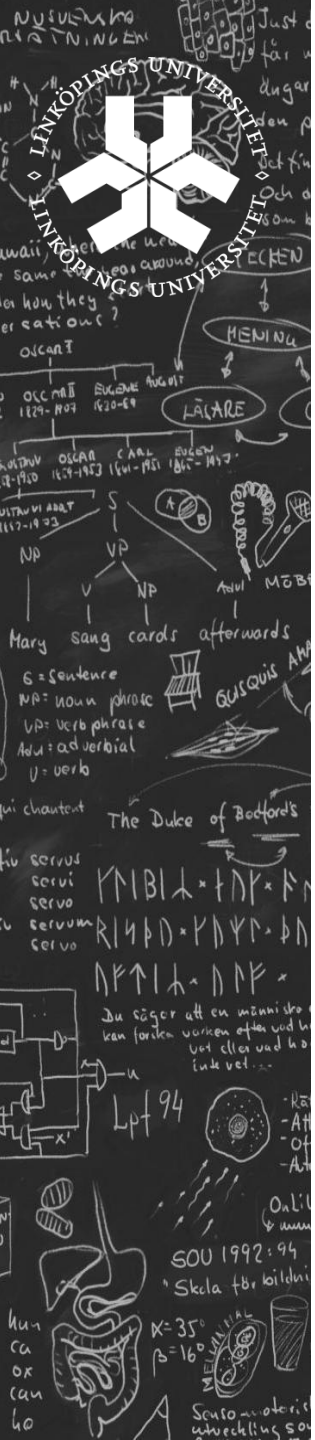


Get my pizza right: Repairing missing is-a relations in ALC ontologies

Patrick Lambrix, Zlatan Dragisic and Valentina Ivanova

Linköping University
Sweden

LiU expanding reality

Introduction

- Developing ontologies is not an easy task
- It may happen that ontologies are
  - Incorrect
  - Incomplete

Defects in ontologies

- Syntactic defects
  - e.g. wrong tags or incorrect format
- Semantic defects
  - e.g. unsatisfiable concepts, incoherent and inconsistent ontologies
- Modeling defects
  - e.g. wrong or missing relations

Example – missing is-a relations

- In 2008 Ontology Alignment Evaluation Initiative (OAEI) Anatomy track, task 4
  - Ontology MA : Adult Mouse Anatomy Dictionary (2744 concepts)
  - Ontology NCI-A : NCI Thesaurus - anatomy (3304 concepts)
  - 988 mappings between MA and NCI-A
    - 121 missing is-a relations in MA
    - 83 missing is-a relations in NCI-A

Influence of defects in structure

- Ontology-based querying.

Medical Subject Headings (MeSH)

All MeSH Categories
- **Diseases Category**
  - **Eye Diseases**
    - **Scleral Diseases**
      - **Scleritis**

...

return 1363 articles

Influence of defects in structure

- Incomplete results from ontology-based queries

Medical Subject Headings (MeSH)

All MeSH Categories
- **Diseases Category**
  - **Eye Diseases**
    - **Scleral Diseases**
      - ~~**Scleritis**~~

...

return 1363 articles

return 613 articles

55% results are missed !

Debugging defects

- Two phases:
  - Detection
  - Repair
    - Missing is-a relations: TBox abduction problem

