Modularity for Ontologies: Theory and Practice

Yevgeny Kazakov
(based on joint works with Bernardo Cuenca Grau, Ian Horrocks and Ulrike Sattler)

The University of Oxford

November 20, 2007

Outline

Ontologies and Ontology Languages

- Ontologies are vocabularies of terms for specific subjects
  - chemical elements
  - genes
  - human anatomy
  - clinical procedures

$$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$

$$\mathrm{O\_Id7894} : \mathsf{Heart}$$

Ontologies and Ontology Languages

- Two types of axoioms

$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$

$O\_Id7894 : \text{Heart}$

Ontologies and Ontology Languages

- Two types of axoioms
  - Terminalogical axiom [Schema]

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf.CirculatorySystem}$$

$$O\_Id7894 : \text{Heart}$$

Ontologies and Ontology Languages

- Two types of axoioms
  - Terminalogical axiom [Schema]
  - Assertions [Data]

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$

$$O\_Id7894 : \text{Heart}$$

Ontologies and Ontology Languages

- The syntax of DL-based ontology languages

$$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \exists\, \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$

$$\mathrm{O\_Id7894} : \mathsf{Heart}$$

Ontologies and Ontology Languages

- The syntax of DL-based ontology languages
  - Atomic concepts [Classes]

$$\boxed{\text{Heart}} \equiv \boxed{\text{MuscularOrgan}} \sqcap \exists \text{isPartOf}.\boxed{\text{CirculatorySystem}}$$

$$\text{O\_Id7894} : \boxed{\text{Heart}}$$

Ontologies and Ontology Languages

- The syntax of DL-based ontology languages
  - Atomic concepts [Classes]
  - Roles [Properties]

$$\mathrm{Heart} \equiv \mathrm{MuscularOrgan} \sqcap \exists \boxed{\mathrm{isPartOf}.}\mathrm{CirculatorySystem}$$

$$O\_Id7894 : \mathrm{Heart}$$

Ontologies and Ontology Languages

- The syntax of DL-based ontology languages
  - Atomic concepts [Classes]
  - Roles [Properties]
  - Individuals

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$

$$\boxed{\text{O\_Id7894}} : \text{Heart}$$

Ontologies and Ontology Languages

- The syntax of DL-based ontology languages
  - Atomic concepts [Classes]
  - Roles [Properties]
  - Individuals
  - Constructors

$$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$

$$\mathrm{O\_Id7894} : \mathsf{Heart}$$

Ontologies and Ontology Languages

- The syntax of DL-based ontology languages
  - Atomic concepts [Classes]
  - Roles [Properties]
  - Individuals
  - Constructors

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$

O_Id7894 : Heart

OWL syntax:
(XML+RDF)

```
<owl:Class rdf:ID="Heart">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:ID="MuscularOrgan">
          <owl:Restriction>
            <owl:onProperty>
               <owl:ObjectProperty rdf:ID="isPartOf">
            </owl:onProperty>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="CirculatorySystem"/>
            </owl:someValuesFrom>
          </owl:Restriction>
        </owl:Class>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
```

Ontologies and Ontology Languages

- The set-theoretic semantics for ontology languages

$$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$

$$\mathrm{O\_Id7894} : \mathsf{Heart}$$

Ontologies and Ontology Languages

- The set-theoretic semantics for ontology languages
  - Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$

$$\text{O\_Id7894} : \text{Heart}$$

Ontologies and Ontology Languages

- The set-theoretic semantics for ontology languages
  - Interpretation $\mathcal{I} = (\boxed{\Delta^{\mathcal{I}},} \cdot^{\mathcal{I}})$
    - $\Delta^{\mathcal{I}}$ is an interpretation domain (non-empty set)

Heart $\equiv$ MuscularOrgan $\sqcap$ $\exists$isPartOf.CirculatorySystem

O_Id7894 : Heart

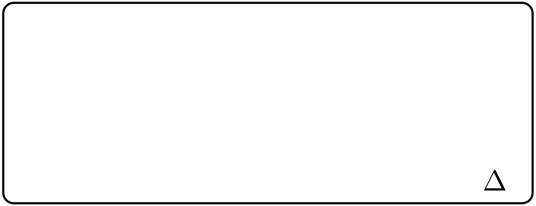
Δ

Ontologies and Ontology Languages

- The set-theoretic semantics for ontology languages
  - Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \boxed{\cdot^{\mathcal{I}}})$
    - $\Delta^{\mathcal{I}}$ is an interpretation domain (non-empty set)
    - $\cdot^{\mathcal{I}}$ is an interpretation function

$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$

$\text{O\_Id7894} : \text{Heart}$



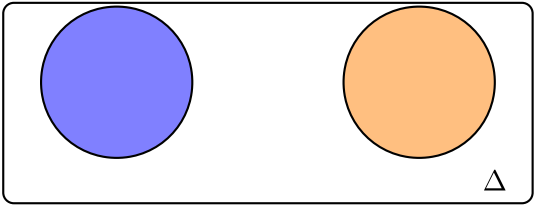
Ontologies and Ontology Languages

- The set-theoretic semantics for ontology languages
  - Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$
    - $\Delta^{\mathcal{I}}$ is an interpretation domain (non-empty set)
    - $\cdot^{\mathcal{I}}$ is an interpretation function

      Atomic concepts $\Rightarrow$ sets

$$\text{Heart} \equiv \boxed{\text{MuscularOrgan}} \sqcap \exists \text{isPartOf}.\boxed{\text{CirculatorySystem}}$$

$$\text{O\_Id7894} : \text{Heart}$$



Ontologies and Ontology Languages

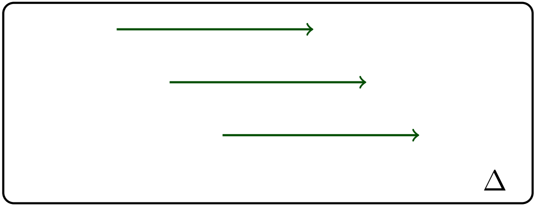
- The set-theoretic semantics for ontology languages
  - Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$
    - $\Delta^{\mathcal{I}}$ is an interpretation domain (non-empty set)
    - $\cdot^{\mathcal{I}}$ is an interpretation function

      Atomic concepts $\Rightarrow$ sets

      Roles $\Rightarrow$ binary relations

Heart $\equiv$ MuscularOrgan $\sqcap$ $\exists$isPartOf.CirculatorySystem

O_Id7894 : Heart



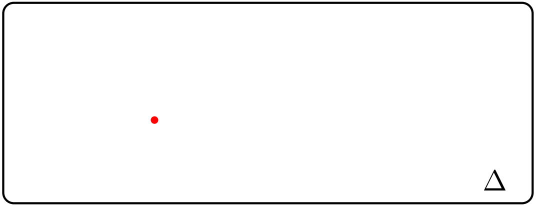
Ontologies and Ontology Languages

- The set-theoretic semantics for ontology languages
  - Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$
    - $\Delta^{\mathcal{I}}$ is an interpretation domain (non-empty set)
    - $\cdot^{\mathcal{I}}$ is an interpretation function

      Atomic concepts $\Rightarrow$ sets
      Roles $\Rightarrow$ binary relations
      Individuals $\Rightarrow$ elements

Heart $\equiv$ MuscularOrgan $\sqcap$ $\exists$isPartOf.CirculatorySystem

$\boxed{\text{O\_Id7894}}$: Heart

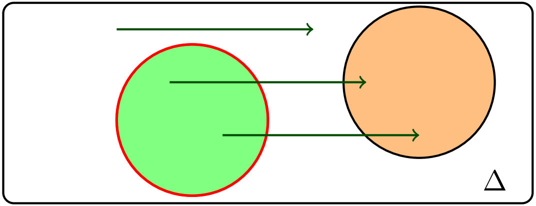


Ontologies and Ontology Languages

- The set-theoretic semantics for ontology languages
  - Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$
    - $\Delta^{\mathcal{I}}$ is an interpretation domain (non-empty set)
    - $\cdot^{\mathcal{I}}$ is an interpretation function
    - Constructors $\Rightarrow$ set operators

$$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \boxed{\exists\, \mathsf{isPartOf}.\mathsf{CirculatorySystem}}$$

$$\mathrm{O\_Id7894} : \mathsf{Heart}$$

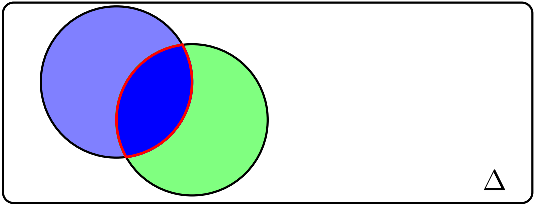


Ontologies and Ontology Languages

- The set-theoretic semantics for ontology languages
  - Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$
    - $\Delta^{\mathcal{I}}$ is an interpretation domain (non-empty set)
    - $\cdot^{\mathcal{I}}$ is an interpretation function
    - Constructors $\Rightarrow$ set operators

$\text{Heart} \equiv \boxed{\text{MuscularOrgan} \sqcap \exists \text{isPartOf.CirculatorySystem}}$

$\text{O\_Id7894} : \text{Heart}$

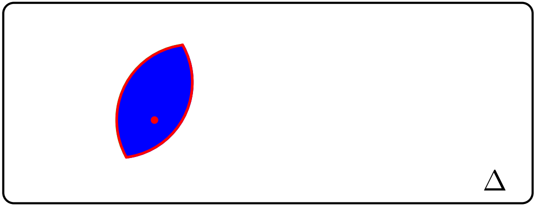


Ontologies and Ontology Languages

- The set-theoretic semantics for ontology languages
  - Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$
    - $\Delta^{\mathcal{I}}$ is an interpretation domain (non-empty set)
    - $\cdot^{\mathcal{I}}$ is an interpretation function
    - Constructors $\Rightarrow$ set operators
    - $\mathcal{I}$ is a model iff all axioms hold

$$\boxed{\mathsf{Heart}} \equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$

$$\boxed{\mathrm{O\_Id7894} : \mathsf{Heart}}$$



Ontologies and First-Order Logic

$$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap$$
$$\sqcap\ \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$

$$\mathrm{O\_Id7894} : \mathsf{Heart}$$

- Translation to the first-order logic:
  - Concept names — Heart, CirculatorySystem
  - Role names — isPartOf
  - Individuals — O_Id7894
  - Operators — $C_1 \sqcap C_2$, $\exists r.C$

