Orchestrating a Network of Mereo(topo)logical Theories

Oliver Kutz[1]
joint work with C. Maria Keet[2]

[1] KRDB Research Centre, Free University of Bozen-Bolzano, Italy
Oliver.Kutz@unibz.it
[2] Department of Computer Science, University of Cape Town, South Africa,
keet@cs.uct.ac.za

LAC 2018, Melbourne

- Motivation: ubiquity of mereo(topo)logy
- Heterogeneity of logics and modelling dilemmas
- DOL as structuring language
- Mereotopologies within DOL networks
- A tool for resolving OWL language feature conflicts
- Outlook and Future Work

Mereology and Mereotopology

- **Mereology** is the theory of part and whole
- **Topology** studies connection, boundary, interior and exterior
- **Mereotopology** combines mereology and topology into a combined theory

Mereology and Mereotopology

- **Mereology** is the theory of part and whole
- **Topology** studies connection, boundary, interior and exterior
- **Mereotopology** combines mereology and topology into a combined theory

Mereology and Mereotopology

- **Mereology** is the theory of part and whole
- **Topology** studies connection, boundary, interior and exterior
- **Mereotopology** combines mereology and topology into a combined theory

Mereology in Knowledge Engineering

- Mereology is well established, studied in ontology and philosophy, and considered a possible alternative to set theory (e.g. Lewis 91)
- widely applied in knowledge-rich application areas such as:
  - medical terminologies (openGalen, with 23 part-whole relations)
  - SNOMET CT, Gene Ontology
  - Foundational Model of Anatomy

Mereotopology in Knowledge Engineering

- Mereotopology may extend mereology for instance by a distinction between interior and tangential part, applications in:
  - geographic information systems
  - geology, image annotation, etc.
- For instance:
  - Let NTPLI be a 'non-tangential proper located in' relation
  - $\texttt{EnclosedCountry} \equiv \texttt{Country} \sqcap \exists\texttt{NTPLI.Country}$
  - 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

1st Challenge: expressivity and reasoning constraints

- Different versions of mereology, topology, mereotopology require different logical languages to be expressible, e.g.
  - OWL 2 DL cannot express antisymmetry, so parthood cannot be fully expressed
  - FOL cannot express general fusion principles, so this requires higher-order logic
- Application scenarios might require a
  - dedicable language: stick to OWL 2 DL or weaker
  - fully automated theorem proving / semi-decidability: stick to FOL
  - most precise logical modelling: stick to HOL

1st Challenge: expressivity and reasoning constraints

- Different versions of mereology, topology, mereotopology require different logical languages to be expressible, e.g.
  - OWL 2 DL cannot express antisymmetry, so parthood cannot be fully expressed
  - FOL cannot express general fusion principles, so this requires higher-order logic
- Application scenarios might require a
  - dedicable language: stick to OWL 2 DL or weaker
  - fully automated theorem proving / semi-decidability: stick to FOL
  - most precise logical modelling: stick to HOL

1st Challenge: expressivity and reasoning constraints

- Different versions of mereology, topology, mereotopology require different logical languages to be expressible, e.g.
  - OWL 2 DL cannot express antisymmetry, so parthood cannot be fully expressed
  - FOL cannot express general fusion principles, so this requires higher-order logic
- Application scenarios might require a
  - dedicable language: stick to OWL 2 DL or weaker
  - fully automated theorem proving / semi-decidability: stick to FOL
  - most precise logical modelling: stick to HOL

1st Challenge: expressivity and reasoning constraints

- Different versions of mereology, topology, mereotopology require different logical languages to be expressible, e.g.
  - OWL 2 DL cannot express antisymmetry, so parthood cannot be fully expressed
  - FOL cannot express general fusion principles, so this requires higher-order logic
- Application scenarios might require a
  - dedicable language: stick to OWL 2 DL or weaker
  - fully automated theorem proving / semi-decidability: stick to FOL
  - most precise logical modelling: stick to HOL

2nd Challenge: Breaking Language Restrictions

- The OWL 2 DL logic SROIQ imposes syntactic restrictions on **legal ontologies**, such that the naive combination of two OWL ontologies is not an OWL ontology anymore.
- **Example:**
  - Ontology O_1 declares the relation *part_of* as transitive
  - Ontology O_2 uses qualified number restrictions on *part_of*
  - when combining O_1 and O_2, e.g. via import, the result is not a legal SROIQ ontology.

2nd Challenge: Breaking Language Restrictions

- The OWL 2 DL logic SROIQ imposes syntactic restrictions on **legal ontologies**, such that the naive combination of two OWL ontologies is not an OWL ontology anymore.
- **Example:**
  - Ontology O_1 declares the relation *part_of* as transitive
  - Ontology O_2 uses qualified number restrictions on *part_of*
  - when combining O_1 and O_2, e.g. via import, the result is not a legal SROIQ ontology.

Example: Breaking Language Restrictions

- Consider the example of humans with their feet and limbs as example. We have the following choices:
  - **(1) drop number restrictions:** humans can have as part any number of limbs and infer that if a foot is part of a limb and a limb part of a human, then that foot is part of that human;
  - **(2) drop transitivity:** a (canonical) human has as part exactly four limbs but it cannot be inferred that the foot is part of the human;
  - **(3) keep both:** a human has exactly four limbs and we can make the (transitive) inferences about feet
- Now, in (1) and (2) we need to give up important entailments (but which one should be given up?), and in (3) we have to give up OWL tool support, possibly decidability, and reasoning is only available via translation to FOL.

