MODULARITY FOR ONTOLOGIES: FROM THEORY TO PRACTICE

Yevgeny Kazakov

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

The University of Oxford

November 20, 2007

Outline

DL-Based Ontology Languages

- Ontologies are vocabularies of terms for specific subjects
  - drugs (NCI)
  - genes (GO)
  - human anatomy (Galen, SNoMed)
  - biological processes (BioPAX)
  - geography (Ordnance Survey)
  - wines (Wine)
  - pizzas (Pizzas)
  - tourism (Travel)
  - . . .

DL-Based Ontology Languages

- Two types of axioms

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

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

DL-Based Ontology Languages

- Two types of axioms
  - Terminalogical axiom [Schema]

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

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

DL-Based Ontology Languages

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

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

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

DL-Based Ontology Languages

- The syntax

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

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

DL-Based Ontology Languages

- The syntax
  - Atomic concepts [Classes]

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

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

DL-Based Ontology Languages

- The syntax
  - Atomic concepts [Classes]
  - Atomic roles [Properties]

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

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

DL-Based Ontology Languages

- The syntax
  - Atomic concepts [Classes]
  - Atomic roles [Properties]
  - Individuals

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

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

DL-Based Ontology Languages

- The syntax
  - Atomic concepts [Classes]
  - Atomic roles [Properties]
  - Individuals
  - Constructors

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

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

DL-Based Ontology Languages

- The semantics

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

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

DL-Based Ontology Languages

- The semantics
  - Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$

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

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

DL-BASED ONTOLOGY LANGUAGES

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

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

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

DL-Based Ontology Languages

- The semantics
  - 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

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

O_Id7894 : Heart

Δ

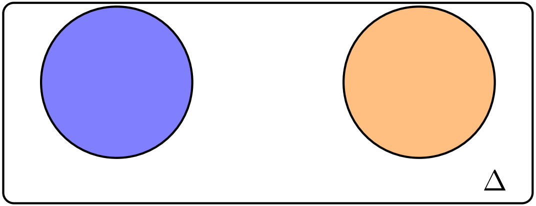
DL-Based Ontology Languages

- The semantics
  - 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

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

O_Id7894 : Heart



DL-Based Ontology Languages

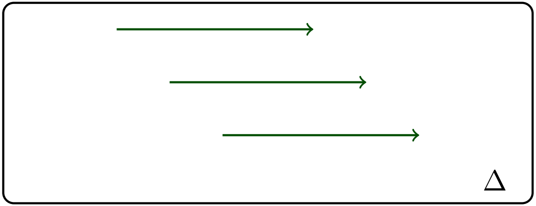
- The semantics
  - Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$
    - $\Delta^{\mathcal{I}}$ is an interpretation domain (non-empty set)
    - $\cdot^{\mathcal{I}}$ is an interpretation function

      Atomic concepts $\Rightarrow$ sets

      Atomic roles $\Rightarrow$ binary relations

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

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



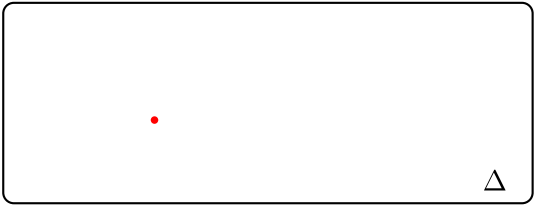
DL-Based Ontology Languages

- The semantics
  - 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
      Atomic roles $\Rightarrow$ binary relations
      Individuals $\Rightarrow$ elements