Example – missing is-a relations

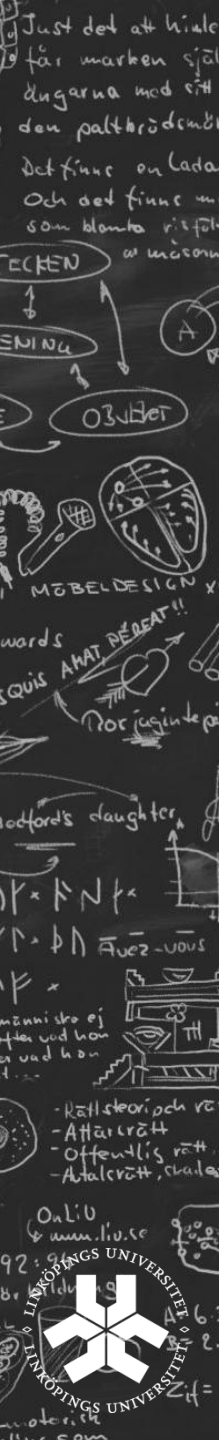
$$\begin{aligned}
&\text{Pizza} \dot{\sqsubseteq} \top \\
&\text{PizzaTopping} \dot{\sqsubseteq} \top \\
&\text{hasTopping} \dot{\sqsubseteq} \top \times \top \\
&\text{AnchoviesTopping} \dot{\sqsubseteq} \text{PizzaTopping} \\
&\text{MeatTopping} \dot{\sqsubseteq} \text{PizzaTopping} \\
&\text{HamTopping} \dot{\sqsubseteq} \text{MeatTopping} \\
&\text{ParmaHamTopping} \dot{\sqsubseteq} \text{PizzaTopping} \\
&\text{FishTopping} \dot{\sqsubseteq} \text{PizzaTopping} \sqcap \neg\text{MeatTopping} \\
&\text{TomatoTopping} \dot{\sqsubseteq} \text{PizzaTopping} \sqcap \neg\text{MeatTopping} \sqcap \neg\text{FishTopping} \\
&\text{GarlicTopping} \dot{\sqsubseteq} \text{PizzaTopping} \sqcap \neg\text{MeatTopping} \sqcap \neg\text{FishTopping} \\
&\underline{\text{MyPizza} \doteq \text{Pizza} \sqcap \exists\, \text{hasTopping.AnchoviesTopping} \sqcap \exists\, \text{hasTopping.ParmaHamTopping}} \\
&\underline{\text{FishyMeatyPizza} \doteq \text{Pizza} \sqcap \exists\, \text{hasTopping.FishTopping} \sqcap \exists\, \text{hasTopping.MeatTopping}} \\
&\text{MyFruttiDiMare} \doteq \text{Pizza} \sqcap \exists\, \text{hasTopping.AnchoviesTopping} \\
&\qquad\qquad \sqcap \exists\, \text{hasTopping.GarlicTopping} \sqcap \exists\, \text{hasTopping.TomatoTopping} \\
&\qquad\qquad \sqcap \forall\, \text{hasTopping.(AnchoviesTopping} \sqcup \text{GarlicTopping} \sqcup \text{TomatoTopping)} \\
&\text{VegetarianPizza} \doteq \text{Pizza} \sqcap \neg \exists\, \text{hasTopping.FishTopping} \sqcap \neg \exists\, \text{hasTopping.MeatTopping} \\
&\text{NonVegetarianPizza} \doteq \text{Pizza} \sqcap \neg\text{VegetarianPizza}
\end{aligned}$$

Missing relation:

$$\text{MyPizza} \dot{\sqsubseteq} \text{FishyMeatyPizza}$$

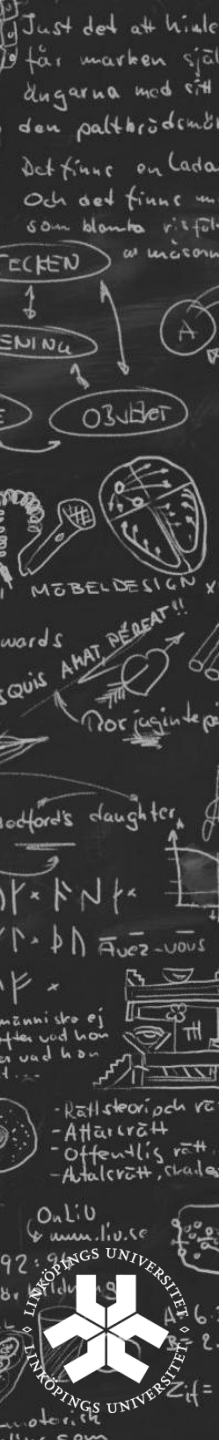
Repairing actions:

$$\{\text{MyPizza} \dot{\sqsubseteq} \text{FishyMeatyPizza}\}$$
$$\{\text{AnchoviesTopping} \dot{\sqsubseteq} \text{FishTopping},\ \text{ParmaHamTopping} \dot{\sqsubseteq} \text{MeatTopping}\}$$
$$\{\text{AnchoviesTopping} \dot{\sqsubseteq} \text{MeatTopping},\ \text{ParmaHamTopping} \dot{\sqsubseteq} \text{FishTopping}\}$$



Outline

- Definitions
- Approach
- Implementation
- Future work



Outline

- Definitions
- Approach
- Implementation
- Future work

Generalized TBox abduction problem

- Given
  - *KB* - a knowledge base in $\mathcal{L}$
  - $\{ C_i \sqsubseteq D_i \mid 1 \leq i \leq m \}$ where C_i, D_i are satisfiable w.r.t. *KB*
  - $KB \cup \{ C_i \sqsubseteq D_i \mid 1 \leq i \leq m \}$ is coherent
- Find
  - $S_{GT}=\{G_j \sqsubseteq H_j \mid j \leq n\}$
    such that $KB \cup S_{GT} \models C_i \sqsubseteq D_i$
- In our setting C_i, D_i, G_j, H_j are named concepts

Description logic ALC

Concepts

Atomic concept	A
Universal concept	T
Bottom concept	⊥
Concept negation	¬C
Intersection of concepts	C ⊓ D
Union of concepts	C ⊔ D
Universal restriction	∀R.C
Existential restriction	∃R.C

Axioms

- Terminological axioms (C ⊑ D, C ≡ D) - TBox
- Assertional axioms (C(a), R(a,b)) – ABox

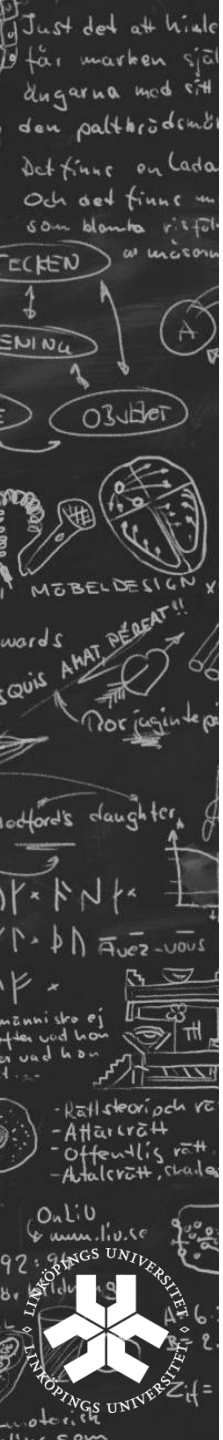
Acyclic terminologies

- Acyclic terminology – finite set of concept definitions (i.e. terminological axioms of the form $C \doteq D$ where C is a concept name) that neither contain mutiple definitions nor cyclic definitions
  - $\text{MeatTopping} \sqsubseteq \text{PizzaTopping}$

    could be replaced with

    $\text{MeatTopping} \doteq \text{PizzaTopping} \sqcap \overline{\text{MeatTopping}}$

    where $\overline{\text{MeatTopping}}$ is a new atomic concept



Outline

- Definitions
- Approach
- Implementation
- Future work

Basic algorithm

- Input – Ontology represented by a knowledge base and a set of missing is-a relations
- Output – set of repairing actions
- Two main steps:
  - Creating repairing actions for individual missing is-a relations
  - Creating repairing actions for all missing is-a relations