Ontologies and First-Order Logic

$$\text{Heart} \sqsubseteq \text{MuscularOrgan} \sqcap$$
$$\sqcap\ \exists \text{isPartOf}.\text{CirculatorySystem}$$

$$O\_Id7894 : \text{Heart}$$

- Translation to the first-order logic:
  - Concept names — Heart, CirculatorySystem
  - Role names — isPartOf
  - Individuals — O_Id7894
  - Operators — $C_1 \sqcap C_2$, $\exists r.C$

Ontologies and First-Order Logic

$$\boxed{\text{Heart}(x)} \sqsubseteq \text{MuscularOrgan}(x) \sqcap$$
$$\sqcap\ \exists\text{isPartOf}.\text{CirculatorySystem}(x)$$

$$O\_Id7894 : \text{Heart}(x)$$

- Translation to the first-order logic:
  - Concept names — Heart, CirculatorySystem
    $\rightsquigarrow$ unary atoms: — Heart(x), CirculatorySystem(x)
  - Role names — isPartOf
  - Individuals — O_Id7894
  - Operators — $C_1 \sqcap C_2$, $\exists r.C$

ONTOLOGIES AND FIRST-ORDER LOGIC

$$\text{Heart}(x) \equiv \text{MuscularOrgan}(x) \sqcap$$
$$\sqcap\ \boxed{\exists \text{isPartOf}.}\text{CirculatorySystem}(x)$$

$$\text{O\_Id7894} : \text{Heart}(x)$$

- Translation to the first-order logic:
  - Concept names — Heart, CirculatorySystem
    $\rightsquigarrow$ unary atoms: — $\text{Heart}(x)$, $\text{CirculatorySystem}(x)$
  - Role names — isPartOf
  - Individuals — O_Id7894
  - Operators — $C_1 \sqcap C_2$, $\exists r.C$

Ontologies and First-Order Logic

$\text{Heart}(x) \equiv \text{MuscularOrgan}(x) \sqcap$
$\sqcap \boxed{\exists \text{isPartOf}(x, y)}.\text{CirculatorySystem}(x)$

$\text{O\_Id7894} : \text{Heart}(x)$

- Translation to the first-order logic:
  - Concept names $\rightsquigarrow$ unary atoms: Heart, CirculatorySystem; $\text{Heart}(x)$, $\text{CirculatorySystem}(x)$
  - Role names $\rightsquigarrow$ binary atoms: isPartOf; $\text{isPartOf}(x, y)$
  - Individuals: O_Id7894
  - Operators: $C_1 \sqcap C_2$, $\exists r.C$

Ontologies and First-Order Logic

$$\text{Heart}(x) \equiv \text{MuscularOrgan}(x) \sqcap$$
$$\sqcap\ \exists \text{isPartOf}(x,y).\text{CirculatorySystem}(x)$$

$\boxed{\text{O\_Id7894}}$: $\text{Heart}(x)$

- Translation to the first-order logic:
  - Concept names — Heart, CirculatorySystem
    ⇝ unary atoms: — $\text{Heart}(x)$, $\text{CirculatorySystem}(x)$
  - Role names — isPartOf
    ⇝ binary atoms: — $\text{isPartOf}(x,y)$
  - Individuals — O_Id7894
  - Operators — $C_1 \sqcap C_2$, $\exists r.C$

Ontologies and First-Order Logic

$\mathsf{Heart}(x) \equiv \mathsf{MuscularOrgan}(\mathsf{x}) \sqcap$
$\sqcap \exists \mathsf{isPartOf}(x,y).\mathsf{CirculatorySystem}(\mathsf{x})$

$\mathsf{Heart}(\mathrm{O\_Id7894})$

- Translation to the first-order logic:
  - Concept names $\rightsquigarrow$ unary atoms: Heart, CirculatorySystem — $\mathsf{Heart}(x)$, $\mathsf{CirculatorySystem}(x)$
  - Role names $\rightsquigarrow$ binary atoms: isPartOf — $\mathsf{isPartOf}(x,y)$
  - Individuals $\rightsquigarrow$ constants: O_Id7894 — O_Id7894
  - Operators $C_1 \sqcap C_2$, $\exists r.C$

Ontologies and First-Order Logic

$$\mathsf{Heart}(x) \equiv \mathsf{MuscularOrgan}(x) \sqcap$$
$$\sqcap\ \exists \mathsf{isPartOf}(x,y).\mathsf{CirculatorySystem}(x)$$

$$\mathsf{Heart}(\mathrm{O\_Id7894})$$

- Translation to the first-order logic:
  - Concept names — Heart, CirculatorySystem
    ⇝ unary atoms: $\mathsf{Heart}(x)$, $\mathsf{CirculatorySystem}(x)$
  - Role names — isPartOf
    ⇝ binary atoms: $\mathsf{isPartOf}(x,y)$
  - Individuals — O_Id7894
    ⇝ constants: O_Id7894
  - Operators — $C_1 \sqcap C_2$, $\exists r.C$

ONTOLOGIES AND FIRST-ORDER LOGIC

$$\mathsf{Heart}(x) \equiv \mathsf{MuscularOrgan}(x) \land$$
$$\land\ \exists y.[\mathsf{isPartOf}(x,y) \land \mathsf{CirculatorySystem}(x)]$$
$$\mathsf{Heart}(\mathrm{O\_Id7894})$$

- Translation to the first-order logic:
  - Concept names $\rightsquigarrow$ unary atoms: Heart, CirculatorySystem — $\mathsf{Heart}(x)$, $\mathsf{CirculatorySystem}(x)$
  - Role names $\rightsquigarrow$ binary atoms: isPartOf — $\mathsf{isPartOf}(x,y)$
  - Individuals $\rightsquigarrow$ constants: O_Id7894 — O_Id7894
  - Operators $\rightsquigarrow$ constructors: $C_1 \sqcap C_2$, $\exists r.C$ — $C_1(x) \land C_2(x)$, $\exists y.[r(x,y) \land C(y)]$

Ontologies and First-Order Logic

$$\text{Heart}(x) \equiv \text{MuscularOrgan}(x) \wedge$$
$$\wedge \exists y.[\text{isPartOf}(x,y) \wedge \text{CirculatorySystem}(x)]$$

$$\text{Heart}(\text{O\_Id7894})$$

- Translation to the first-order logic:
  - Concept names — Heart, CirculatorySystem
    $\rightsquigarrow$ unary atoms: $\text{Heart}(x)$, $\text{CirculatorySystem}(x)$
  - Role names — isPartOf
    $\rightsquigarrow$ binary atoms: $\text{isPartOf}(x,y)$
  - Individuals — O_Id7894
    $\rightsquigarrow$ constants: O_Id7894
  - Operators — $C_1 \sqcap C_2$, $\exists r.C$
    $\rightsquigarrow$ constructors: $C_1(x) \wedge C_2(x)$, $\exists y.[r(x,y) \wedge C(y)]$

A Hierarchy of Ontology Languages

Name	DL syntax	First-Order syntax	
intersection	$C_1 \sqcap C_2$	$C_1(x) \wedge C_2(x)$	
union	$C_1 \sqcup C_2$	$C_1(x) \vee C_2(x)$	$= \mathcal{A}$
complement	$\neg C$	$\neg C(x)$	$\mathcal{L}$
value restriction	$\forall r.C$	$\forall y.[r(x,y) \rightarrow C(y)]$	$\mathcal{C}$
exist restriction	$\exists r.C$	$\exists y.[r(x,y) \wedge C(y)]$	
concept assertion	$i : C$	$C(i)$	
role assertion	$(i_1, i_2) : r$	$r(i_1, i_2)$	

A Hierarchy of Ontology Languages

Name	DL syntax	First-Order syntax	
intersection	$C_1 \sqcap C_2$	$C_1(x) \wedge C_2(x)$	
union	$C_1 \sqcup C_2$	$C_1(x) \vee C_2(x)$	$= \mathcal{A}$
complement	$\neg C$	$\neg C(x)$	$\mathcal{L}$
value restriction	$\forall r.C$	$\forall y.[r(x,y) \to C(y)]$	$\mathcal{C}$
exist restriction	$\exists r.C$	$\exists y.[r(x,y) \wedge C(y)]$	
concept assertion	$i : C$	$C(i)$	
role assertion	$(i_1, i_2) : r$	$r(i_1, i_2)$	
transitivity	$Trans(r)$	$\forall xyz.[r(x,y) \wedge r(y,z) \to r(x,z)]$	$= \mathcal{S}$
functionality	$Funct(r)$	$\forall xyz.[r(x,y) \wedge r(x,z) \to y \simeq z]$	$+\mathcal{F}$
role inclusion	$r_1 \sqsubseteq r_2$	$\forall xy.[r_1(x,y) \to r_2(x,y)]$	$+\mathcal{H}$

A Hierarchy of Ontology Languages

Name	DL syntax	First-Order syntax	
intersection	$C_1 \sqcap C_2$	$C_1(x) \wedge C_2(x)$	
union	$C_1 \sqcup C_2$	$C_1(x) \vee C_2(x)$	$= \mathcal{A}$
complement	$\neg C$	$\neg C(x)$	$\mathcal{L}$
value restriction	$\forall r.C$	$\forall y.[r(x,y) \rightarrow C(y)]$	$\mathcal{C}$
exist restriction	$\exists r.C$	$\exists y.[r(x,y) \wedge C(y)]$	
concept assertion	$i : C$	$C(i)$	
role assertion	$(i_1, i_2) : r$	$r(i_1, i_2)$	
transitivity	$Trans(r)$	$\forall xyz.[r(x,y) \wedge r(y,z) \rightarrow r(x,z)]$	$= \mathcal{S}$
functionality	$Funct(r)$	$\forall xyz.[r(x,y) \wedge r(x,z) \rightarrow y \simeq z]$	$+\mathcal{F}$
role inclusion	$r_1 \sqsubseteq r_2$	$\forall xy.[r_1(x,y) \rightarrow r_2(x,y)]$	$+\mathcal{H}$
inverse roles	$[\ldots]r^-[\ldots]$	$[\ldots]r(y,x)[\ldots]$	$+\mathcal{I}$