Example: Breaking Language Restrictions

- Consider the example of humans with their feet and limbs as example. We have the following choices:
  - **(1) drop number restrictions:** humans can have as part any number of limbs and infer that if a foot is part of a limb and a limb part of a human, then that foot is part of that human;
  - **(2) drop transitivity:** a (canonical) human has as part exactly four limbs but it cannot be inferred that the foot is part of the human;
  - **(3) keep both:** a human has exactly four limbs and we can make the (transitive) inferences about feet
- Now, in (1) and (2) we need to give up important entailments (but which one should be given up?), and in (3) we have to give up OWL tool support, possibly decidability, and reasoning is only available via translation to FOL.

Example: Breaking Language Restrictions

- Consider the example of humans with their feet and limbs as example. We have the following choices:
  - **(1) drop number restrictions:** humans can have as part any number of limbs and infer that if a foot is part of a limb and a limb part of a human, then that foot is part of that human;
  - **(2) drop transitivity:** a (canonical) human has as part exactly four limbs but it cannot be inferred that the foot is part of the human;
  - **(3) keep both:** a human has exactly four limbs and we can make the (transitive) inferences about feet
- Now, in (1) and (2) we need to give up important entailments (but which one should be given up?), and in (3) we have to give up OWL tool support, possibly decidability, and reasoning is only available via translation to FOL.

Example: Breaking Language Restrictions

- Consider the example of humans with their feet and limbs as example. We have the following choices:
  - **(1) drop number restrictions:** humans can have as part any number of limbs and infer that if a foot is part of a limb and a limb part of a human, then that foot is part of that human;
  - **(2) drop transitivity:** a (canonical) human has as part exactly four limbs but it cannot be inferred that the foot is part of the human;
  - **(3) keep both:** a human has exactly four limbs and we can make the (transitive) inferences about feet
- Now, in (1) and (2) we need to give up important entailments (but which one should be given up?), and in (3) we have to give up OWL tool support, possibly decidability, and reasoning is only available via translation to FOL.

Example: Breaking Language Restrictions

- Consider the example of humans with their feet and limbs as example. We have the following choices:
  - **(1) drop number restrictions:** humans can have as part any number of limbs and infer that if a foot is part of a limb and a limb part of a human, then that foot is part of that human;
  - **(2) drop transitivity:** a (canonical) human has as part exactly four limbs but it cannot be inferred that the foot is part of the human;
  - **(3) keep both:** a human has exactly four limbs and we can make the (transitive) inferences about feet
- Now, in (1) and (2) we need to give up important entailments (but which one should be given up?), and in (3) we have to give up OWL tool support, possibly decidability, and reasoning is only available via translation to FOL.

3th Challenge: Re-use, Alignment and Extension

- Standard KR languages, here OWL, FOL, HOL, have essentially no build-in support for structuring logical theories. Basic problems are:
  - extending a (part of) an available theory
  - renaming the symbols of a given theory
  - aligning the symbols of two related theories
  - switching the logic of a given theory
  - extending a given theory with axioms from a more expressive logic

3th Challenge: Re-use, Alignment and Extension

- Standard KR languages, here OWL, FOL, HOL, have essentially no build-in support for structuring logical theories. Basic problems are:
  - extending a (part of) an available theory
  - renaming the symbols of a given theory
  - aligning the symbols of two related theories
  - switching the logic of a given theory
  - extending a given theory with axioms from a more expressive logic

3th Challenge: Re-use, Alignment and Extension

- Standard KR languages, here OWL, FOL, HOL, have essentially no build-in support for structuring logical theories. Basic problems are:
  - extending a (part of) an available theory
  - renaming the symbols of a given theory
  - aligning the symbols of two related theories
  - switching the logic of a given theory
  - extending a given theory with axioms from a more expressive logic

3th Challenge: Re-use, Alignment and Extension

- Standard KR languages, here OWL, FOL, HOL, have essentially no build-in support for structuring logical theories. Basic problems are:
  - extending a (part of) an available theory
  - renaming the symbols of a given theory
  - aligning the symbols of two related theories
  - switching the logic of a given theory
  - extending a given theory with axioms from a more expressive logic

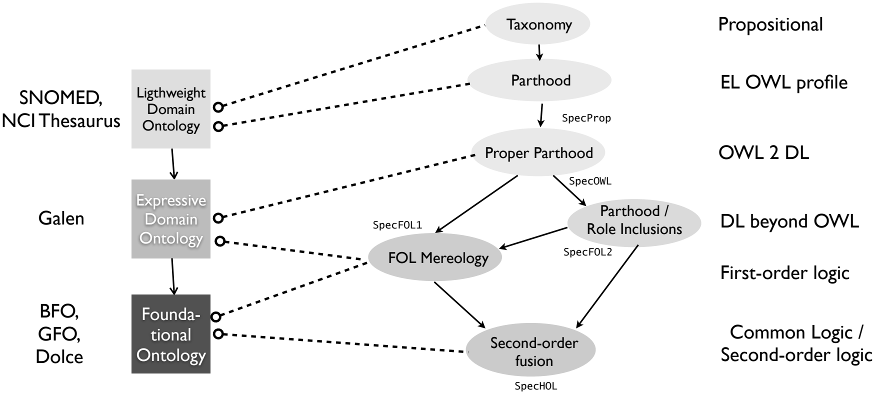
3th Challenge: Re-use, Alignment and Extension

- Standard KR languages, here OWL, FOL, HOL, have essentially no build-in support for structuring logical theories. Basic problems are:
  - extending a (part of) an available theory
  - renaming the symbols of a given theory
  - aligning the symbols of two related theories
  - switching the logic of a given theory
  - extending a given theory with axioms from a more expressive logic

3th Challenge: Re-use, Alignment and Extension

- Standard KR languages, here OWL, FOL, HOL, have essentially no build-in support for structuring logical theories. Basic problems are:
  - extending a (part of) an available theory
  - renaming the symbols of a given theory
  - aligning the symbols of two related theories
  - switching the logic of a given theory
  - extending a given theory with axioms from a more expressive logic