Repairing missing is-a relations using tableaux-based algorithm

- For a missing is-a relation $C \sqsubseteq D$ run a tableaux algorithm on x: $C \sqcap \neg D$
- Try to find set of is-a relations $\{G_1 \sqsubseteq H_1, \ldots, G_n \sqsubseteq H_n\}$ which would close open branches in the completion graph

Tableaux-based algorithm

⊓-rule: if the ABox contains $(C_1 \sqcap C_2)(x)$, but it does not contain both $C_1(x)$ and $C_2(x)$,
then these are added to the ABox.
⊔-rule: if the ABox contains $(C_1 \sqcup C_2)(x)$, but it contains neither $C_1(x)$ nor $C_2(x)$,
then two ABoxes are created representing the two choices of adding $C_1(x)$ or adding $C_2(x)$.
∀-rule: if the ABox contains $(\forall\ r.C)(x)$ and r(x,y), but it does not contain C(y),
then this is added to the ABox.
∃-rule: if the ABox contains $(\exists\ r.C)(x)$ but there is no individual z such that r(x,z) and C(z) are in the ABox,
then r(x,y) and C(y) with y an individual name not occurring in the ABox, are added.

ABox 1:
(1) x: MyPizza ⊓ ¬FishyMeatyPizza

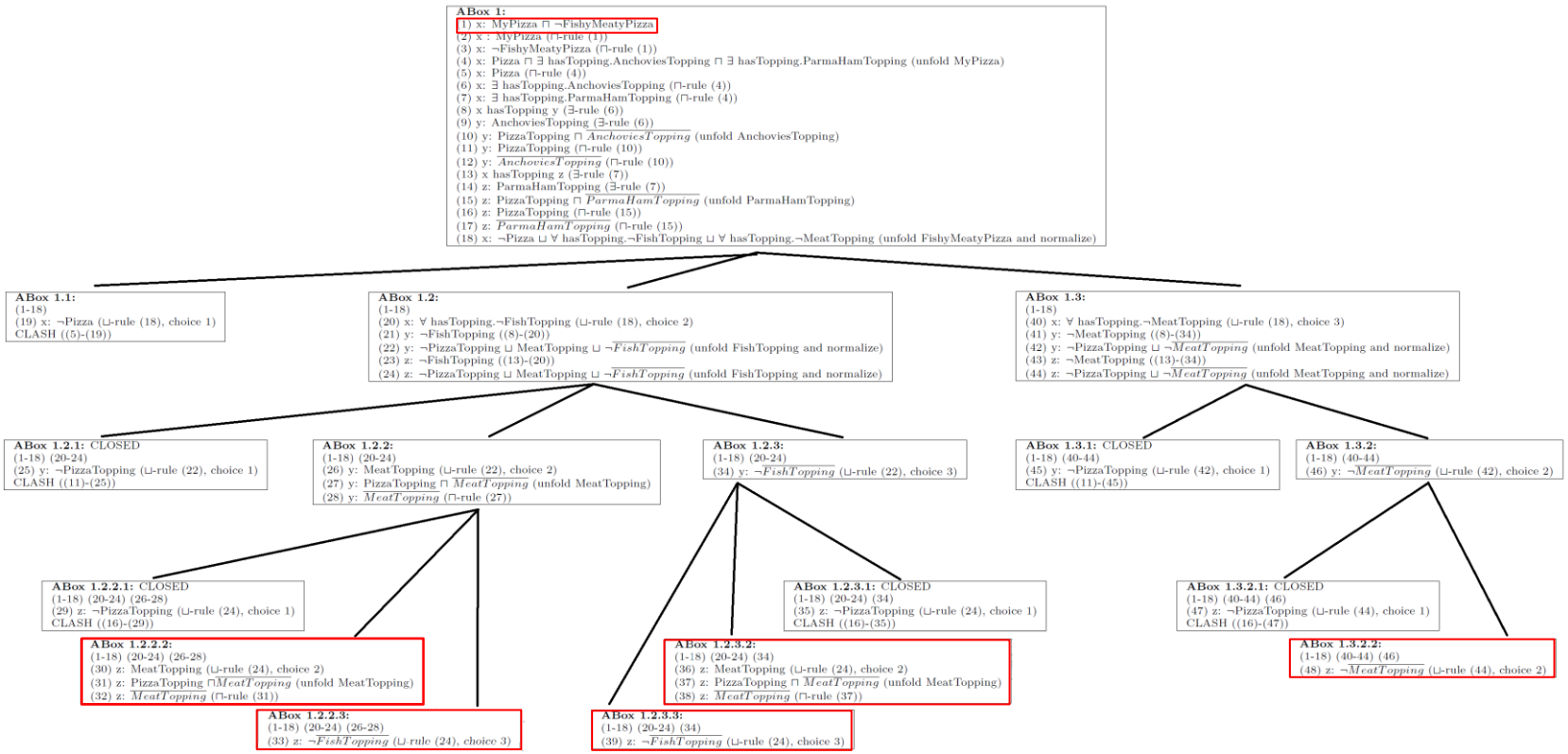
ABox 1.1:
(1-18)
(19) x: ¬Pizza (⊔-rule (18), choice 1)