A Hierarchy of Ontology Languages

Name	DL syntax	First-Order syntax	
intersection	$C_1 \sqcap C_2$	$C_1(x) \wedge C_2(x)$	
union	$C_1 \sqcup C_2$	$C_1(x) \vee C_2(x)$	$= \mathcal{A}$
complement	$\neg C$	$\neg C(x)$	$\mathcal{L}$
value restriction	$\forall r.C$	$\forall y.[r(x,y) \rightarrow C(y)]$	$\mathcal{C}$
exist restriction	$\exists r.C$	$\exists y.[r(x,y) \wedge C(y)]$	
concept assertion	$i : C$	$C(i)$	
role assertion	$(i_1, i_2) : r$	$r(i_1, i_2)$	
transitivity	$Trans(r)$	$\forall xyz.[r(x,y) \wedge r(y,z) \rightarrow r(x,z)]$	$= \mathcal{S}$
functionality	$Funct(r)$	$\forall xyz.[r(x,y) \wedge r(x,z) \rightarrow y \simeq z]$	$+\mathcal{F}$
role inclusion	$r_1 \sqsubseteq r_2$	$\forall xy.[r_1(x,y) \rightarrow r_2(x,y)]$	$+\mathcal{H}$
inverse roles	$[\ldots]r^-[\ldots]$	$[\ldots]r(y,x)[\ldots]$	$+\mathcal{I}$
number restriction	$\leqslant n\, r$	$\exists^{\leq n} y.r(x,y)$	$+\mathcal{N}$
qualified nr. restr.	$\leqslant n\, r.C$	$\exists^{\leq n} y.[r(x,y) \wedge C(y)]$	$+\mathcal{Q}$

A Hierarchy of Ontology Languages

Name	DL syntax	First-Order syntax	
intersection	$C_1 \sqcap C_2$	$C_1(x) \wedge C_2(x)$	
union	$C_1 \sqcup C_2$	$C_1(x) \vee C_2(x)$	$= \mathcal{A}$
complement	$\neg C$	$\neg C(x)$	$\mathcal{L}$
value restriction	$\forall r.C$	$\forall y.[r(x,y) \rightarrow C(y)]$	$\mathcal{C}$
exist restriction	$\exists r.C$	$\exists y.[r(x,y) \wedge C(y)]$	
concept assertion	$i : C$	$C(i)$	
role assertion	$(i_1, i_2) : r$	$r(i_1, i_2)$	
transitivity	$Trans(r)$	$\forall xyz.[r(x,y) \wedge r(y,z) \rightarrow r(x,z)]$	$= \mathcal{S}$
functionality	$Funct(r)$	$\forall xyz.[r(x,y) \wedge r(x,z) \rightarrow y \simeq z]$	$+\mathcal{F}$
role inclusion	$r_1 \sqsubseteq r_2$	$\forall xy.[r_1(x,y) \rightarrow r_2(x,y)]$	$+\mathcal{H}$
inverse roles	$[\ldots]r^-[\ldots]$	$[\ldots]r(y,x)[\ldots]$	$+\mathcal{I}$
number restriction	$\leqslant n\, r$	$\exists^{\leq n} y.r(x,y)$	$+\mathcal{N}$
qualified nr. restr.	$\leqslant n\, r.C$	$\exists^{\leq n} y.[r(x,y) \wedge C(y)]$	$+\mathcal{Q}$
nominals	$\{i\}$	$x \simeq i$	$+\mathcal{O}$

e.g. OWL DL $\rightsquigarrow$ $\mathcal{SHOIN}$

Reasoning in Ontologies

$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$

$\mathsf{MuscularOrgan} \equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{MuscularSystem}$

$\mathsf{CardiovascularOrgan} \equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$

$\mathrm{O\_Id7894} : \mathsf{Heart}$

- Ontology reasoning:
  – extracting implicit information from the explicit information in ontologies

Reasoning in Ontologies

$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$

$\text{MuscularOrgan} \equiv \text{Organ} \sqcap \exists \text{isPartOf}.\text{MuscularSystem}$

$\text{CardiovascularOrgan} \equiv \text{Organ} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$

$O\_Id7894 : \text{Heart}$

- Ontology reasoning:
  – extracting implicit information from the explicit information in ontologies
  - $\text{Heart} \sqsubseteq \text{CardiovascularOrgan}$

Reasoning in Ontologies

$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$

$\mathsf{MuscularOrgan} \equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{MuscularSystem}$

$\mathsf{CardiovascularOrgan} \equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$

$O\_Id7894 : \mathsf{Heart}$

- Ontology reasoning:
  – extracting implicit information from the explicit information in ontologies
  - $\mathsf{Heart} \sqsubseteq \mathsf{CardiovascularOrgan}$
  - $O\_Id7894 : \exists \mathsf{isPartOf}.(\mathsf{MuscularSystem} \sqcup \mathsf{CirculatorySystem})$

Reasoning in Ontologies

$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$

$\mathsf{MuscularOrgan} \equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{MuscularSystem}$

$\mathsf{CardiovascularOrgan} \equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$

$O\_Id7894 : \mathsf{Heart}$

- Ontology reasoning:
  – extracting implicit information from the explicit information in ontologies
  - $\mathsf{Heart} \sqsubseteq \mathsf{CardiovascularOrgan}$
  - $O\_Id7894 : \exists \mathsf{isPartOf}.(\mathsf{MuscularSystem} \sqcup \mathsf{CirculatorySystem})$
- Standard reasoning tasks:

Reasoning in Ontologies

$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$

$\mathsf{MuscularOrgan} \equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{MuscularSystem}$

$\mathsf{CardiovascularOrgan} \equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$

$O\_Id7894 : \mathsf{Heart}$

- Ontology reasoning:
  – extracting implicit information from the explicit information in ontologies
  - $\mathsf{Heart} \sqsubseteq \mathsf{CardiovascularOrgan}$
  - $O\_Id7894 : \exists \mathsf{isPartOf}.(\mathsf{MuscularSystem} \sqcup \mathsf{CirculatorySystem})$
- Standard reasoning tasks:
  - Classification:
    – compute all subsumptions $A \sqsubseteq B$ between named classes

Reasoning in Ontologies

$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$

$\text{MuscularOrgan} \equiv \text{Organ} \sqcap \exists \text{isPartOf}.\text{MuscularSystem}$

$\text{CardiovascularOrgan} \equiv \text{Organ} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$

$\text{O\_Id7894} : \text{Heart}$

- Ontology reasoning:
  – extracting implicit information from the explicit information in ontologies
  - $\text{Heart} \sqsubseteq \text{CardiovascularOrgan}$
  - $\text{O\_Id7894} : \exists \text{isPartOf}.(\text{MuscularSystem} \sqcup \text{CirculatorySystem})$
- Standard reasoning tasks:
  - Classification:
    – compute all subsumptions $A \sqsubseteq B$ between <u>named</u> classes
  - Instance retrieval:
    – compute all implicit instances i of a class C.

Reasoning in Ontologies

$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$

$\text{MuscularOrgan} \equiv \text{Organ} \sqcap \exists \text{isPartOf}.\text{MuscularSystem}$

$\text{CardiovascularOrgan} \equiv \text{Organ} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$

$O\_Id7894 : \text{Heart}$

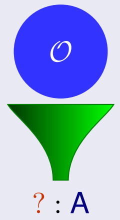
- Ontology reasoning:
  – extracting implicit information from the explicit information in ontologies
  - $\text{Heart} \sqsubseteq \text{CardiovascularOrgan}$
  - $O\_Id7894 : \exists \text{isPartOf}.(\text{MuscularSystem} \sqcup \text{CirculatorySystem})$
- Standard reasoning tasks:
  - Classification:
    – compute all subsumptions $A \sqsubseteq B$ between named classes
  - Instance retrieval:
    – compute all implicit instances i of a class C.
- Ontology reasoners: FaCT++, KAON2, Pellet, Racer, CEL,

Outline

Classical Reasoning Support for Ontologies

- Provides engine for querying of ontologies

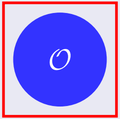
Ontology



CLASSICAL REASONING SUPPORT FOR ONTOLOGIES

- Provides engine for querying of ontologies
- Provides tools for ontology development:

ONTOLOGY



Classical Reasoning Support for Ontologies

- Provides engine for querying of ontologies
- Provides tools for ontology development:
  - ✔ Checking global consistency

Ontology

$\mathcal{O}$

$$\begin{array}{l} \text{Male} \sqcap \text{Female} \sqsubseteq \bot \\ \text{Sam} : \text{Male} \\ \text{Sam} : \text{Female} \\ \hline \models \bot \end{array}$$

Classical Reasoning Support for Ontologies

- Provides engine for querying of ontologies
- Provides tools for ontology development:
  - ✔ Checking global consistency
  - ✔ Detecting unsatisfiable classes

Ontology

$$\begin{array}{l} \text{Male} \sqcap \text{Female} \sqsubseteq \bot \\ \text{Hermaphrodite} \sqsubseteq \\ \qquad \text{Male} \sqcap \text{Female} \\ \hline \models \mathit{Hermaphrodite} \sqsubseteq \bot \end{array}$$

Classical Reasoning Support for Ontologies

- Provides engine for querying of ontologies
- Provides tools for ontology development:
  - ✔ Checking global consistency
  - ✔ Detecting unsatisfiable classes
  - ✔ Detecting unintended subsumptions

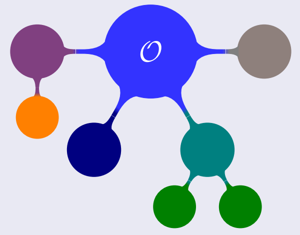
Ontology

$$\begin{array}{l} \text{Man} \equiv \text{Male} \sqcup \text{Female} \\ \text{Man} \equiv \text{Male} \\ \hline \models \mathit{Female} \sqsubseteq \mathit{Male} \end{array}$$

Classical Reasoning Support for Ontologies

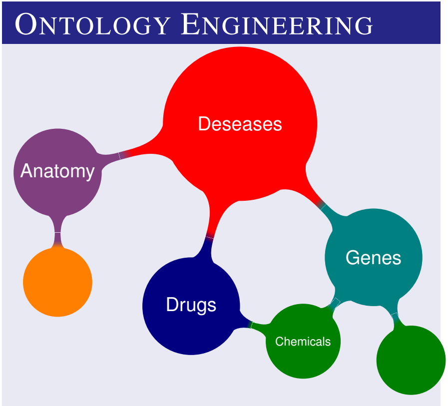
- Provides engine for querying of ontologies
- Provides tools for ontology development:
  - ✔ Checking global consistency
  - ✔ Detecting unsatisfiable classes
  - ✔ Detecting unintended subsumptions
- Not sufficient for large-scale ontology development