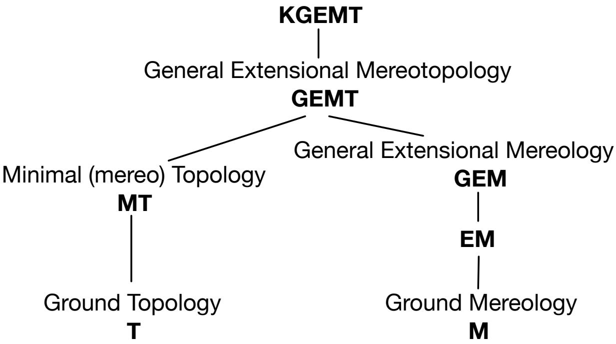
Example: Mereology in bio-medical ontologies



Solutions in a Nutshell

- We organise a heterogeneous network of 28 mereo(topo)logical theories into a structured network
- We employ the DOL language to allow for logical heterogeneity and to solve problems of alignment, extensions, and re-use (explained below)
- We give mechanisms to record lemmas, consistency proofs, and counterexamples
- We have developed a tool to resolve OWL language feature conflicts
- Our framework is fully general and extensible

Kuratowski extension of GEMT (KGEMT)



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

Ground Topology

Core axioms and definitions			
$P(x,x)$	(t1)	$P(x,y) \wedge P(y,z) \to P(x,z)$	(t2)
$P(x,y) \wedge P(y,x) \to x = y$	(t3)	$\neg P(y,x) \to \exists z(P(z,y) \wedge \neg O(z,x))$	(t4)
$\exists w \phi(w) \to \exists z \forall w(O(w,z) \leftrightarrow \exists v(\phi(v) \wedge O(w,v)))$			(t5)
$C(x,x)$	(t6)	$C(x,y) \to C(y,x)$	(t7)
$P(x,y) \to E(x,y)$	(t8)	$E(x,y) =_{df} \forall z(C(z,x) \to C(z,y))$	(t9)
$E(x,y) \to P(x,y)$	(t10)	$SC(x) \leftrightarrow \forall y,z(x = y + z \to C(y,z))$	(t11)
$\exists z(SC(z) \wedge O(z,x) \wedge O(z,y) \wedge \forall w(P(w,z) \to (O(w,x) \vee O(w,y)))) \to C(x,y)$			(t12)
$z = \sum x \phi x \to \forall y(C(y,z) \to \exists x(\phi x \wedge C(y,x)))$			(t13)
$P(x,cx)$	(t14)	$c(cx) = cx$	(t15)
$c(x+y) = cx + cy$	(t16)	$cx =_{df} \sim(ex)$	(t17)
$ex =_{df} i(\sim x)$	(t18)	$ix =_{df} \sum z \forall y(C(z,y) \to O(x,y))$	(t19)
Additional axioms, definitions, and theorems			
$PP(x,y) =_{df} P(x,y) \wedge \neg P(y,x)$	(t20)	$O(x,y) =_{df} \exists z(P(z,x) \wedge P(z,y))$	(t21)
$EQ(x,y) =_{df} P(x,y) \wedge P(y,x)$	(t22)	$TPP(x,y) =_{df} PP(x,y) \wedge \neg IPP(x,y)$	(t23)
$IPP(x,y) =_{df} PP(x,y) \wedge \forall z(C(z,x) \to O(z,y))$			(t24)
$\neg PP(x,x)$	(t25)	$PP(x,y) \wedge PP(y,z) \to PP(x,z)$	(t26)
$PP(x,y) \to \neg PP(y,x)$	(t27)		

Minimal (mereo) Topology

Core axioms and definitions			
$P(x,x)$	(t1)	$P(x,y) \wedge P(y,z) \rightarrow P(x,z)$	(t2)
$P(x,y) \wedge P(y,x) \rightarrow x = y$	(t3)	$\neg P(y,x) \rightarrow \exists z(P(z,y) \wedge \neg O(z,x))$	(t4)
$\exists w \phi(w) \rightarrow \exists z \forall w(O(w,z) \leftrightarrow \exists v(\phi(v) \wedge O(w,v)))$			(t5)
$C(x,x)$	(t6)	$C(x,y) \rightarrow C(y,x)$	(t7)
$P(x,y) \rightarrow E(x,y)$	(t8)	$E(x,y) =_{df} \forall z(C(z,x) \rightarrow C(z,y))$	(t9)
$E(x,y) \rightarrow P(x,y)$	(t10)	$SC(x) \leftrightarrow \forall y,z(x = y + z \rightarrow C(y,z))$	(t11)
$\exists z(SC(z) \wedge O(z,x) \wedge O(z,y) \wedge \forall w(P(w,z) \rightarrow (O(w,x) \vee O(w,y)))) \rightarrow C(x,y)$			(t12)
$z = \sum x \phi x \rightarrow \forall y(C(y,z) \rightarrow \exists x(\phi x \wedge C(y,x)))$			(t13)
$P(x,cx)$	(t14)	$c(cx) = cx$	(t15)
$c(x+y) = cx + cy$	(t16)	$cx =_{df} \sim (ex)$	(t17)
$ex =_{df} i(\sim x)$	(t18)	$ix =_{df} \sum z \forall y(C(z,y) \rightarrow O(x,y))$	(t19)
Additional axioms, definitions, and theorems			
$PP(x,y) =_{df} P(x,y) \wedge \neg P(y,x)$	(t20)	$O(x,y) =_{df} \exists z(P(z,x) \wedge P(z,y))$	(t21)
$EQ(x,y) =_{df} P(x,y) \wedge P(y,x)$	(t22)	$TPP(x,y) =_{df} PP(x,y) \wedge \neg IPP(x,y)$	(t23)
$IPP(x,y) =_{df} PP(x,y) \wedge \forall z(C(z,x) \rightarrow O(z,y))$			(t24)
$\neg PP(x,x)$	(t25)	$PP(x,y) \wedge PP(y,z) \rightarrow PP(x,z)$	(t26)
$PP(x,y) \rightarrow \neg PP(y,x)$	(t27)		