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

O_Id7894: Heart

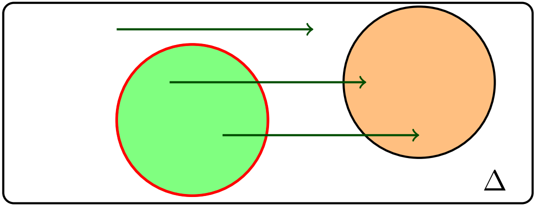


DL-Based Ontology Languages

- The semantics
  - 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

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

O_Id7894 : Heart

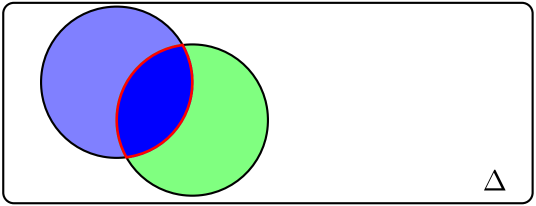


DL-Based Ontology Languages

- The semantics
  - 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

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

O_Id7894 : Heart

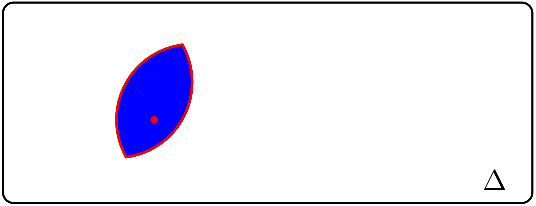


DL-Based Ontology Languages

- The semantics
  - 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

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

O_Id7894 : Heart



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	$\forall r.C$	$\forall y.[r(x,y) \rightarrow C(y)]$	$\mathcal{C}$
existential restr.	$\exists r.C$	$\exists y.[r(x,y) \wedge C(y)]$	
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. W3C standard OWL DL $\rightsquigarrow \mathcal{SHOIN}$

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

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
  - $\text{Heart} \sqsubseteq \text{CardiovascularOrgan}$

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
  - $\text{Heart} \sqsubseteq \text{CardiovascularOrgan}$

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
  - $\text{Heart} \sqsubseteq \text{CardiovascularOrgan}$

Reasoning in Ontologies

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

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

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

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

- Ontology reasoning = extracting implicit information
  - $\text{Heart} \sqsubseteq \text{CardiovascularOrgan}$

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
  - $\text{Heart} \sqsubseteq \text{CardiovascularOrgan}$

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
  - $\text{Heart} \sqsubseteq \text{CardiovascularOrgan}$
  - $\text{O\_Id7894} : \exists \text{isPartOf}.(\text{MuscularSystem} \sqcap \text{CirculatorySystem})$

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
  - $\mathsf{Heart} \sqsubseteq \mathsf{CardiovascularOrgan}$
  - $O\_Id7894 : \exists\mathsf{isPartOf}.(\mathsf{MuscularSystem} \sqcap \mathsf{CirculatorySystem})$
- Standard reasoning tasks:

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
  - $\text{Heart} \sqsubseteq \text{CardiovascularOrgan}$
  - $\text{O\_Id7894} : \exists \text{isPartOf}.(\text{MuscularSystem} \sqcap \text{CirculatorySystem})$
- Standard reasoning tasks:
  - Classification:
    – compute all subsumptions $A \sqsubseteq B$ between named classes

REASONING IN ONTOLOGIES

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

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

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

$O\_Id7894 : \text{Heart}$

- Ontology reasoning = extracting implicit information
  - $\text{Heart} \sqsubseteq \text{CardiovascularOrgan}$
  - $O\_Id7894 : \exists \text{isPartOf}.(\text{MuscularSystem} \sqcap \text{CirculatorySystem})$
- Standard reasoning tasks:
  - Classification:
    – compute all subsumptions $A \sqsubseteq B$ between named classes
  - Instance retrieval:
    – compute all (implicit) instances i of a class C.

Reasoning in Ontologies

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

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

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

$O\_Id7894 : \text{Heart}$

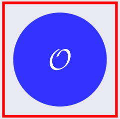
- Ontology reasoning = extracting implicit information
  - $\text{Heart} \sqsubseteq \text{CardiovascularOrgan}$
  - $O\_Id7894 : \exists \text{isPartOf}.(\text{MuscularSystem} \sqcap \text{CirculatorySystem})$
- Standard reasoning tasks:
  - Classification:
    – compute all subsumptions $A \sqsubseteq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

Outline

Reasoning Support for Ontology Development

- Debugging:

Ontology



Reasoning Support for Ontology Development

- Debugging:
  - ✔ Checking global consistency

Ontology

Reasoning Support for Ontology Development

- Debugging:
  - ✔ Checking global consistency
  - ✔ Detecting unsatisfiable classes

Ontology

Reasoning Support for Ontology Development

- Debugging:
  - ✔ Checking global consistency
  - ✔ Detecting unsatisfiable classes
  - ✔ Detecting unintended subsumptions

Ontology

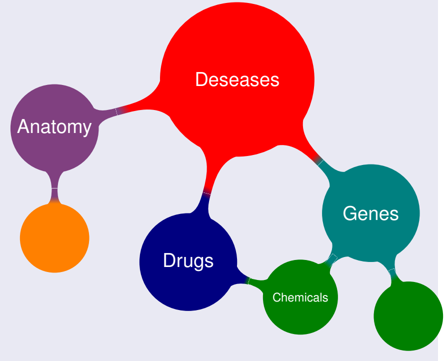
Reasoning Support for Ontology Development

