tangential

nontangential

Equal

tangential-proper-contained-in

nontangential-proper-contained-in

same

equal-contained-in

equal-contained-in

equal-contained-in

equal-contained-in

equal-contained-in

Introductory

Semantics of relations

Some common relations

Modelling and reasoning

Recap
Subsets of KGEMT that can be represented in OWL

- Reason of differences: the object property characteristics (e.g. \( t_1/t_6 = \text{ref. of } P/C, t_{25} = \text{irr. of } PP, t_2 = \text{trans.} \)).
- The six definitions (PP, O, TPP, etc.) can be simplified and added as primitives to each one.

<table>
<thead>
<tr>
<th>OWL species</th>
<th>Subsets of KGEMT axioms</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWL 2 DL</td>
<td>(t1, t2, t6, t7, t8, t10, t26) or (t1, t2, t6, t7, t8, t10, t27) or (t1, t2, t6, t7, t8, t10, t25)</td>
</tr>
<tr>
<td>OWL DL</td>
<td>t2, t7, t8, t10, t26</td>
</tr>
<tr>
<td>OWL Lite</td>
<td>t2, t7, t8, t10, t26</td>
</tr>
<tr>
<td>OWL 2 RL</td>
<td>t2, t7, t8, t10, t26</td>
</tr>
<tr>
<td>OWL 2 EL</td>
<td>t1, t2, t6, t8, t10, t26</td>
</tr>
<tr>
<td>OWL 2 QL</td>
<td>t1, t6, t7, t8, t10</td>
</tr>
</tbody>
</table>

- Importance depends on the desired inference scenarios; thus far, Trans, Sym, Asym, and Irr seem to be more interesting, i.e., giving precedence to OWL 2 DL and OWL 2 RL (See [Keet et al.(2012)] for details on reasoning trade-offs)
Other relations in (foundational) ontologies

- Relation Ontology [Smith et al.(2005)]
- Relations that are sort-of temporal, but now not used as such; hence, one cannot reason ‘fully’ with them w.r.t. intended meaning
  - e.g.: derived-from, transformation-of
- dependence, inherence
- Attributes
- DOLCE’s qualities
Some other aspects of relations (not covered now)

- constraints on participation (essential vs. immutable vs. mandatory)
- Modality, necessity, telic, atelic
- Temporal relations, relation migration
- $n$-ary relations and reifying (objectifying) them
<table>
<thead>
<tr>
<th>Introduction</th>
<th>Semantics of relations</th>
<th>Some common relations</th>
<th>Modelling and reasoning</th>
<th>Recap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Outline**

1. **Introduction**

2. **Semantics of relations**
   - Positionalism
   - Hierarchies of relations

3. **Some common relations**
   - Part-whole relations
   - Mereotopology
   - Beyond parts and space

4. **Modelling and reasoning**
   - Reasoner-mediated modelling
   - Performance considerations
   - Hands-on

5. **Recap**
Any suggestions for actual ontology development?

- Using the taxonomy of part-whole relations
- Reasoner-guided relation selection
- Performance tradeoffs with inverses
Using the taxonomy of part-whole relations

- Representing it correctly in ontologies and conceptual data models

- Reasoning with a taxonomy of relations
Using the taxonomy of part-whole relations

- Representing it correctly in ontologies and conceptual data models
  - Decision diagram
  - Using the categories of the foundational ontology
  - Examples
    - *Software application* that simplifies all that: OntoParts [Keet et al. (2012)] and OntoParts-2 [Keet et al. (2013b)]
- Reasoning with a taxonomy of relations
Using the taxonomy of part-whole relations

- Representing it correctly in ontologies and conceptual data models
  - Decision diagram
  - Using the categories of the foundational ontology
  - Examples
    - *Software application* that simplifies all that: **OntoParts** [Keet et al.(2012)] and **OntoParts-2** [Keet et al.(2013b)]
- Reasoning with a taxonomy of relations
  - The **RBox reasoning service** [Keet and Artale(2008)] or **SubProS** [Keet(2014)] to pinpoint errors
GENERATOR: Guided ENtity reuse and class Expression geneRATOR [Keet et al.(2013a)]
GENERATOR with FORZA