Minimal (mereo) Topology; Ground Mereology

Core axioms and definitions			
$P(x,x)$	(t1)	$P(x,y) \wedge P(y,z) \rightarrow P(x,z)$	(t2)
$P(x,y) \wedge P(y,x) \rightarrow x = y$	(t3)	$\neg P(y,x) \rightarrow \exists z(P(z,y) \wedge \neg O(z,x))$	(t4)
$\exists w \phi(w) \rightarrow \exists z \forall w(O(w,z) \leftrightarrow \exists v(\phi(v) \wedge O(w,v)))$			(t5)
$C(x,x)$	(t6)	$C(x,y) \rightarrow C(y,x)$	(t7)
$P(x,y) \rightarrow E(x,y)$	(t8)	$E(x,y) =_{df} \forall z(C(z,x) \rightarrow C(z,y))$	(t9)
$E(x,y) \rightarrow P(x,y)$	(t10)	$SC(x) \leftrightarrow \forall y,z(x = y + z \rightarrow C(y,z))$	(t11)
$\exists z(SC(z) \wedge O(z,x) \wedge O(z,y) \wedge \forall w(P(w,z) \rightarrow (O(w,x) \vee O(w,y)))) \rightarrow C(x,y)$			(t12)
$z = \sum x \phi x \rightarrow \forall y(C(y,z) \rightarrow \exists x(\phi x \wedge C(y,x)))$			(t13)
$P(x, cx)$	(t14)	$c(cx) = cx$	(t15)
$c(x+y) = cx + cy$	(t16)	$cx =_{df} \sim(ex)$	(t17)
$ex =_{df} i(\sim x)$	(t18)	$ix =_{df} \sum z \forall y(C(z,y) \rightarrow O(x,y))$	(t19)
Additional axioms, definitions, and theorems			
$PP(x,y) =_{df} P(x,y) \wedge \neg P(y,x)$	(t20)	$O(x,y) =_{df} \exists z(P(z,x) \wedge P(z,y))$	(t21)
$EQ(x,y) =_{df} P(x,y) \wedge P(y,x)$	(t22)	$TPP(x,y) =_{df} PP(x,y) \wedge \neg IPP(x,y)$	(t23)
$IPP(x,y) =_{df} PP(x,y) \wedge \forall z(C(z,x) \rightarrow O(z,y))$			(t24)
$\neg PP(x,x)$	(t25)	$PP(x,y) \wedge PP(y,z) \rightarrow PP(x,z)$	(t26)
$PP(x,y) \rightarrow \neg PP(y,x)$	(t27)		

Minimal (mereo) Topology; General Extensional Mereology

Core axioms and definitions			
$P(x,x)$	(t1)	$P(x,y) \wedge P(y,z) \rightarrow P(x,z)$	(t2)
$P(x,y) \wedge P(y,x) \rightarrow x = y$	(t3)	$\neg P(y,x) \rightarrow \exists z(P(z,y) \wedge \neg O(z,x))$	(t4)
$\exists w \phi(w) \rightarrow \exists z \forall w(O(w,z) \leftrightarrow \exists v(\phi(v) \wedge O(w,v)))$			(t5)
$C(x,x)$	(t6)	$C(x,y) \rightarrow C(y,x)$	(t7)
$P(x,y) \rightarrow E(x,y)$	(t8)	$E(x,y) =_{df} \forall z(C(z,x) \rightarrow C(z,y))$	(t9)
$E(x,y) \rightarrow P(x,y)$	(t10)	$SC(x) \leftrightarrow \forall y,z(x = y + z \rightarrow C(y,z))$	(t11)
$\exists z(SC(z) \wedge O(z,x) \wedge O(z,y) \wedge \forall w(P(w,z) \rightarrow (O(w,x) \vee O(w,y)))) \rightarrow C(x,y)$			(t12)
$z = \sum x \phi x \rightarrow \forall y(C(y,z) \rightarrow \exists x(\phi x \wedge C(y,x)))$			(t13)
$P(x,cx)$	(t14)	$c(cx) = cx$	(t15)
$c(x+y) = cx + cy$	(t16)	$cx =_{df} \sim (ex)$	(t17)
$ex =_{df} i(\sim x)$	(t18)	$ix =_{df} \sum z \forall y(C(z,y) \rightarrow O(x,y))$	(t19)
Additional axioms, definitions, and theorems			
$PP(x,y) =_{df} P(x,y) \wedge \neg P(y,x)$	(t20)	$O(x,y) =_{df} \exists z(P(z,x) \wedge P(z,y))$	(t21)
$EQ(x,y) =_{df} P(x,y) \wedge P(y,x)$	(t22)	$TPP(x,y) =_{df} PP(x,y) \wedge \neg IPP(x,y)$	(t23)
$IPP(x,y) =_{df} PP(x,y) \wedge \forall z(C(z,x) \rightarrow O(z,y))$			(t24)
$\neg PP(x,x)$	(t25)	$PP(x,y) \wedge PP(y,z) \rightarrow PP(x,z)$	(t26)
$PP(x,y) \rightarrow \neg PP(y,x)$	(t27)		