Ontology



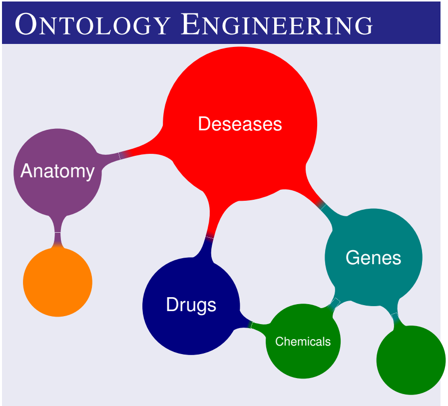
Ontology Engineering at the Large Scale

- Collaborative development
- Involves experts in different fields
- Continuous process
- The notion of modularity becomes apparent



Ontology Engineering at the Large Scale

- Collaborative development
- Involves experts in different fields
- Continuous process
- The notion of modularity becomes apparent
- Problems:
  - ✔ Safe integration of ontologies
  - ✔ Partial ontology reuse



A Motivating Example

Ontology Reuse

Ontology of Research Projects

$\text{CysticFibrosis\_EUProject} \equiv$

$\qquad \text{EUProject} \sqcap \exists \text{hasFocus}.\text{CysticFibrosis}$

$\text{GeneticDisorder\_Project} \equiv$

$\qquad \text{Project} \sqcap \exists \text{hasFocus}.\text{GeneticDisorder}$

$\text{EUProject} \sqsubseteq \text{Project}$

A Motivating Example

Ontology Reuse

Ontology of Research Projects

$\text{CysticFibrosis\_EUProject} \equiv$
$\text{EUProject} \sqcap \exists \text{hasFocus}.\text{CysticFibrosis}$

$\text{GeneticDisorder\_Project} \equiv$
$\text{Project} \sqcap \exists \text{hasFocus}.\text{GeneticDisorder}$

$\text{EUProject} \sqsubseteq \text{Project}$

A Motivating Example

Ontology of Medical Terms

GeneticDisorder $\equiv \ldots$

CysticFibrosis $\equiv \ldots$

Ontology of Research Projects

CysticFibrosis_EUProject $\equiv$

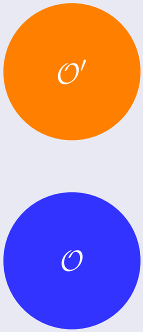
EUProject $\sqcap \exists$hasFocus.CysticFibrosis

GeneticDisorder_Project $\equiv$

Project $\sqcap \exists$hasFocus.GeneticDisorder

EUProject $\sqsubseteq$ Project

Ontology Reuse



A Motivating Example

Ontology of Medical Terms

GeneticDisorder $\equiv \ldots$

CysticFibrosis $\equiv \ldots$

Ontology of Research Projects

CysticFibrosis_EUProject $\equiv$

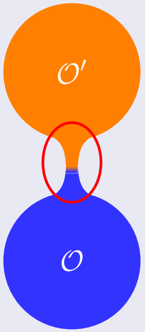
EUProject $\sqcap \exists$hasFocus.CysticFibrosis

GeneticDisorder_Project $\equiv$

Project $\sqcap \exists$hasFocus.GeneticDisorder

EUProject $\sqsubseteq$ Project

Ontology Reuse



A Motivating Example

Ontology of Medical Terms

$\textsf{GeneticDisorder} \equiv \ldots$

$\textsf{CysticFibrosis} \equiv \ldots$

$\models \quad \textsf{CysticFibrosis} \sqsubseteq \textsf{GeneticDisorder}$

Ontology of Research Projects

$\textsf{CysticFibrosis\_EUProject} \equiv$

$\qquad \textsf{EUProject} \sqcap \exists \textsf{hasFocus}.\textsf{CysticFibrosis}$

$\textsf{GeneticDisorder\_Project} \equiv$

$\qquad \textsf{Project} \sqcap \exists \textsf{hasFocus}.\textsf{GeneticDisorder}$

$\textsf{EUProject} \sqsubseteq \textsf{Project}$

Ontology Reuse

A Motivating Example

Ontology of Medical Terms

$\text{GeneticDisorder} \equiv \ldots$

$\text{CysticFibrosis} \equiv \ldots$

$\models \quad \text{CysticFibrosis} \sqsubseteq \text{GeneticDisorder}$

Ontology of Research Projects

$\text{CysticFibrosis\_EUProject} \equiv$
$\qquad \text{EUProject} \sqcap \exists \text{hasFocus}.\text{CysticFibrosis}$

$\text{GeneticDisorder\_Project} \equiv$
$\qquad \text{Project} \sqcap \exists \text{hasFocus}.\text{GeneticDisorder}$

$\boxed{\text{EUProject} \sqsubseteq \text{Project}}$

Ontology Reuse

A Motivating Example

Ontology of Medical Terms

$\mathsf{GeneticDisorder} \equiv \ldots$

$\mathsf{CysticFibrosis} \equiv \ldots$

$\models \boxed{\mathsf{CysticFibrosis} \sqsubseteq \mathsf{GeneticDisorder}}$

Ontology of Research Projects

$\mathsf{CysticFibrosis\_EUProject} \equiv$

$\mathsf{EUProject} \sqcap \exists \mathsf{hasFocus}.\mathsf{CysticFibrosis}$

$\mathsf{GeneticDisorder\_Project} \equiv$

$\mathsf{Project} \sqcap \exists \mathsf{hasFocus}.\mathsf{GeneticDisorder}$

$\mathsf{EUProject} \sqsubseteq \mathsf{Project}$

Ontology Reuse

A Motivating Example

Ontology of Medical Terms

GeneticDisorder $\equiv \ldots$

CysticFibrosis $\equiv \ldots$

$\models$ CysticFibrosis $\sqsubseteq$ GeneticDisorder

Ontology of Research Projects

CysticFibrosis_EUProject $\equiv$

EUProject $\sqcap$ $\exists$hasFocus.CysticFibrosis

GeneticDisorder_Project $\equiv$

Project $\sqcap$ $\exists$hasFocus.GeneticDisorder

EUProject $\sqsubseteq$ Project

$\models$ $\boxed{\text{CysticFibrosis\_EUProject} \sqsubseteq \text{GeneticDisorder\_Project}}$

Ontology Reuse

A Motivating Example

Ontology of Medical Terms

$\mathsf{GeneticDisorder} \equiv \ldots$

$\mathsf{CysticFibrosis} \equiv \ldots$

$\models \quad \mathsf{CysticFibrosis} \sqsubseteq \mathsf{GeneticDisorder}$

Ontology of Research Projects

$\mathsf{CysticFibrosis\_EUProject} \equiv$

$\qquad \mathsf{EUProject} \sqcap \exists \mathsf{hasFocus}.\mathsf{CysticFibrosis}$

$\mathsf{GeneticDisorder\_Project} \equiv$

$\qquad \mathsf{Project} \sqcap \exists \mathsf{hasFocus}.\mathsf{GeneticDisorder}$

$\mathsf{EUProject} \sqsubseteq \mathsf{Project}$

$\models \quad \mathsf{CysticFibrosis\_EUProject} \sqsubseteq \mathsf{GeneticDisorder\_Project}$

Ontology Reuse

Modelling Errors

Ontology of Research Projects

CysticFibrosis_EUProject $\equiv$ [...]

GeneticDisorder_Project $\equiv$ [...]

EUProject $\sqsubseteq$ Project

Ontology Reuse

Modelling Errors

Ontology of Research Projects

CysticFibrosis_EUProject $\equiv$ [...]

GeneticDisorder_Project $\equiv$ [...]

EUProject $\sqsubseteq$ Project

Ontology Reuse

"If something hasFocus *then it is a* Project*"*

Modelling Errors

Ontology of Research Projects

CysticFibrosis_EUProject $\equiv [\ldots]$

GeneticDisorder_Project $\equiv [\ldots]$

EUProject $\sqsubseteq$ Project

$\exists$hasFocus.$\top \sqsubseteq$ Project

Ontology Reuse

"If something hasFocus *then it is a* Project*"*

Modelling Errors

Ontology of Research Projects

CysticFibrosis_EUProject $\equiv$ [...]

GeneticDisorder_Project $\equiv$ [...]

EUProject $\sqsubseteq$ Project

$\exists$hasFocus.$\top$ $\sqsubseteq$ Project

Ontology Reuse

"Any instance of Project *is different from any instance of* CysticFibrosis *and any instance of* GeneticDisorder*"*

Modelling Errors

Ontology of Research Projects

$\text{CysticFibrosis\_EUProject} \equiv [\ldots]$

$\text{GeneticDisorder\_Project} \equiv [\ldots]$

$\text{EUProject} \sqsubseteq \text{Project}$

$\exists \text{hasFocus}.\top \sqsubseteq \text{Project}$

$\text{Project} \sqcap (\text{CysticFibrosis} \sqcap \text{GeneticDisorder}) \sqsubseteq \bot$

Ontology Reuse

"Any instance of Project *is different from any instance of* CysticFibrosis *and any instance of* GeneticDisorder*"*

Modelling Errors

Ontology of Research Projects

CysticFibrosis_EUProject $\equiv$ [...]

GeneticDisorder_Project $\equiv$ [...]