- FORZA: Foundational Ontology and Reasoner-enhanced axiomatisation [Keet et al.(2013b)]
- Automated support for the linking with DOLCE categories
- Novel decision tree to categorise a subject domain class as a subclass of a DOLCE class (named D3)
- Novel algorithm that uses an automated reasoner to compute the applicable part-whole relation(s) between the selected classes (named OntoPartS-2)
- Avoids the common post-hoc checking, uses the reasoner to guide the ‘trial’ phase and reduce errors
FORZA implementation

- OntoParts-2 (jar file)
- D3 as XML file
- Integrated in MoKI modelling wiki [Ghidini et al.(2009)]
- sourceforge.net/projects/cikmontology/files/CIKM2013.zip/download
### Example (1/3)

#### Part: tour:Room

#### Whole: tour:Hotel

#### Dolce Category

#### Select the direction of Relation
- Part to Whole
- Whole to Part
- Both directions

#### QA Support

**Choose one of these options to proceed further**

- Is [Room] something that is happening or occurring?
- Is [Room] wholly present at any time of its existence?
- Does [Room] exist neither in space nor in time or does so because some other items that are not among its parts occupy that region?
- Is [Room] something that can be perceived or measured (like color, size, smell, etc.,)?

---

**References:**
- owl:Thing
- DOLCE:Particular
- DOLCE:Abstract
- DOLCE:Fact
- DOLCE:Set
- DOLCE:Region
Example (2/3)
Example (3/3)
Effects of features on reasoning

- Disjoint OPs, reflexivity, and qualified cardinality only on *simple* OPs in OWL 2. with *non-simple* when:
  - if $O$ contains an axiom $S \circ T \sqsubseteq R$
  - if $R$ is non-simple, then so is its inverse $R^{-}$
  - if $R$ is non-simple and $O$ contains any of the axioms $R \sqsubseteq S$, $S \equiv R$ or $R \equiv S$, then $S$ is also non-simple

- Domain and range axioms

- Role hierarchy with domain and range axioms vs. ‘specialising’ in class axioms (with existential) [Hammar(2014)]

- Inverses (next slide)

- ‘understanding’ the reasoner, predicting performance a hot topic; e.g. [Goncalves et al.(2012), Kang et al.(2012)]
Inverses

- Unsurprising (?) surprising reasoner performance with DMOP ontology [www.dmo-foundry.org]
Inverses

- Unsurprising (?) surprising reasoner performance with DMOP ontology [www.dmo-foundry.org]
- OWL 2 “new feature”:
  - `ObjectInverseOf(OP)` instead of only `InverseObjectProperties(OPE1 OPE2)` in OWL 1 for two object properties in the ontology
  - E.g., addresses with as inverse addressed by vs. addresses and using (in Protégé notation) `inverse(addresses)` in an axiom
- New feature slows down reasoner?
Inverses

- Unsurprising (?) surprising reasoner performance with DMOP ontology [www.dmo-foundry.org]
- OWL 2 “new feature”:
  - ObjectInverseOf(OP) instead of only InverseObjectProperties(OP1 OP2) in OWL 1 for two object properties in the ontology
  - E.g., addresses with as inverse addressed by vs. addresses and using (in Protégé notation) inverse(addresses) in an axiom
- New feature slows down reasoner?
- DMOP v5.4 with all InverseObjectProperties; test now by replacing those (n = 45) with ObjectInverseOf(OP) and compare
Inverses: performance results

Table: Classification times (in minutes) of DMOP and DMOP with ObjectInverseOf(). [Keet et al.(2014)]