General Extensional MereoTopology

Core axioms and definitions			
$P(x,x)$	(t1)	$P(x,y) \wedge P(y,z) \rightarrow P(x,z)$	(t2)
$P(x,y) \wedge P(y,x) \rightarrow x = y$	(t3)	$\neg P(y,x) \rightarrow \exists z(P(z,y) \wedge \neg O(z,x))$	(t4)
$\exists w \phi(w) \rightarrow \exists z \forall w(O(w,z) \leftrightarrow \exists v(\phi(v) \wedge O(w,v)))$			(t5)
$C(x,x)$	(t6)	$C(x,y) \rightarrow C(y,x)$	(t7)
$P(x,y) \rightarrow E(x,y)$	(t8)	$E(x,y) =_{df} \forall z(C(z,x) \rightarrow C(z,y))$	(t9)
$E(x,y) \rightarrow P(x,y)$	(t10)	$SC(x) \leftrightarrow \forall y,z(x = y + z \rightarrow C(y,z))$	(t11)
$\exists z(SC(z) \wedge O(z,x) \wedge O(z,y) \wedge \forall w(P(w,z) \rightarrow (O(w,x) \vee O(w,y)))) \rightarrow C(x,y)$			(t12)
$z = \sum x \phi x \rightarrow \forall y(C(y,z) \rightarrow \exists x(\phi x \wedge C(y,x)))$			(t13)
$P(x,cx)$	(t14)	$c(cx) = cx$	(t15)
$c(x+y) = cx + cy$	(t16)	$cx =_{df} \sim(ex)$	(t17)
$ex =_{df} i(\sim x)$	(t18)	$ix =_{df} \sum z \forall y(C(z,y) \rightarrow O(x,y))$	(t19)
Additional axioms, definitions, and theorems			
$PP(x,y) =_{df} P(x,y) \wedge \neg P(y,x)$	(t20)	$O(x,y) =_{df} \exists z(P(z,x) \wedge P(z,y))$	(t21)
$EQ(x,y) =_{df} P(x,y) \wedge P(y,x)$	(t22)	$TPP(x,y) =_{df} PP(x,y) \wedge \neg IPP(x,y)$	(t23)
$IPP(x,y) =_{df} PP(x,y) \wedge \forall z(C(z,x) \rightarrow O(z,y))$			(t24)
$\neg PP(x,x)$	(t25)	$PP(x,y) \wedge PP(y,z) \rightarrow PP(x,z)$	(t26)
$PP(x,y) \rightarrow \neg PP(y,x)$	(t27)		

Kuratowski General Extensional MereoTopology

Core axioms and definitions			
$P(x,x)$	(t1)	$P(x,y) \wedge P(y,z) \rightarrow P(x,z)$	(t2)
$P(x,y) \wedge P(y,x) \rightarrow x = y$	(t3)	$\neg P(y,x) \rightarrow \exists z(P(z,y) \wedge \neg O(z,x))$	(t4)
$\exists w \phi(w) \rightarrow \exists z \forall w(O(w,z) \leftrightarrow \exists v(\phi(v) \wedge O(w,v)))$			(t5)
$C(x,x)$	(t6)	$C(x,y) \rightarrow C(y,x)$	(t7)
$P(x,y) \rightarrow E(x,y)$	(t8)	$E(x,y) =_{df} \forall z(C(z,x) \rightarrow C(z,y))$	(t9)
$E(x,y) \rightarrow P(x,y)$	(t10)	$SC(x) \leftrightarrow \forall y,z(x = y + z \rightarrow C(y,z))$	(t11)
$\exists z(SC(z) \wedge O(z,x) \wedge O(z,y) \wedge \forall w(P(w,z) \rightarrow (O(w,x) \vee O(w,y)))) \rightarrow C(x,y)$			(t12)
$z = \sum x \phi x \rightarrow \forall y(C(y,z) \rightarrow \exists x(\phi x \wedge C(y,x)))$			(t13)
$P(x,cx)$	(t14)	$c(cx) = cx$	(t15)
$c(x+y) = cx + cy$	(t16)	$cx =_{df} \sim (ex)$	(t17)
$ex =_{df} i(\sim x)$	(t18)	$ix =_{df} \sum z \forall y(C(z,y) \rightarrow O(x,y))$	(t19)
Additional axioms, definitions, and theorems			
$PP(x,y) =_{df} P(x,y) \wedge \neg P(y,x)$	(t20)	$O(x,y) =_{df} \exists z(P(z,x) \wedge P(z,y))$	(t21)
$EQ(x,y) =_{df} P(x,y) \wedge P(y,x)$	(t22)	$TPP(x,y) =_{df} PP(x,y) \wedge \neg IPP(x,y)$	(t23)
$IPP(x,y) =_{df} PP(x,y) \wedge \forall z(C(z,x) \rightarrow O(z,y))$			(t24)
$\neg PP(x,x)$	(t25)	$PP(x,y) \wedge PP(y,z) \rightarrow PP(x,z)$	(t26)
$PP(x,y) \rightarrow \neg PP(y,x)$	(t27)		