EUProject $\sqsubseteq$ Project

$\exists$hasFocus.$\top$ $\sqsubseteq$ Project

Project $\sqcap$
(CysticFibrosis $\sqcap$ GeneticDisorder) $\sqsubseteq$ $\bot$

Ontology Reuse

"Every instance of Project *that* hasFocus *on* CysticFibrosis, *also* hasFocus *on* GeneticDisorder*"*

Modelling Errors

Ontology of Research Projects

CysticFibrosis_EUProject $\equiv [\ldots]$

GeneticDisorder_Project $\equiv [\ldots]$

EUProject $\sqsubseteq$ Project

$\exists$hasFocus.$\top \sqsubseteq$ Project

Project $\sqcap$ (CysticFibrosis $\sqcap$ GeneticDisorder) $\sqsubseteq \bot$

$\forall$hasFocus.CysticFibrosis $\sqsubseteq$ $\exists$hasFocus.GeneticDisorder

Ontology Reuse

"Every instance of Project *that* hasFocus *on* CysticFibrosis, *also* hasFocus *on* GeneticDisorder*"*

MODELLING ERRORS

ONTOLOGY OF RESEARCH PROJECTS

$\mathsf{CysticFibrosis\_EUProject} \equiv [\ldots]$

$\mathsf{GeneticDisorder\_Project} \equiv [\ldots]$

$\mathsf{EUProject} \sqsubseteq \mathsf{Project}$

$\exists \mathsf{hasFocus}.\top \sqsubseteq \mathsf{Project}$

$\mathsf{Project} \sqcap (\mathsf{CysticFibrosis} \sqcap \mathsf{GeneticDisorder}) \sqsubseteq \bot$

$\forall \mathsf{hasFocus}.\mathsf{CysticFibrosis} \sqsubseteq \exists \mathsf{hasFocus}.\mathsf{GeneticDisorder}$

ONTOLOGY REUSE

"Any instance of Project *is different from any instance of* CysticFibrosis *and any instance of* GeneticDisorder*"*

Modelling Errors

Ontology of Research Projects

$\text{CysticFibrosis\_EUProject} \equiv [\ldots]$

$\text{GeneticDisorder\_Project} \equiv [\ldots]$

$\text{EUProject} \sqsubseteq \text{Project}$

$\exists \text{hasFocus}.\top \sqsubseteq \text{Project}$

$$\text{Project} \sqcap (\text{CysticFibrosis} \sqcap \text{GeneticDisorder}) \sqsubseteq \bot$$

$$\forall \text{hasFocus}.\text{CysticFibrosis} \sqsubseteq \exists \text{hasFocus}.\text{GeneticDisorder}$$

Ontology Reuse

"*Every instance of* Project *that* hasFocus *on* CysticFibrosis, *also* hasFocus *on* GeneticDisorder"

Modelling Errors

Ontology of Research Projects

$\mathsf{CysticFibrosis\_EUProject} \equiv [\dots]$

$\mathsf{GeneticDisorder\_Project} \equiv [\dots]$

$\mathsf{EUProject} \sqsubseteq \mathsf{Project}$

$\exists \mathsf{hasFocus}.\top \sqsubseteq \mathsf{Project}$

$\mathsf{Project} \sqcap (\mathsf{CysticFibrosis} \sqcap \mathsf{GeneticDisorder}) \sqsubseteq \bot$

$\forall \mathsf{hasFocus}.\mathsf{CysticFibrosis} \sqsubseteq \exists \mathsf{hasFocus}.\mathsf{GeneticDisorder}$

Ontology Reuse

$\models \top \sqsubseteq \exists \mathsf{hasFocus}.[\neg \mathsf{CysticFibrosis} \sqcup \mathsf{GeneticDisorder}]$

Modelling Errors

Ontology of Research Projects

$\text{CysticFibrosis\_EUProject} \equiv [\ldots]$

$\text{GeneticDisorder\_Project} \equiv [\ldots]$

$\text{EUProject} \sqsubseteq \text{Project}$

$\exists \text{hasFocus}.\top \sqsubseteq \text{Project}$ ◀

$\text{Project} \sqcap (\text{CysticFibrosis} \sqcap \text{GeneticDisorder}) \sqsubseteq \bot$

$\forall \text{hasFocus}.\text{CysticFibrosis} \sqsubseteq \exists \text{hasFocus}.\text{GeneticDisorder}$

Ontology Reuse

$\models \top \sqsubseteq \exists \text{hasFocus}.[\neg \text{CysticFibrosis} \sqcup \text{GeneticDisorder}]$

$\models \top \sqsubseteq \text{Project}$

Modelling Errors

Ontology of Research Projects

$\mathsf{CysticFibrosis\_EUProject} \equiv [\ldots]$

$\mathsf{GeneticDisorder\_Project} \equiv [\ldots]$

$\mathsf{EUProject} \sqsubseteq \mathsf{Project}$

$\exists \mathsf{hasFocus}.\top \sqsubseteq \mathsf{Project}$

$\mathsf{Project} \sqcap (\mathsf{CysticFibrosis} \sqcap \mathsf{GeneticDisorder}) \sqsubseteq \bot$

$\forall \mathsf{hasFocus}.\mathsf{CysticFibrosis} \sqsubseteq \exists \mathsf{hasFocus}.\mathsf{GeneticDisorder}$

Ontology Reuse

$\models \top \sqsubseteq \exists \mathsf{hasFocus}.[\neg \mathsf{CysticFibrosis} \sqcup \mathsf{GeneticDisorder}]$

$\models \top \sqsubseteq \mathsf{Project}$

$\boxed{\models \mathsf{CysticFibrosis} \sqcap \mathsf{GeneticDisorder} \sqsubseteq \bot}$

Safe Ontology Integration: Why is it Important?

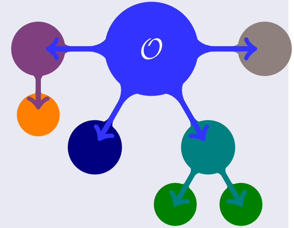
1. Independent ontology development:
   - ➡ Every ontology developer is responsible for his own domain
   - ➡ The ontology which is merely reused, is not supposed to change even implicitly

Ontology Reuse

SAFE ONTOLOGY INTEGRATION: WHY IS IT IMPORTANT?

1. Independent ontology development:
   - ➡ Every ontology developer is responsible for his own domain
   - ➡ The ontology which is merely reused, is not supposed to change even implicitly
2. Modular integration of ontologies:

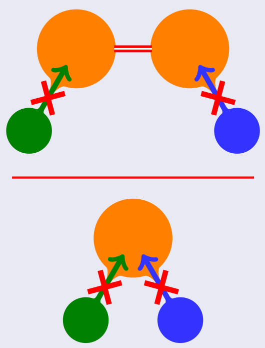
ONTOLOGY REUSE



Safe Ontology Integration: Why is it Important?

1. Independent ontology development:
   - ➡ Every ontology developer is responsible for his own domain
   - ➡ The ontology which is merely reused, is not supposed to change even implicitly
2. Modular integration of ontologies:
   - ➡ Ontologies which import safely a common ontology can be combined

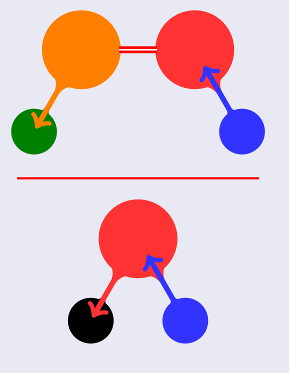
Ontology Reuse



Safe Ontology Integration: Why is it Important?

1. Independent ontology development:
   - ➡ Every ontology developer is responsible for his own domain
   - ➡ The ontology which is merely reused, is not supposed to change even implicitly
2. Modular integration of ontologies:
   - ➡ Ontologies which import safely a common ontology can be combined
   - ➡ Non-safety leads to corrupted ontologies

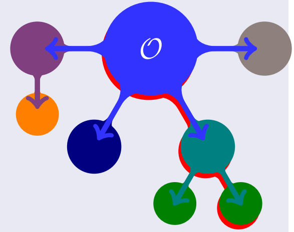
Ontology Reuse



Safe Ontology Integration: Why is it Important?

1. Independent ontology development:
   - ➡ Every ontology developer is responsible for his own domain
   - ➡ The ontology which is merely reused, is not supposed to change even implicitly
2. Modular integration of ontologies:
   - ➡ Ontologies which import safely a common ontology can be combined
   - ➡ Non-safety leads to corrupted ontologies
   - ➡ Ontology developers can continue working independently

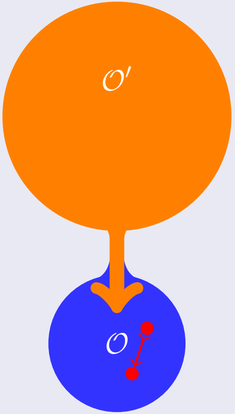
Ontology Reuse



Partial Ontology Reuse

- Available ontologies often big and contain lots of irrelevant information

Ontology Reuse



Ontology of Research Projects

CysticFibrosis_EUProject $\equiv$

$\quad\quad$ EUProject $\sqcap \exists$hasFocus.CysticFibrosis

GeneticDisorder_Project $\equiv$

$\quad\quad$ Project $\sqcap \exists$hasFocus.GeneticDisorder

EUProject $\sqsubseteq$ Project

Partial Ontology Reuse

- Available ontologies often big and contain lots of irrelevant information
- Instead of importing the full ontology one could import a part that describes just the necessary vocabulary — A module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathcal{O}$.

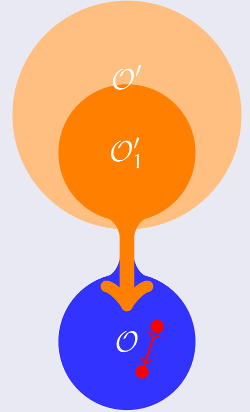
Ontology of Research Projects

$$\text{CysticFibrosis\_EUProject} \equiv \text{EUProject} \sqcap \exists \text{hasFocus}.\text{CysticFibrosis}$$

$$\text{GeneticDisorder\_Project} \equiv \text{Project} \sqcap \exists \text{hasFocus}.\text{GeneticDisorder}$$

$$\text{EUProject} \sqsubseteq \text{Project}$$

Ontology Reuse



Safe Reuse of Ontologies

Informal Definition

An ontology $\mathcal{O}$ safely reuses ontology $\mathcal{O}'$ if $\mathcal{O}$ does not change the “meaning” of the reused symbols from $\mathcal{O}'$ during the import.

Ontology Reuse

Safe Reuse of Ontologies

Definition (1)

$\mathcal{O}' \cup \mathcal{O}$ is a conservative extension of $\mathcal{O}'$ w.r.t. ontology language $\mathcal{L}$ if for every axiom α over $\mathcal{O}'$ expressed in $\mathcal{L}$, we have:

$$\boxed{\mathcal{O}' \cup \mathcal{O} \models \alpha \quad \text{iff} \quad \mathcal{O}' \models \alpha}$$

Informal Definition

An ontology $\mathcal{O}$ safely reuses ontology $\mathcal{O}'$ if $\mathcal{O}$ does not change the “meaning” of the reused symbols from $\mathcal{O}'$ during the import.