- Debugging:
  - ✔ Checking global consistency
  - ✔ Detecting unsatisfiable classes
  - ✔ Detecting unintended subsumptions
- Not sufficient for large-scale ontology development
  Ontologies $\sim$ Wikipedia

ONTOLOGY ENGINEERING AT THE LARGE SCALE

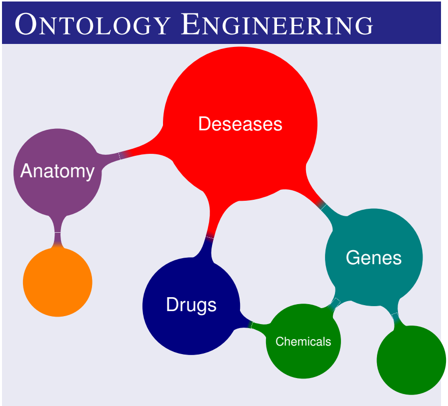
- Sharing of resources
- Collaborative development
- Continuous process
- The notion of modularity becomes apparent

ONTOLOGY ENGINEERING



Ontology Engineering at the Large Scale

- Sharing of resources
- Collaborative development
- Continuous process
- The notion of modularity becomes apparent
- Challenges:
  1. Safe integration
  2. Partial reuse



A Motivating Example

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

A Motivating Example

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 $\equiv$

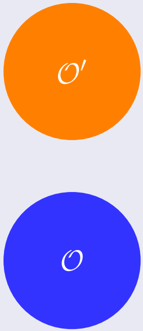
EUProject $\sqcap \exists$hasFocus.CysticFibrosis

GeneticDisorder_Project $\equiv$

Project $\sqcap \exists$hasFocus.GeneticDisorder

EUProject $\sqsubseteq$ Project

Ontology Reuse



A Motivating Example

Ontology of Medical Terms

$\mathsf{GeneticDisorder} \equiv \ldots$

$\mathsf{CysticFibrosis} \equiv \ldots$

Ontology of Research Projects

$\mathsf{CysticFibrosis\_EUProject} \equiv$

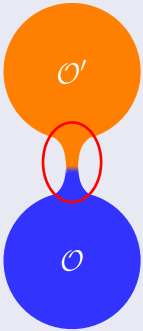
$\mathsf{EUProject} \sqcap \exists \mathsf{hasFocus}.\mathsf{CysticFibrosis}$

$\mathsf{GeneticDisorder\_Project} \equiv$

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

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

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

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

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

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

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

$\models$ CysticFibrosis_EUProject $\sqsubseteq$ GeneticDisorder_Project

Ontology Reuse

Partial Ontology Reuse

- Available ontologies are often big and contain lots of irrelevant information

Ontology of Research Projects

CysticFibrosis_EUProject $\equiv$

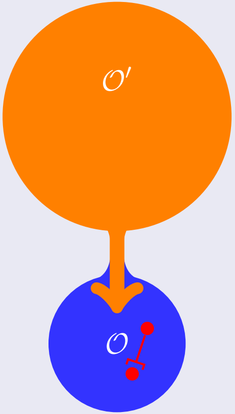
EUProject $\sqcap$ $\exists$hasFocus.CysticFibrosis

GeneticDisorder_Project $\equiv$

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

EUProject $\sqsubseteq$ Project

Ontology Reuse



Partial Ontology Reuse

- Available ontologies are often big and contain lots of irrelevant information
- A module is a part that "*expresses completely*" the reused vocabulary.

Ontology of Research Projects

CysticFibrosis_EUProject $\equiv$

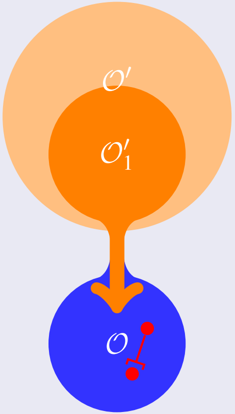
EUProject $\sqcap$ $\exists$hasFocus.CysticFibrosis

GeneticDisorder_Project $\equiv$

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

EUProject $\sqsubseteq$ Project

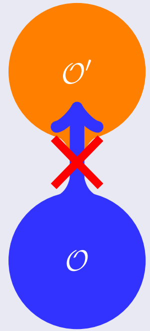
Ontology Reuse



Outline

SAFE REUSE OF ONTOLOGIES

ONTOLOGY REUSE



INFORMALLY

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

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:

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

Informally

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

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:

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

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:

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

Example (2)

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

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

$\mathcal{O}' \cup \mathcal{O}$ 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:

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

Example (2)

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

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

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

The "meaning" of A has changed, but $\mathcal{L} = \mathcal{ALC}$ cannot "see" the change.