<table>
<thead>
<tr>
<th>Component of classification</th>
<th>DMOP v5.4</th>
<th>DMOP v5.4 inverses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Hierarchy</td>
<td>6 mins</td>
<td>3 mins</td>
</tr>
<tr>
<td>Object Property Hierarchy</td>
<td>2 mins</td>
<td>1 min</td>
</tr>
<tr>
<td>Data Type Property Hierarchy</td>
<td>&lt;1 min</td>
<td>few secs</td>
</tr>
<tr>
<td>Class instances</td>
<td>about 1 min</td>
<td>&lt;1 min</td>
</tr>
</tbody>
</table>

The ObjectInverseOf() feature of OWL 2 improves the reasoner performance in the ODE by at least a third.
Choose one involvement between Chewing and Eating

- Chewing involved-in some Eating
  \[\text{Chewing} \sqsubseteq \exists \text{involves-in}.\text{Eating}\]
- Chewing inverse(involves) some Eating
  \[\text{Chewing} \sqsubseteq \exists \text{involves}.\text{Eating}\]
- Eating involves some Chewing
  \[\text{Eating} \sqsubseteq \exists \text{involves}.\text{Chewing}\]
- Eating inverse(involved-in) some Chewing
  \[\text{Eating} \sqsubseteq \exists \text{involved-in}.\text{Chewing}\]

(simplified notation online)
How to formalise the UML diagram in OWL?

- teaches, taught-by, InverseObjectProperties(teaches taught-by)
  - teaches $\subseteq T \times T$
  - taughtBy $\subseteq T \times T$
  - teaches $\equiv$ taughtBy$^-$
- domain teaches: Prof, and range teaches: Course
  - teaches $\subseteq$ Prof $\times$ Course
- domain teaches: Prof, and range teaches: Course, domain taught-by: Course, range taught-by: Prof
  - teaches $\subseteq$ Prof $\times$ Course
  - taughtBy $\subseteq$ Course $\times$ Prof

(simplified notation online)
OWL files

- http://www.meteck.org/teaching/ontologies/ has various versions of the African Wildlife Ontology (alone, linked to DOLCE, link to GFO)

- http://www.meteck.org/files/ontologies/EvalComputer.owl has no object properties at all. add both properties and axioms (details of exercise depends on number of participants)

- Pick one. Add missing object properties and/or axioms (details of exercise depends on number of participants)
The Wildlife Ontology and DOLCE

- Giraffes eat leaves and twigs. how do Plant and Twig relate?
- The elephant’s tusks (ivory) are made of apatite (calcium phosphate); which DOLCE relation can be reused?
- How would you represent the Size (Height, Weight, etc.) of an average adult elephant?
  - with quality and quale
  - OWL data properties
  -
The Wildlife Ontology and DOLCE

- Giraffes eat leaves and twigs. how do Plant and Twig relate?
  - (some type of) parthood relation
- The elephant’s tusks (ivory) are made of apatite (calcium phosphate); which DOLCE relation can be reused?
  - constitution
- How would you represent the Size (Height, Weight, etc.) of an average adult elephant?
  - with quality and quale
  - OWL data properties
The Wildlife Ontology and DOLCE

- Giraffes eat leaves and twigs. How do Plant and Twig relate?
  - (some type of) parthood relation
- The elephant’s tusks (ivory) are made of apatite (calcium phosphate); which DOLCE relation can be reused?
  - constitution
- How would you represent the Size (Height, Weight, etc.) of an average adult elephant?
  - with quality and quale
  - OWL data properties
    - What is the data type; integer, float, real, string?
    - Measure in meter, feet, kg, lb?
    - Introduce “ElephantHeight”, and also “LionHeight”, “GiraffeHeight”, “ImpalaHeight”, etc.?
A computer ontology

- CPU and Desktop?
- Who are members of an Agile team?
A computer ontology

- CPU and Desktop?
  - containment

- Who are members of an Agile team?
  - hasMember vs. memberOf
Outline

1. Introduction
2. Semantics of relations
   - Positionalism
   - Hierarchies of relations
3. Some common relations
   - Part-whole relations
   - Mereotopology
   - Beyond parts and space
4. Modelling and reasoning
   - Reasoner-mediated modelling
   - Performance considerations
   - Hands-on
5. Recap
## Recap