Ontology Reuse

Safe Reuse of Ontologies

Definition (1)

$\mathcal{O}' \cup \mathcal{O}$ is a conservative extension of $\mathcal{O}'$ w.r.t. ontology language $\mathcal{L}$ if for every axiom α over $\mathcal{O}'$ expressed in $\mathcal{L}$, we have:

$$\boxed{\mathcal{O}' \cup \mathcal{O} \models \alpha \quad \text{iff} \quad \mathcal{O}' \models \alpha}$$

Example (1)

$$\mathcal{O}' = \begin{cases} \mathsf{A} \equiv \cdots \\ \mathsf{B} \equiv \cdots \end{cases} \qquad \not\models \mathsf{B} \sqsubseteq \mathsf{A}$$

$$\mathcal{O} = \begin{cases} \mathsf{C}_1 \equiv \mathsf{A} \sqcap \mathsf{C}_2 \\ \mathsf{B} \sqsubseteq \mathsf{C}_1 \end{cases} \qquad \models \mathsf{B} \sqsubseteq \mathsf{A}$$

$\mathcal{O}' \cup \mathcal{O}$ is not a conservative extension of $\mathcal{O}'$ w.r.t. $\mathcal{L} = \mathcal{ALC}$.

Ontology Reuse

Safe Reuse of Ontologies

Definition (1)

$\mathcal{O}' \cup \mathcal{O}$ is a conservative extension of $\mathcal{O}'$ w.r.t. ontology language $\mathcal{L}$ if for every axiom α over $\mathcal{O}'$ expressed in $\mathcal{L}$, we have:

$$\boxed{\mathcal{O}' \cup \mathcal{O} \models \alpha \quad \text{iff} \quad \mathcal{O}' \models \alpha}$$

Example (2)

$$\mathcal{O}' = \{\ \mathsf{A} \equiv \cdots \qquad \not\models \top \sqsubseteq \mathsf{A},\ \mathsf{A} \sqsubseteq \bot$$

$$\mathcal{O} = \left\{ \begin{array}{l} a : (\mathsf{A} \sqcap \mathsf{B}) \\ b : (\mathsf{A} \sqcap \neg \mathsf{B}) \end{array} \right. \qquad \not\models \top \sqsubseteq \mathsf{A},\ \mathsf{A} \sqsubseteq \bot$$

$\mathcal{O}' \cup \mathcal{O}$ is a conservative extension of $\mathcal{O}'$ w.r.t. $\mathcal{L} = \mathcal{ALC}$

Ontology Reuse

Safe Reuse of Ontologies

Definition (1)

$\mathcal{O}' \cup \mathcal{O}$ is a conservative extension of $\mathcal{O}'$ w.r.t. ontology language $\mathcal{L}$ if for every axiom α over $\mathcal{O}'$ expressed in $\mathcal{L}$, we have:

$$\boxed{\mathcal{O}' \cup \mathcal{O} \models \alpha \quad \text{iff} \quad \mathcal{O}' \models \alpha}$$

Example (2)

$$\mathcal{O}' = \{\ \mathsf{A} \equiv \cdots \qquad \not\models \top \sqsubseteq \mathsf{A},\ \mathsf{A} \sqsubseteq \bot$$

$$\mathcal{O} = \left\{ \begin{array}{ll} a : (\mathsf{A} \sqcap \mathsf{B}) & \not\models \top \sqsubseteq \mathsf{A},\ \mathsf{A} \sqsubseteq \bot \\ b : (\mathsf{A} \sqcap \neg\mathsf{B}) & \models |\mathsf{A}| \geq 2 \end{array} \right.$$

$\mathcal{O}' \cup \mathcal{O}$ is a conservative extension of $\mathcal{O}'$ w.r.t. $\mathcal{L} = \mathcal{ALC}$

The "meaning" of A has been changed, but $\mathcal{L} = \mathcal{ALC}$ cannot "detect" it using axioms.

Ontology Reuse

Safe Reuse of Ontologies

Definition (2)

$\mathcal{O}' \cup \mathcal{O}$ is a model conservative extension of $\mathcal{O}'$ w.r.t. ontology language $\mathcal{L}$ if every model of $\mathcal{O}'$ can be expanded to a model of $\mathcal{O}' \cup \mathcal{O}$:

$$\boxed{\forall \mathcal{I} \models \mathcal{O}' \ \exists \mathcal{J} \models \mathcal{O} : \ \mathcal{I}|_{\mathcal{O}'} = \mathcal{J}|_{\mathcal{O}'}}$$

Example (2)

$$\mathcal{O}' = \{ \ \mathsf{A} \equiv \cdots \qquad \not\models \top \sqsubseteq \mathsf{A},\ \mathsf{A} \sqsubseteq \bot$$

$$\mathcal{O} = \begin{cases} a : (\mathsf{A} \sqcap \mathsf{B}) & \not\models \top \sqsubseteq \mathsf{A},\ \mathsf{A} \sqsubseteq \bot \\ b : (\mathsf{A} \sqcap \neg\mathsf{B}) & \models |\mathsf{A}| \geq 2 \end{cases}$$

$\mathcal{O}' \cup \mathcal{O}$ is a conservative extension of $\mathcal{O}'$ w.r.t. $\mathcal{L} = \mathcal{ALC}$, but not model conservative

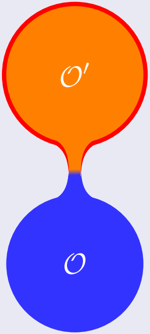
The "meaning" of A has been changed, but $\mathcal{L} = \mathcal{ALC}$ cannot "detect" it using axioms.

Ontology Reuse

SAFETY FOR EVOLVING ONTOLOGIES

- Ontologies are developed ⇒ evolve

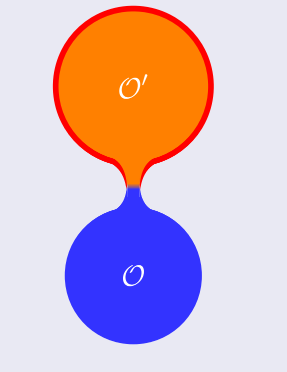
ONTOLOGY REUSE



Safety for Evolving Ontologies

- Ontologies are developed ⇒ evolve

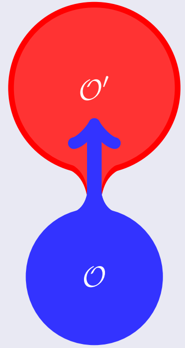
Ontology Reuse



Safety for Evolving Ontologies

- Ontologies are developed $\Rightarrow$ evolve
- Even if $\mathcal{O}$ is importing safely one version of $\mathcal{O}'$, this might no longer hold for another version

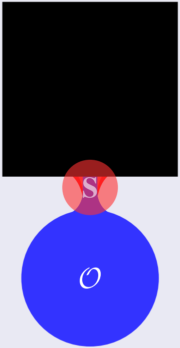
Ontology Reuse



Safety for Evolving Ontologies

- Ontologies are developed $\Rightarrow$ evolve
- Even if $\mathcal{O}$ is importing safely one version of $\mathcal{O}'$, this might no longer hold for another version
- Instead of focusing on the reused ontology one could focus just on the reused symbols and treat the ontology as a “black box”.

Ontology Reuse

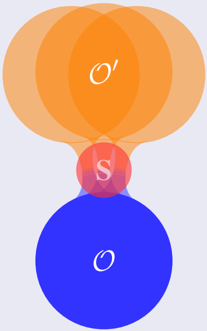


Safety of an Ontology for a Signature

Definition (Safety for a Signature)

$\mathcal{O}$ is safe for a signature $\mathbf{S}$ w.r.t. an ontology language $\mathcal{L}$ if for every $\mathcal{O}'$ formulated over $\mathcal{L}$ with $\mathsf{Sg}(\mathcal{O}') \cap \mathsf{Sg}(\mathcal{O}) \subseteq \mathbf{S}$, we have that $\mathcal{O} \cup \mathcal{O}'$ is a conservative extension of $\mathcal{O}'$.

Ontology Reuse



Safety of an Ontology for a Signature

Definition (Safety for a Signature)

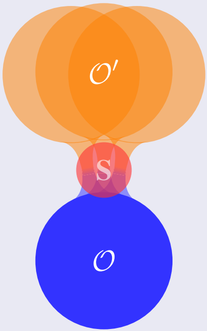
$\mathcal{O}$ is safe for a signature $\mathbf{S}$ w.r.t. an ontology language $\mathcal{L}$ if for every $\mathcal{O}'$ formulated over $\mathcal{L}$ with $\mathsf{Sg}(\mathcal{O}') \cap \mathsf{Sg}(\mathcal{O}) \subseteq \mathbf{S}$, we have that $\mathcal{O} \cup \mathcal{O}'$ is a conservative extension of $\mathcal{O}'$.

Theorem (Sufficient Condition)

An ontology $\mathcal{O}$ is safe for a signature $\mathbf{S}$ if for every interpretation $\mathcal{I}$ there exists a model $\mathcal{J}$ of $\mathcal{O}$ that coincides with $\mathcal{I}$ on $\mathbf{S}$:

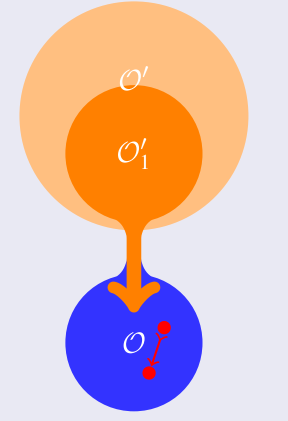
$$\boxed{\forall\, \mathcal{I}\ \exists\, \mathcal{J} \models \mathcal{O} : \ \mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}}$$

Ontology Reuse



Module for Ontology

Ontology Reuse



Informal Definition

An ontology $\mathcal{O}'_1$ is a module in ontology $\mathcal{O}'$ for the importing ontology $\mathcal{O}$, if importing $\mathcal{O}'_1$ into $\mathcal{O}$ instead of $\mathcal{O}'$ has the same impact on the "meaning" symbols in $\mathcal{O}$.

Module for Ontology

Definition (Module for Ontology)

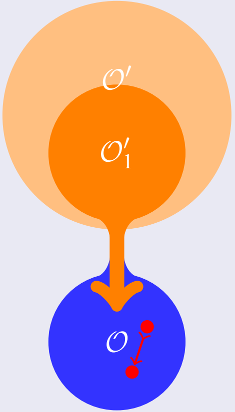
$\mathcal{O}'_1$ is a module in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and ontology language $\mathcal{L}$ if for every axiom α over $\mathcal{O}$ expressed in $\mathcal{L}$, we have:

$$\boxed{\mathcal{O}'_1 \cup \mathcal{O} \models \alpha \quad \text{iff} \quad \mathcal{O}' \cup \mathcal{O} \models \alpha}$$

Informal Definition

An ontology $\mathcal{O}'_1$ is a module in ontology $\mathcal{O}'$ for the importing ontology $\mathcal{O}$, if importing $\mathcal{O}'_1$ into $\mathcal{O}$ instead of $\mathcal{O}'$ has the same impact on the "meaning" symbols in $\mathcal{O}$.