Ontology Reuse

SAFE REUSE OF ONTOLOGIES

DEFINITION (2)

$\mathcal{O}' \cup \mathcal{O}$ is a model conservative extension of $\mathcal{O}'$ if every model of $\mathcal{O}'$ can be expanded to a model of $\mathcal{O}' \cup \mathcal{O}$:

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

EXAMPLE (2)

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

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

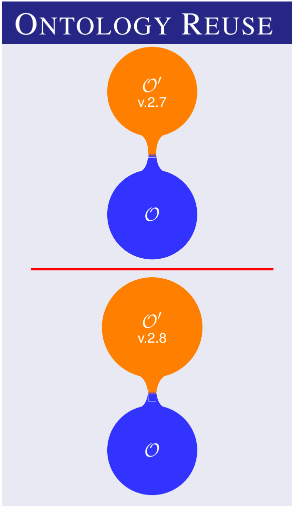
$\mathcal{O}' \cup \mathcal{O}$ is a conservative extension of $\mathcal{O}'$ w.r.t. $\mathcal{L} = \mathcal{ALC}$, but not model conservative

The "meaning" of A has changed, but $\mathcal{L} = \mathcal{ALC}$ cannot "see" the change.

ONTOLOGY REUSE

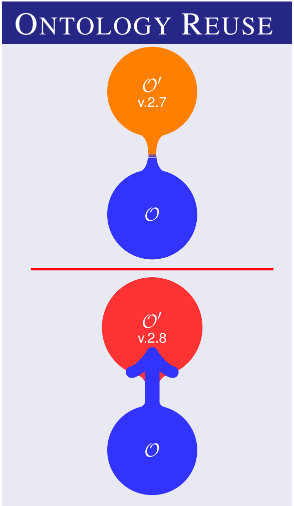
Safety for Evolving Ontologies

- Ontologies evolve over time



SAFETY FOR EVOLVING ONTOLOGIES

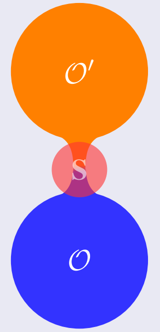
- Ontologies evolve over time
- If $\mathcal{O}$ safely for $\mathcal{O}'$ then it is expected to remain safe for new versions of $\mathcal{O}'$.



Safety for Evolving Ontologies

- Ontologies evolve over time
- If $\mathcal{O}$ safely for $\mathcal{O}'$ then it is expected to remain safe for new versions of $\mathcal{O}'$.
- The notion of safety can be formulated for the interface signature instead

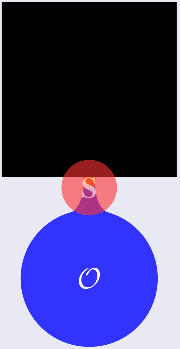
Ontology Reuse



SAFETY FOR EVOLVING ONTOLOGIES

- Ontologies evolve over time
- If $\mathcal{O}$ safely for $\mathcal{O}'$ then it is expected to remain safe for new versions of $\mathcal{O}'$.
- The notion of safety can be formulated for the interface signature instead
- Thus treating $\mathcal{O}'$ as a black box

ONTOLOGY REUSE

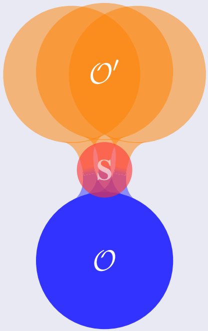


Safety for a Signature

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

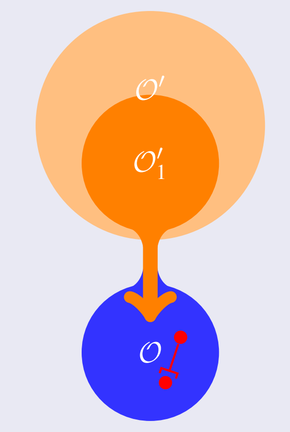


Module for an Ontology

Informally

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 $\mathcal{O}'$ does not change the "meaning" of the symbols in $\mathcal{O}$.