1. **Introduction**
2. **Semantics of relations**
   - Positionalism
   - Hierarchies of relations
3. **Some common relations**
   - Part-whole relations
   - Mereotopology
   - Beyond parts and space
4. **Modelling and reasoning**
   - Reasoner-mediated modelling
   - Performance considerations
   - Hands-on
5. **Recap**
References I

Elena Beisswanger, Stefan Schulz, Holger Stenzhorn, and Udo Hahn.
BioTop: An upper domain ontology for the life sciences - a description of its current structure, contents, and interfaces to OBO ontologies.

Anthony G. Cohn and Jochen Renz.
Chapter 13 qualitative spatial representation and reasoning.

M. J. Egenhofer and J. R. Herring.
Categorizing binary topological relations between regions, lines, and points in geographic databases.
Technical Report 90-12, National Center for Geographic Information and Analysis, University of California, 1990.

C. Eschenbach and W. Heydrich.
Classical mereology and restricted domains.

Kit Fine.
Neutral relations.

Chiara Ghidini, Barbara Kump, Stefanie Lindstaedt, Nahid Mabhub, Viktoria Pammer, Marco Rospocher, and Luciano Serafini.
Moki: The enterprise modelling wiki.
References II

R.S. Goncalves, B. Parsia, and U. Sattler.
Performance heterogeneity and approximate reasoning in description logic ontologies.

Giancarlo Guizzardi and Gerd Wagner.
What’s in a relationship: An ontological analysis.

K. Hammar.
Ontology design pattern property specialisation strategies.

M. Horridge, B. Parsia, and U. Sattler.
Laconic and precise justifications in OWL.

The even more irresistible $\mathcal{SROIQ}$.
References III

Repairing unsatisfiable concepts in OWL ontologies.

Y.-B. Kang, Y.-F. Li, and S. Krishnaswamy.
Predicting reasoning performance using ontology metrics.

C. Maria Keet.
Positionalism of relations and its consequences for fact-oriented modelling.

C. Maria Keet.
Preventing, detecting, and revising flaws in object property expressions.

C. Maria Keet and Alessandro Artale.
Representing and reasoning over a taxonomy of part-whole relations.
C. Maria Keet and Pablo Rubén Fillottrani.

Toward an ontology-driven unifying metamodel for UML class diagrams, EER, and ORM2.
11-13 November, 2013, Hong Kong.

C. Maria Keet, Francis C. Fernández-Reyes, and Annette Morales-González.

Representing mereotopological relations in OWL ontologies with ONTOPARTS.
29-31 May 2012, Heraklion, Crete, Greece.

C. Maria Keet, M. Tahir Khan, and Chiara Ghidini.

Guided ENtity reuse and class Expression geneRATOR.
23-26 June 2013, Banff, Canada.

C. Maria Keet, M. Tahir Khan, and Chiara Ghidini.

Ontology authoring with FORZA.
Oct. 27 - Nov. 1, 2013, San Francisco, USA.
References V

C.M. Keet, C. d’Amato, Z.C. Khan, and A. Lawrynowicz.  
Exploring reasoning with the DMOP ontology.  
July 13, 2014, Vienna, Austria.

Joop Leo.  
Modeling relations.  

C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari.  
Ontology library.  

B. Parsia, E. Sirin, and A. Kalyanpur.  
Debugging OWL ontologies.  
May 10-14, 2005, Chiba, Japan.

D. A. Randell, Z. Cui, and A. G. Cohn.  
A spatial logic based on regions and connection.  
References VI

Relations in biomedical ontologies.

A.C. Varzi.
*Handbook of Spatial Logics*, chapter Spatial reasoning and ontology: parts, wholes, and locations, pages 945–1038.
Berlin Heidelberg: Springer Verlag, 2007.