Ontology Reuse



Module for Ontology

Definition (Module for Ontology)

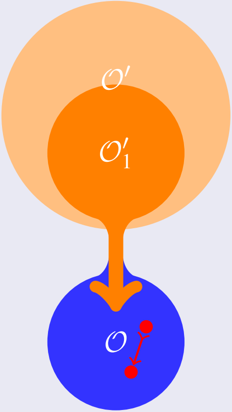
$\mathcal{O}'_1$ is a module in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and ontology language $\mathcal{L}$ if for every axiom α over $\mathcal{O}$ expressed in $\mathcal{L}$, we have:

$$\boxed{\mathcal{O}'_1 \cup \mathcal{O} \models \alpha \quad \text{iff} \quad \mathcal{O}' \cup \mathcal{O} \models \alpha}$$

Observation

The empty ontology is a module in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$ if $\mathcal{O}' \cup \mathcal{O}$ is a conservative extension of $\mathcal{O}$ w.r.t. $\mathcal{L}$.

Ontology Reuse



Module for Ontology

Definition (Module for Ontology)

$\mathcal{O}'_1$ is a module in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and ontology language $\mathcal{L}$ if for every axiom α over $\mathcal{O}$ expressed in $\mathcal{L}$, we have:

$$\boxed{\mathcal{O}'_1 \cup \mathcal{O} \models \alpha \quad \text{iff} \quad \mathcal{O}' \cup \mathcal{O} \models \alpha}$$

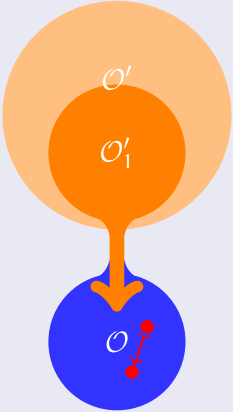
Example

$$\mathcal{O}' = \begin{cases} \mathsf{A} \equiv \mathsf{B} \sqcap \exists r.\mathsf{C} \\ \mathsf{A} \sqcap \mathsf{D} \sqsubseteq \bot \end{cases}$$

$$\mathcal{O} = \begin{cases} \mathsf{C}_1 \equiv \cdots \\ \mathsf{C}_2 \sqsubseteq \cdots \end{cases}$$

(does not contain D)

Ontology Reuse



Module for Ontology

Definition (Module for Ontology)

$\mathcal{O}'_1$ is a module in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and ontology language $\mathcal{L}$ if for every axiom α over $\mathcal{O}$ expressed in $\mathcal{L}$, we have:

$$\boxed{\mathcal{O}'_1 \cup \mathcal{O} \models \alpha \quad \text{iff} \quad \mathcal{O}' \cup \mathcal{O} \models \alpha}$$

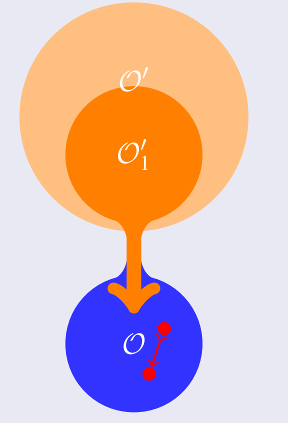
Example

$$\mathcal{O}'_1 = \begin{cases} \mathsf{A} \equiv \mathsf{B} \sqcap \exists r.\mathsf{C} \\ \cancel{\mathsf{A} \sqcap \mathsf{D} \sqsubseteq \bot} \end{cases}$$

$$\mathcal{O} = \begin{cases} \mathsf{C}_1 \equiv \cdots \\ \mathsf{C}_2 \sqsubseteq \cdots \end{cases} \qquad \text{(does not contain D)}$$

$\mathcal{O}'_1$ is a module in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L} = \mathcal{ALC}$.

Ontology Reuse

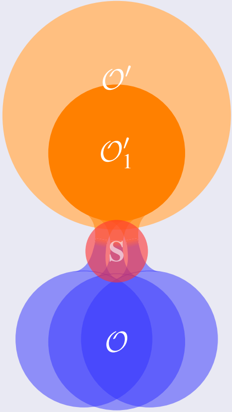


Module in Ontology for a Signature

Definition (Module for Signature)

$\mathcal{O}'_1$ is a module in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and ontology language $\mathcal{L}$ if for every ontology $\mathcal{O}$ formulated over $\mathcal{L}$ with $\mathsf{Sg}(\mathcal{O}) \cap \mathsf{Sg}(\mathcal{O}') \subseteq \mathbf{S}$, we have that $\mathcal{O}'_1$ is a module in $\mathcal{O}'$ w.r.t. $\mathcal{O}$.

Ontology Reuse



Module in Ontology for a Signature

Definition (Module for Signature)

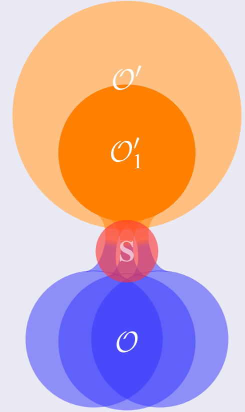
$\mathcal{O}'_1$ is a module in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and ontology language $\mathcal{L}$ if for every ontology $\mathcal{O}$ formulated over $\mathcal{L}$ with $\mathsf{Sg}(\mathcal{O}) \cap \mathsf{Sg}(\mathcal{O}') \subseteq \mathbf{S}$, we have that $\mathcal{O}'_1$ is a module in $\mathcal{O}'$ w.r.t. $\mathcal{O}$.

Theorem (Sufficient Condition)

An ontology $\mathcal{O}'_1$ is module in $\mathcal{O}'$ w.r.t. a signature $\mathbf{S}$ if for every model $\mathcal{I}$ of $\mathcal{O}'_1$ there exists a model $\mathcal{J}$ of $\mathcal{O}'$ that coincides with $\mathcal{I}$ on $\mathbf{S}$:

$$\boxed{\forall \mathcal{I} \,\exists \mathcal{J} \models \mathcal{O} : \mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}}$$

Ontology Reuse



Outline

1 Background

2 Safety and Modules

- Motivation
- Formalization

3 Algorithms

Reasoning Problems

Not	Input	Task	
T1	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$	
T2	$\mathcal{O}$, **S**, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for **S** w.r.t. $\mathcal{L}$	

Reasoning Problems

Not	Input	Task	
T1	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$	
T2	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$	
T3	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	
T4	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	

Reasoning Problems

Not	Input	Task	
T1	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$	
T2	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$	
T3	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	
T4	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	
T3m	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	
T4m	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	

Reasoning Problems

Not	Input	Task	
T1	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$	
T2	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$	
T3	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	
T4	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	
T3m*	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	
T4m*	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	

*variants=[all / some / union of] minimal modules

Reasoning Problems

Not	Input	Task	
T1	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$	
T2	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$	
T3	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	
T4	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	
T3m*	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	
T4m*	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	

*variants=[all / some / union of] minimal modules

Theorem

Checking if $\mathcal{O}' \cup \mathcal{O}$ is a conservative extension of $\mathcal{O}'$ w.r.t. $\mathcal{L}$ is 2-EXPTIME-complete for $\mathcal{L} = \mathcal{ALCQI}$ [Ghilardi, Lutz & Wolter, 2006] and is uncecidable for $\mathcal{L} = \mathcal{ALCQIO}$ [Lutz, Walther & Wolter, 2007].

Reasoning Problems

Not	Input	Task	
T1	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$	☹
T2	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$	
T3	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	
T4	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	
T3m*	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	☹
T4m*	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	

*variants=[all / some / union of] minimal modules

Theorem

Checking if $\mathcal{O}' \cup \mathcal{O}$ is a conservative extension of $\mathcal{O}'$ w.r.t. $\mathcal{L}$ is 2-EXPTIME-complete for $\mathcal{L} = \mathcal{ALCQI}$ [Ghilardi, Lutz & Wolter, 2006] and is uncecidable for $\mathcal{L} = \mathcal{ALCQIO}$ [Lutz, Walther & Wolter, 2007].

Corollary: Then so are the tasks T1 and T3m*

Reasoning Problems

Not	Input	Task	
T1	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$	☹
T2	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$	
T3	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	
T4	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	
T3m*	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	☹
T4m*	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	

*variants=[all / some / union of] minimal modules

Theorem

Given an ontology $\mathcal{O}$ consisting only of a single $\mathcal{ALC}$-axiom, and a signature $\mathbf{S}$, it is undecidable whether $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L} = \mathcal{ALCO}$.

Reasoning Problems

Not	Input	Task	
T1	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$	☹
T2	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$	☹
T3	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	
T4	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	
T3m*	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	☹
T4m*	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	☹

*variants=[all / some / union of] minimal modules

Theorem

Given an ontology $\mathcal{O}$ consisting only of a single $\mathcal{ALC}$-axiom, and a signature $\mathbf{S}$, it is undecidable whether $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L} = \mathcal{ALCO}$.

Corollary: Then so are the tasks T2 and T4m*

Reasoning Problems

Not	Input	Task	
T1	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$	☹
T2	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$	☹
T3	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	?
T4	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	?
T3m*	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	☹
T4m*	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	☹

*variants=[all / some / union of] minimal modules

How to obtain a practical solution for T3 and T4?