Subsets of KGEMT in OWL / FOL / HOL

N	Language	Subsets of KGEMT axioms	Comments
1	OWL 2 QL	t1, t21p, t22p	M, with p, partially
2	OWL 2 QL	t6, t7	T, c
3	OWL 2 QL	t20p, t21p, t22p, t25, t27	M, pp
4	OWL 2 QL	t6, t7, t8, t9p	MT
5	OWL 2 QL	t1, t6, t7, t8, t9p , t10, t20p, t21p, t22p, t23p, t24p, t25, t27	GEMT, partial
6	OWL 2 EL, 2 QL	t1, t2, t21p, t22p	M, with p, partially
7	OWL 2 EL	t6	T, c, partial
8	OWL 2 EL, 2QL	t6, t8, t9p	MT, partially
9	OWL 2 EL	t1, t2, t6, t8, t9p, t10, t26, t20p, t21p, t22p, t23p, t24p	GEMT, partial
10	OWL 2 RL, OWL Lite, DL	t2, t21p, t22p	M, p, partial
11	OWL 2 RL, 2QL, OWL Lite, DL	t7	T c, partial
12	OWL 2 RL, EL, DL, OWL Lite, DL	t2, t26, t20p, t21p, t22p	M, with p and pp both partially
13	OWL 2 RL, OWL Lite, DL	t7,t8, t9p	MT partial
14	OWL 2 RL, OWL Lite, DL	t2, t7, t8, t9p, t10, t26, t20p, t21p, t22p, t23p, t24p	GEMT, partial
15	OWL 2 DL	t1, t2, t6, t7, t8, t9p, t10, t25, t27, t20p, t21p, t22p, t23p, t24p	GEMT, partial
16	OWL 2 DL	t1, t2, t6, t7, t8, t9p, t10, t26, t20p, t21p, t22p, t23p, t24p	GEMT, partial
17	OWL 2 RL	t2, t20p, t21p, t22p, t25, t27	M with p and pp, both partial
18	OWL 2 DL	t1, t2, t25, t27, t20p, t21p, t22p	M with p and pp, both partial
19	OWL 2 EL	t1, t2, t26, t20p, t21p, t22p	M with p and pp, partial
20	FOL, HOL	t1, t2, t3, t21, t22, t4	M, with p
21	FOL, HOL	t1, t2, t3, t20, t21, t22, t25, t26, t27	M, with p and pp
22	FOL	t1-t4, t20, t21, t22, t25, t26, t27	GEM, partial
23	FOL, HOL	t6, t7, t8, t9	MT
24	FOL	t1-t4, t6-t12, t20-t27	GEMT, partial
25	FOL	t1-t4, t6-t12, t14-t27	KGEMT, partial
26	HOL	t1-t5, t20, t21, t22, t25, t26, t27	GEM
27	HOL	t1-t13, t20-t27	GEMT
28	HOL	t1-t27	KGEMT

Properties of (proper) parthood / connection

Table: Properties of parthood ($.^P$) and proper parthood ($.^{PP}$) in Ground Mereology, and connection ($.^C$) in Ground Topology and their inclusion in the OWL family and FOL.

Language ⇒	DL-based OWL species						FOL
Feature ⇓	DL	Lite	2DL	2QL	2RL	2EL	
Symmetry^C	+	+	+	+	+	−	+
Reflexivity^{P,C}	−	−	+	+	−	+	+
Antisymmetry^P	−	−	−	−	−	−	+
Transitivity^{P,PP}	+	+	+	−	+	+	+
Asymmetry^{PP}	−	−	+	+	+	−	+
Irreflexivity^{PP}	−	−	+	+	+	−	+

Conflict Resolution tool

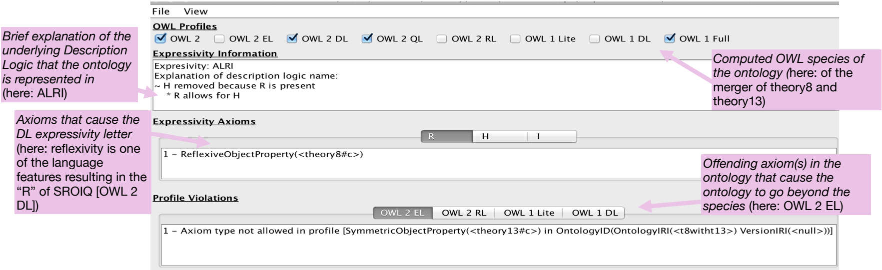


Figure: Annotated screenshot of the OWL species classifier output of the merger between theory8 and theory13.

What is DOL?

DOL is a **metalanguage** that enables

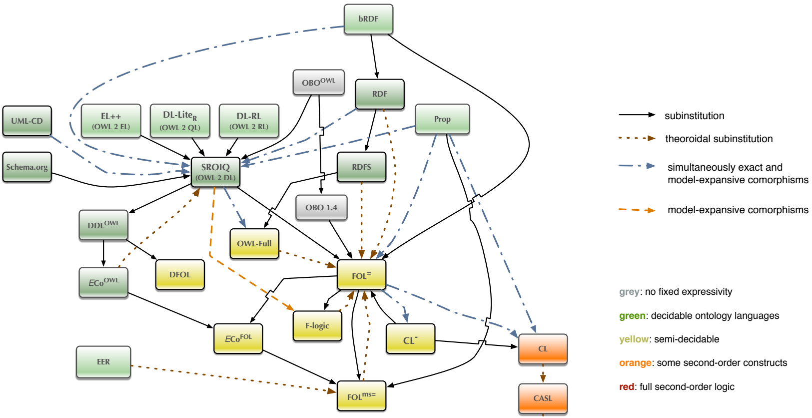
- definition of new, structured ontologies based on old ontologies
- based on a graph of logics and logic translations; logic independent
- declaration of relations between ontologies (interpretations etc.)
- specification of intended relationships (proof obligations)
- combination of ontologies along links (colimits)
- for homo- and heterogeneous ontologies
- model-theoretic semantics
- under development since 2009 building on HetCASL
- standardised at OMG and adopted in 2016, finalised this week at an OMG meeting in San Francisco.