Ontology Reuse



Module for an Ontology

Definition (Module for Ontology)

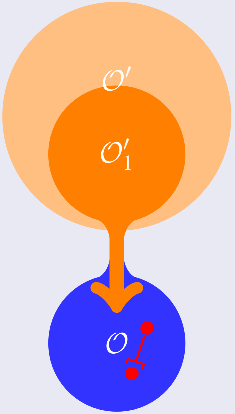
$\mathcal{O}'_1$ is a module in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and ontology language $\mathcal{L}$ if for every axiom α over $\mathcal{O}$ expressed in $\mathcal{L}$, we have:

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

Informally

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 $\mathcal{O}'$ does not change the "meaning" of the symbols in $\mathcal{O}$.

Ontology Reuse

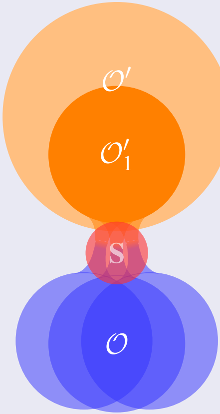


Module for a Signature

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



REASONING PROBLEMS

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

REASONING PROBLEMS

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

- *uncecidable for $\mathcal{L} = \mathcal{ALCQIO}$ [2].*
- *2-ExpTime-complete for $\mathcal{L} = \mathcal{ALC}$ [1] and $\mathcal{L} = \mathcal{ALCQI}$ [2]*
- *ExpTime-complete for $\mathcal{L} = \mathcal{EL}$ [3]*

[1] Ghilardi, Lutz & Wolter, KR'06
[2] Lutz, Walther & Wolter, IJCAI'07 [3] Lutz & Wolter, CADE'07

Reasoning Problems

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

- *uncecidable for $\mathcal{L} = \mathcal{ALCQIO}$ [2].*
- *2-ExpTime-complete for $\mathcal{L} = \mathcal{ALC}$ [1] and $\mathcal{L} = \mathcal{ALCQI}$ [2]*
- *ExpTime-complete for $\mathcal{L} = \mathcal{EL}$ [3]*

Corollary: Then so are the tasks T1 and T3

[1] Ghilardi, Lutz & Wolter, KR'06
[2] Lutz, Walther & Wolter, IJCAI'07 [3] Lutz & Wolter, CADE'07

Reasoning Problems

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

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

Theorem ([3])

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

[3] Cuenca-Grau, Horrocks, Kazakov & Sattler, WWW-2007

Reasoning Problems

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

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

Theorem ([3])

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

Corollary: Then so are the tasks T2 and T4

[3] Cuenca-Grau, Horrocks, Kazakov & Sattler, WWW-2007

Reasoning Problems

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

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

Open Problems

Is safety for a signature decidable for $\mathcal{L} = \mathcal{ALC}$? $\mathcal{L} = \mathcal{EL}$?
If yes, what is the complexity?

Outline

Reasoning Problems: A Pragmatic Approach

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

Reasoning Problems: A Pragmatic Approach

Not	Input	Task "sufficiently"	
T1	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Check whether $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$	☹
T2	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Check whether $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$	☹
T3	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	☹
T4	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Extract minimal module(s) in $\mathcal{O}'$ w.r.t. $\mathbf{S}$ and $\mathcal{L}$	☹

- Develop practical sufficient conditions for safety

Reasoning Problems: A Pragmatic Approach

Not	Input	Task	
T1	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Check whether $\mathcal{O}$ is "sufficiently" safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$	☹
T2	$\mathcal{O}$, **S**, $\mathcal{L}$	Check whether $\mathcal{O}$ is "sufficiently" safe for **S** w.r.t. $\mathcal{L}$	☹
T3	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract ~~minimal~~ small module(s) in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$	☹
T4	$\mathcal{O}$, **S**, $\mathcal{L}$	Extract ~~minimal~~ small module(s) in $\mathcal{O}'$ w.r.t. **S** and $\mathcal{L}$	☹

- Develop practical sufficient conditions for safety
- Develop algorithms for extracting reasonably small but not necessarily minimal modules

A Sufficient Condition for Safety

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 $\mathbf{S}$:

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

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)

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 $\mathbf{S}$:

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

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 $\mathcal{O}$
- We try to interpret every new symbol as the empty set

A Sufficient Condition for Safety

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 $\mathbf{S}$:

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

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 $\mathcal{O}$
- We try to interpret every new symbol as the empty set

Example

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

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 $\mathcal{O}$ 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 $\mathcal{I} \models \mathcal{O}$ for every $\mathcal{I}$ which interprets all concept and role names not in $\mathbf{S}$ as the empty set.