A Sufficient Condition for Safety

Theorem (Sufficient Condition, Reminder)

An ontology $\mathcal{O}$ is safe for a signature $\mathbf{S}$ if for every interpretation $\mathcal{I}$ there exists a model $\mathcal{J}$ of $\mathcal{O}$ that coincides with $\mathcal{I}$ on $\mathbf{S}$:

$$\boxed{\forall\, \mathcal{I}\ \exists\, \mathcal{J} \models \mathcal{O} :\ \mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}}$$

The main idea:

- To prove that $\mathcal{O}$ is safe for $\mathbf{S}$ it is sufficient to extend any interpretation $\mathcal{I}$ of symbols from $\mathbf{S}$ to a model of $\mathcal{O}$

A Sufficient Condition for Safety

Theorem (Sufficient Condition, Reminder)

An ontology $\mathcal{O}$ is safe for a signature $\mathbf{S}$ if for every interpretation $\mathcal{I}$ there exists a model $\mathcal{J}$ of $\mathcal{O}$ that coincides with $\mathcal{I}$ on $\mathbf{S}$:

$$\boxed{\forall \mathcal{I}\ \exists \mathcal{J} \models \mathcal{O} : \mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}}$$

The main idea:

- To prove that $\mathcal{O}$ is safe for $\mathbf{S}$ it is sufficient to extend any interpretation $\mathcal{I}$ of symbols from $\mathbf{S}$ to a model of $\mathcal{O}$
- Let us try to extend $\mathcal{I}$ by interpreting every new symbol as the empty set

A Sufficient Condition for Safety

Theorem (Sufficient Condition, Reminder)

An ontology $\mathcal{O}$ is safe for a signature $\mathbf{S}$ if for every interpretation $\mathcal{I}$ there exists a model $\mathcal{J}$ of $\mathcal{O}$ that coincides with $\mathcal{I}$ on $\mathbf{S}$:

$$\boxed{\forall \mathcal{I}\ \exists \mathcal{J} \models \mathcal{O} : \mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}}$$

The main idea:

- To prove that $\mathcal{O}$ is safe for $\mathbf{S}$ it is sufficient to extend any interpretation $\mathcal{I}$ of symbols from $\mathbf{S}$ to a model of $\mathcal{O}$
- Let us try to extend $\mathcal{I}$ by interpreting every new symbol as the empty set

Example

$$\mathcal{O} = \begin{cases} \mathsf{A} \equiv \mathsf{B} \sqcap \exists r.\mathsf{C} \\ \mathsf{A} \sqcap \mathsf{B} \sqsubseteq \bot \\ \exists r.\top \sqsubseteq \mathsf{C} \end{cases} \xRightarrow[\mathsf{A} \leftarrow \emptyset]{r \leftarrow \emptyset} \begin{matrix} \bot \equiv \mathsf{B} \sqcap \bot \checkmark \\ \bot \sqcap \mathsf{B} \sqsubseteq \bot \checkmark \\ \bot \sqsubseteq \mathsf{C} \quad \checkmark \end{matrix}$$

Locality

Definition (Locality for Ontology Languages)

An ontology $\mathcal{O}$ is local w.r.t. $\mathbf{S}$ if $\mathcal{J} \models \mathcal{O}$ for every $\mathcal{J}$ which interprets all concept and role names not in $\mathbf{S}$ as the empty set.

Locality

Definition (Locality for Ontology Languages)

An ontology $\mathcal{O}$ is local w.r.t. $\mathbf{S}$ if $\mathcal{J} \models \mathcal{O}$ for every $\mathcal{J}$ which interprets all concept and role names not in $\mathbf{S}$ as the empty set.

+ If every $\mathcal{O}$ is local w.r.t. $\mathbf{S}$ then $\mathcal{O}$ is safe for $\mathbf{S}$:

Locality

Definition (Locality for Ontology Languages)

An ontology $\mathcal{O}$ is local w.r.t. $\mathbf{S}$ if $\mathcal{J} \models \mathcal{O}$ for every $\mathcal{J}$ which interprets all concept and role names not in $\mathbf{S}$ as the empty set.

+ If every $\mathcal{O}$ is local w.r.t. $\mathbf{S}$ then $\mathcal{O}$ is safe for $\mathbf{S}$:
+ Checking locality can be done using any standard DL-reasoner.

Locality

Definition (Locality for Ontology Languages)

An ontology $\mathcal{O}$ is local w.r.t. $\mathbf{S}$ if $\mathcal{J} \models \mathcal{O}$ for every $\mathcal{J}$ which interprets all concept and role names not in $\mathbf{S}$ as the empty set.

+ If every $\mathcal{O}$ is local w.r.t. $\mathbf{S}$ then $\mathcal{O}$ is safe for $\mathbf{S}$:
+ Checking locality can be done using any standard DL-reasoner.
+ There is a sufficient syntactical condition for locality which can be verified in polynomial time.

Locality

Definition (Locality for Ontology Languages)

An ontology $\mathcal{O}$ is local w.r.t. $\mathbf{S}$ if $\mathcal{J} \models \mathcal{O}$ for every $\mathcal{J}$ which interprets all concept and role names not in $\mathbf{S}$ as the empty set.

- If every $\mathcal{O}$ is local w.r.t. $\mathbf{S}$ then $\mathcal{O}$ is safe for $\mathbf{S}$:
- Checking locality can be done using any standard DL-reasoner.
- There is a sufficient syntactical condition for locality which can be verified in polynomial time.

Syntactic Locality

$$C^{\emptyset} ::= \mathsf{A}^{\emptyset} \mid C^{\emptyset} \sqcap C \mid C^{\emptyset} \sqcup C^{\emptyset} \mid \neg C^{\Delta} \mid \exists r^{\emptyset}.C \mid \exists r.C^{\emptyset}$$

$$C^{\Delta} ::= C^{\Delta} \sqcup C \mid C^{\Delta} \sqcap C^{\Delta} \mid \neg C^{\emptyset} \mid \forall r^{\emptyset}.C \mid \forall r.C^{\Delta}$$

$$Ax\_synt\_local ::= C^{\emptyset} \sqsubseteq C \mid C \sqsubseteq C^{\Delta}$$

A Module-Extraction Algorithm Based on Locality

Theorem (Sufficient Condition, Reminder)

An ontology $\mathcal{O}'_1$ is module in $\mathcal{O}'$ w.r.t. a signature $\mathbf{S}$ if for every model $\mathcal{I}$ of $\mathcal{O}'_1$ there exists a model $\mathcal{J}$ of $\mathcal{O}'$ that coincides with $\mathcal{I}$ on $\mathbf{S}$: $\boxed{\forall\, \mathcal{I}\; \exists\, \mathcal{J} \models \mathcal{O} : \mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}}$

Proposition (Modules and Safety)

If $\mathcal{O}' \setminus \mathcal{O}'_1$ is safe for $\mathbf{S} \cup \mathsf{Sg}(\mathcal{O}'_1)$ then $\mathcal{O}'_1$ is a module in $\mathcal{O}'$ for $\mathbf{S}$.

A Module-Extraction Algorithm Based on Locality

Theorem (Sufficient Condition, Reminder)

An ontology $\mathcal{O}'_1$ is module in $\mathcal{O}'$ w.r.t. a signature $\mathbf{S}$ if for every model $\mathcal{I}$ of $\mathcal{O}'_1$ there exists a model $\mathcal{J}$ of $\mathcal{O}'$ that coincides with $\mathcal{I}$ on $\mathbf{S}$: $\boxed{\forall \mathcal{I}\, \exists \mathcal{J} \models \mathcal{O} : \mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}}$

Proposition (Modules and Safety)

If $\mathcal{O}' \setminus \mathcal{O}'_1$ is safe for $\mathbf{S} \cup \mathsf{Sg}(\mathcal{O}'_1)$ then $\mathcal{O}'_1$ is a module in $\mathcal{O}'$ for $\mathbf{S}$.

Algorithm for extracting a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathbf{S}$:

1. Initialize $\mathcal{O}'_1$ to be an empty ontology: $\mathcal{O}'_1 := \emptyset$
2. Find an axiom $\alpha \in \mathcal{O}' \setminus \mathcal{O}'_1$ that is is local w.r.t. $\mathbf{S} \cup \mathsf{Sg}(\mathcal{O}'_1)$
3. Move α into $\mathcal{O}'_1$ and repeat until no other α left.

EMPERICAL EVALUATION

Ontology	♯ Atomic	A1: Prompt-Factor		A2: Mod. in [CG'06]		A3: Loc.-based mod.	
	Concepts	Max.(%)	Avg.(%)	Max.(%)	Avg.(%)	Max.(%)	Avg.(%)
NCI	27772	87.6	75.84	55	30.8	0.8	0.08
SNOMED	255318	100	100	100	100	0.5	0.05
GO	22357	1	0.1	1	0.1	0.4	0.05
SUMO	869	100	100	100	100	2	0.09
GALEN-Small	2749	100	100	100	100	10	1.7
GALEN-Full	24089	100	100	100	100	29.8	3.5
SWEET	1816	96.4	88.7	83.3	51.5	1.9	0.1
DOLCE-Lite	499	100	100	100	100	37.3	24.6

[SK'04] H. Stuckenschmidt & M. Klein Structure-based partitioning of large class hierarchies. ISWC 2004

[CG'06] B. Cuenca Grau, B. Parsia, E. Sirin, & A. Kalyanpur. Modularity and Web Ontologies. KR 2006

CONTRIBUTIONS

- Formalization for the notions for safety and modules using logical notions of conservative extension
- Theoretical studies for the relevant tasks (decidability, complexity)
- Practical algorithms for extracting modules and safety checking with guarantied correctness of the results

1. B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U.Sattler. A logical framework for modularity of ontologies. In Proc. of IJCAI 2007
2. B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Just the right amout: Extracting modules from ontologies. In Proc. of WWW 2007
3. B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular Reuse of Ontologies: Theory and Practice. JAIR 2008, to appear

Other Locality Conditions

Other locality conditions can be defined by choosing different ways to interpret the symbols that are not in **S**:

Examples and Comparison of Different Localities

$\mathsf{r} \leftarrow$	$\emptyset$	$\Delta \times \Delta$	id	$\emptyset$	$\Delta \times \Delta$	id
$\mathsf{A} \leftarrow$	$\emptyset$	$\emptyset$	$\emptyset$	Δ	Δ	Δ
$\mathsf{A} \equiv \mathsf{B} \sqcap \exists r.\mathsf{C}$	✓	✓	✓	✗	✗	✗
$\mathsf{A} \sqcap \mathsf{C} \sqsubseteq \bot$	✓	✓	✓	✗	✗	✗
$\exists r.\top \sqsubseteq \mathsf{A}$	✓	✗	✗	✓	✓	✓
$Functional(r)$	✓	✗	✓	✓	✗	✓
$a : \mathsf{A}$	✗	✗	✗	✓	✓	✓
$\mathsf{r}(a, \mathsf{b})$	✗	✓	✗	✗	✓	✗
$\forall r.\mathsf{C} \sqsubseteq \exists \mathsf{r}.\mathsf{D}$	✗	✗	✗	✗	✗	✗