Distinctive DOL features

Key features that DOL adds on top of OWL DL
(and other supported logics) are:

- structuring constructs for building ontologies from existing ontologies, namely imports, extensions, unions, forgetting, interpolation, filtering
- open-world versus closed-world semantics (using circumscription)
- module extraction
- mappings between ontologies, like interpretations of theories, conservative and definitional extensions etc.
- alignments and networks of ontologies
- combination of networks

DOL Semantic Foundations: Logic Translations



DOL Extensions

We **extend** theory8 (OWL 2 EL/QL) into theory4 (OWL 2 QL) by adding symmetry (t7). This is written as follows:

```
logic OWL2.QL
ontology theory4 =
theory8
then
ObjectProperty: C Characteristics: Symmetric %(t7)
```

Semantic domains for OMS in DOL

Flattenable OMS (can be flattened to a basic OMS)

- basic OMS
- extensions, unions, translations
- approximations, module extractions, filterings
- *semantics*: (Σ, Ψ) (theory-level)
  - Σ: a signature in I, also written $Sig(O)$
  - Ψ: a set of Σ-sentences, also written $Th(O)$

Elusive OMS (= non-flattenable OMS)

- reductions, minimization, maximization, (co)freeness (elusive)
- *semantics:* $(\Sigma, \mathcal{M})$ (model-level)
  - Σ: a signature in I, also written $Sig(O)$
  - $\mathcal{M}$: a class of Σ-models, also written $Mod(O)$

We can obtain the model-level semantics from the theory-level semantics by taking $\mathcal{M} = \{M \in \mathbf{Mod}(\Sigma) \mid M \models \Psi\}$.

Semantics of basic OMS

We assume that $[\![O]\!]_{basic} = (\Sigma, \Psi)$ for some OMS language based on I. The semantics consists of

- a signature Σ in I
- a set Ψ of Σ-sentences

This direct leads to a theory-level semantics for OMSs:

$$[\![O]\!]_{\Gamma}^{T} = [\![O]\!]_{basic}$$

Generally, if a theory-level semantics is given: $[\![O]\!]_{\Gamma}^{T} = (\Sigma, \Psi)$, this leads to a model-level semantics as well:

$$[\![O]\!]_{\Gamma}^{M} = (\Sigma, \{M \in Mod(\Sigma) \mid M \models \Psi\})$$

Semantics of extensions

O_1 flattenable $[\![O_1 \textbf{ then } O_2]\!]^T_\Gamma = (\Sigma_1 \cup \Sigma_2, \Psi_1 \cup \Psi_2)$
where

- $[\![O_1]\!]^T_\Gamma = (\Sigma_1, \Psi_1)$
- $[\![O_2]\!]_{basic} = (\Sigma_2, \Psi_2)$

O_1 elusive $[\![O_1 \textbf{ then } O_2]\!]^M_\Gamma = (\Sigma_1 \cup \Sigma_2, \mathcal{M}')$
where

- $[\![O_1]\!]^M_\Gamma = (\Sigma_1, \mathcal{M}_1)$
- $[\![O_2]\!]_{basic} = (\Sigma_2, \Psi_2)$
- $\mathcal{M}' = \{M \in \textbf{Mod}(\Sigma_1 \cup \Sigma_2) \mid M \models \Psi_2, M|_{\Sigma_1} \in \mathcal{M}_1\}$

Semantics of extensions (cont'd)

%mcons (%def, %wdef %mono) leads to the additional requirement that

> *each model in $\mathcal{M}_1$ has some (a unique, at most one, unique up to isomorphism) $\Sigma_1 \cup \Sigma_2$-expansion to a model in $\mathcal{M}'$.*

%implies leads to the additional requirements that

> $\Sigma_2 \subseteq \Sigma_1$ *and* $\mathcal{M}' = \mathcal{M}_1$.

%ccons leads to the additional requirement that

> $\mathcal{M}' \models \varphi$ *implies* $\mathcal{M}_1 \models \varphi$ *for any* Σ_1*-sentence* φ.

Theorem

%mcons implies %ccons, but not vice versa.

DOL Unions

A **union** of two theories that simultaneously renames and unifies the vocabulary.

```
logic OWL2.QL
ontology theory2 =
theory7 with Con |- > C
 and
theory11 with Co |- > C
```

Semantics of unions

O_1, O_2 flattenable $[\![O_1 \textbf{ and } O_2]\!]^T_\Gamma = (\Sigma_1 \cup \Sigma_2, \Psi_1 \cup \Psi_2)$, where

- $[\![O_i]\!]^T_\Gamma = (\Sigma_i, \Psi_i)$ $(i = 1, 2)$

one of O_1, O_2 elusive $[\![O_1 \textbf{ and } O_2]\!]^M_\Gamma = (\Sigma_1 \cup \Sigma_2, \mathcal{M})$, where

- $[\![O_i]\!]^M_\Gamma = (\Sigma_i, \mathcal{M}_i)$ $(i = 1, 2)$
- $\mathcal{M} = \{M \in \textbf{Mod}(\Sigma_1 \cup \Sigma_2) \mid M|_{\Sigma_i} \in \mathcal{M}_i, i = 1, 2\}$

DOL Logic Translation

An OWL theory can be **translated** to FOL and be further extended with FOL axioms.

```
logic CASL.FOL
ontology theory6_plus_antisym_and_WS =
theory6 with translation OWL22CASL
then
forall x,y:Thing . P(x,y) /\ P(y,x) => x =y %%(t3)
forall x,y:Thing . not P(y,x) =>
       exists z:Thing . P(z,y) /\ not O(z,x) %%(t4)
```

Semantics of heterogeneous translations

O flattenable Let $[\![O]\!]_\Gamma^T = (I, \Sigma, \Psi)$

- homogeneous translation
  $[\![O \textbf{ with } \sigma : \Sigma \to \Sigma']\!]_\Gamma^T = (I, \Sigma', \sigma(\Psi))$
- heterogeneous translation
  $[\![O \textbf{ with translation } \rho : I \to I']\!]_\Gamma^T = (I', \rho^{Sig}(\Sigma), \rho^{Sen}(\Psi))$