- ✔ If $\mathcal{O}$ is local w.r.t. $\mathbf{S}$ then $\mathcal{O}$ is safe for $\mathbf{S}$:
- ✔ Can be verified using any DL-reasoner.

EXTRACTING LOCALITY-BASED MODULES

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 $\mathbf{S}$.

Extracting Locality-Based Modules

Proposition (Modules and Safety)

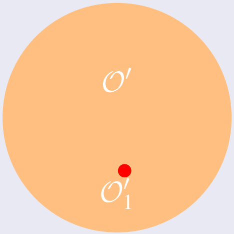
If $\mathcal{O}' \setminus \mathcal{O}'_1$ is safe for $\mathbf{S} \cup \mathsf{Sg}(\mathcal{O}'_1)$
then $\mathcal{O}'_1$ is a module in $\mathcal{O}'$ for $\mathbf{S}$.

Extracting Modules

Given: $\mathcal{O}'$ and $\mathbf{S}$
Compute: a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathbf{S}$

1. Initialize $\mathcal{O}'_1 := \emptyset$
2. Find $\alpha \in \mathcal{O}' \setminus \mathcal{O}'_1$ such that α is not local w.r.t. $\mathbf{S} \cup \mathsf{Sg}(\mathcal{O}'_1)$
3. Move α into $\mathcal{O}'_1$
4. Repeat until fixpoint

Module for S



Extracting Locality-Based Modules

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 $\mathbf{S}$.

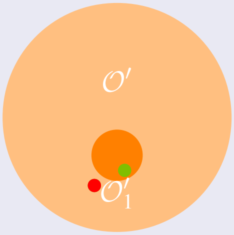
Extracting Modules

Given: $\mathcal{O}'$ and $\mathbf{S}$

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

1. Initialize $\mathcal{O}'_1 := \emptyset$
2. Find $\alpha \in \mathcal{O}' \setminus \mathcal{O}'_1$ such that α is not local w.r.t. $\mathbf{S} \cup \mathsf{Sg}(\mathcal{O}'_1)$
3. Move α into $\mathcal{O}'_1$
4. Repeat until fixpoint

Module for S



Extracting Locality-Based Modules

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 $\mathbf{S}$.

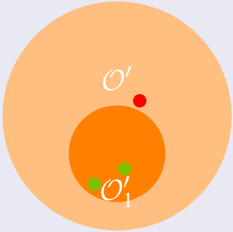
Extracting Modules

Given: $\mathcal{O}'$ and $\mathbf{S}$

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

1. Initialize $\mathcal{O}'_1 := \emptyset$
2. Find $\alpha \in \mathcal{O}' \setminus \mathcal{O}'_1$ such that α is not local w.r.t. $\mathbf{S} \cup \mathsf{Sg}(\mathcal{O}'_1)$
3. Move α into $\mathcal{O}'_1$
4. Repeat until fixpoint

Module for S



EXTRACTING LOCALITY-BASED MODULES

PROPOSITION (MODULES AND SAFETY)

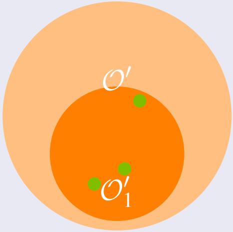
If $\mathcal{O}' \setminus \mathcal{O}'_1$ is safe for $\mathbf{S} \cup \mathsf{Sg}(\mathcal{O}'_1)$
then $\mathcal{O}'_1$ is a module in $\mathcal{O}'$ for $\mathbf{S}$.

EXTRACTING MODULES

Given: $\mathcal{O}'$ and $\mathbf{S}$
Compute: a module $\mathcal{O}'_1$ in $\mathcal{O}'$ w.r.t. $\mathbf{S}$

1. Initialize $\mathcal{O}'_1 := \emptyset$
2. Find $\alpha \in \mathcal{O}' \setminus \mathcal{O}'_1$ such that
   α is not local w.r.t. $\mathbf{S} \cup \mathsf{Sg}(\mathcal{O}'_1)$
3. Move α into $\mathcal{O}'_1$
4. Repeat until fixpoint

MODULE FOR S



Empirical Results

Ontology	♯ Atomic Concepts	A1: Prompt-Factor [1]		A2: Mod. in [2]		A3: Loc.-based mod.	
		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

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

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

Our Contributions

- Formalization for the notions for safety and modules using 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

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