O elusive Let $[\![O]\!]_\Gamma^M = (I, \Sigma, \mathcal{M})$

- homogeneous translation
  $[\![O \textbf{ with } \sigma : \Sigma \to \Sigma']\!]_\Gamma^M = (I, \Sigma', \mathcal{M}')$
  where $\mathcal{M}' = \{M \in \textbf{Mod}(\Sigma') \mid M|_\sigma \in \mathcal{M}\}$
- heterogeneous translation
  $[\![O \textbf{ with translation } \rho : I \to I']\!]_\Gamma^M = (I', \rho^{Sig}(\Sigma), \mathcal{M}')$ where
  $\mathcal{M}' = \{M \in \textbf{Mod}^{I'}(\rho^{Sig}(\Sigma)) \mid \rho^{Mod}(M) \in \mathcal{M}\}$

DOL Definitional Extension

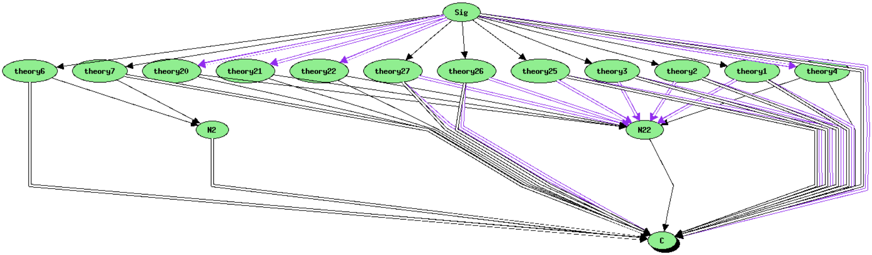
New symbols can be **definitionally** introduced.

```
logic CASL.FOL
ontology theory20 =
theory6_plus_antisym_and_WS
 then %wdef
. forall x,y:Thing . O(x,y) <=> exists z:Thing  (P(z,x)
                                              /\ P(z,y)) %(t21)
. forall x,y:Thing . EQ(x,y) <=> P(x,y) /\ P(y,x) %(t22)
```

DOL Networks

An entire **network** can be specified by naming the corresponding theories and mappings between them.

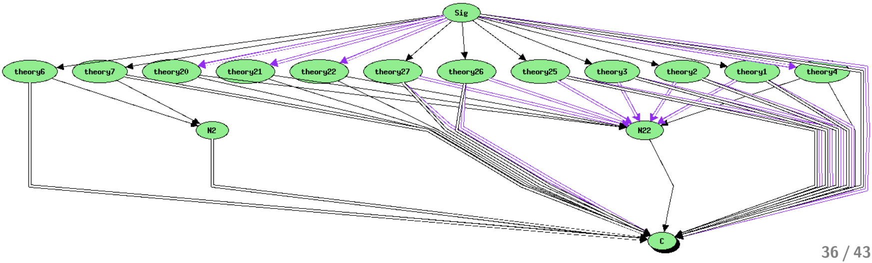
```
network KGEMT_network = theory1,...,theory28,M1,...,Mp
```



DOL Combinations of Networks (Colimits)

The **combination of a network** computes the integration of all participating nodes and reflecting the semantics of the mappings. Technically it is realised via a categorical colimit operation. For instance, the network related to two of our theories (N2 in OWL and N22 in heterog. FOL) is as follows:

```
ontology KGEMT = combine KGEMT_network
```



DOL: Keeping track of lemmas

Implied consequences can be recorded as lemmas (generating an **obligation for proof**).

```
logic CASL.FOL
ontology theory20_with_PP_lemmas =
theory20
 then %wdef
. forall x,y: Thing . PP(x,y) <=> P(x,y) /\ not P(y,x)
then  %implies
.forall x,y: Thing . not PP(x,x) %(t25)
.forall x,y,z: Thing . PP(x,y) /\ P(y,z) => PP(x,z) %(t26)
.forall x,y,z: Thing . PP(x,y) => not PP(y,z) %(t27)
```

DOL: Asserting consistency

Consistency as well as **countermodels** can be asserted by specifying models resp. countermodels, and by interpreting the theory resp. negation of claim, into these models.

```
logic CASL.FOL
interpretation Cons : theory23 to M = C |-> co, E |-> e
```

Tool support: Ontohub web portal and repository

Ontohub is a web-based repository engine for distributed heterogeneous (multi-language) OMS

web-based
: prototype available at `ontohub.org`

multi-logic
: speaks the same languages as Hets

multiple repositories
: ontologies can be organized in multiple repositories, each with its own management of editing and ownership rights,

Git interface
: version control of ontologies is supported via interfacing the Git version control system,

linked-data compliant
: one and the same URL is used for referencing an ontology, downloading it (for use with tools), and for user-friendly presentation in the browser.

DOL and OWL Classifier Resources

- http://dol-omg.org Central page for DOL
- http://hets.eu Analysis and Proof Tool Hets, speaking DOL
- http://ontohub.org Ontohub web platform, speaking DOL
- http://ontohub.org/dol-examples DOL examples
- http://ontoiop.org Initial standardization initiative
- https://ontohub.org/esslli-2016 ESSLLI course and repository of DOL examples
- https://keet.wordpress.com/2016/06/19/an-exhaustive-owl-species-classifier/

Summary

- We organised a heterogeneous network of 28 mereo(topo)logical theories into a structured network
- We employed the DOL language to allow for logical heterogeneity and to solve problems of alignment and extensions
- We gave mechanism to record lemmas and couterexamples
- We showed a tool to resolve OWL language feature conflicts
- Our framework is fully general and extensible

Outlook

Current and future work includes

- full specification and interlinking of a comprehensive library of mereo(topologies) ranging from lightweight DLs to full second-order logic
- collecting a library of important lemmas and counterexamples to help ontology developers
- develop the species detection and conflict resolution into a Protege plugin and integrate with debugging workflows
- extend to cover incompatible extensions and apply the theory of coherence to find maximally coherent subnetworks

References

- C Maria Keet, Oliver Kutz. Orchestrating a Network of Mereo(topo)logical Theories, Proceedings of the Knowledge Capture Conference, K-CAP, ACM, 2017.