ABox 1.2:
(1-18)
(20) x: ∀ hasTopping.¬FishTopping (⊔-rule (18), choice 2)

ABox 1.3:
(1-18)
(40) x: ∀ hasTopping.¬MeatTopping (⊔-rule (18), choice 3)

Basic algorithm - example

⊓-rule: if the ABox contains $(C_1 \sqcap C_2)(x)$, but it does not contain both $C_1(x)$ and $C_2(x)$,
then these are added to the ABox.
⊔-rule: if the ABox contains $(C_1 \sqcup C_2)(x)$, but it contains neither $C_1(x)$ nor $C_2(x)$,
then two ABoxes are created representing the two choices of adding $C_1(x)$ or adding $C_2(x)$.
∀-rule: if the ABox contains $(\forall\ r.C)(x)$ and $r(x,y)$, but it does not contain $C(y)$,
then this is added to the ABox.
∃-rule: if the ABox contains $(\exists\ r.C)(x)$ but there is no individual z such that $r(x,z)$ and $C(z)$ are in the ABox,
then $r(x,y)$ and $C(y)$ with y an individual name not occurring in the ABox, are added.



Basic algorithm - example

ABox 1.2.2.2:

(1) x: MyPizza ⊓ ¬FishyMeatyPizza
(2) x : MyPizza (⊓-rule (1))
(3) x: ¬FishyMeatyPizza (⊓-rule (1))
(4) x: Pizza ⊓ ∃ hasTopping.AnchoviesTopping ⊓ ∃ hasTopping.ParmaHamTopping (unfold MyPizza)
(5) x: Pizza (⊓-rule (4))
(6) x: ∃ hasTopping.AnchoviesTopping (⊓-rule (4))
(7) x: ∃ hasTopping.ParmaHamTopping (⊓-rule (4))
(8) x hasTopping y (∃-rule (6))
(9) y: AnchoviesTopping (∃-rule (6))
(10) y: PizzaTopping ⊓ $\overline{AnchoviesTopping}$ (unfold AnchoviesTopping)
(11) y: PizzaTopping (⊓-rule (10))
(12) y: $\overline{AnchoviesTopping}$ (⊓-rule (10))
(13) x hasTopping z (∃-rule (7))
(14) z: ParmaHamTopping (∃-rule (7))
(15) z: PizzaTopping ⊓ $\overline{ParmaHamTopping}$ (unfold ParmaHamTopping)
(16) z: PizzaTopping (⊓-rule (15))
(17) z: $\overline{ParmaHamTopping}$ (⊓-rule (15))
(18) x: ¬Pizza ⊔ ∀ hasTopping.¬FishTopping ⊔ ∀ hasTopping.¬MeatTopping (unfold FishyMeatyPizza and normalize)
(20) x: ∀ hasTopping.¬FishTopping (⊔-rule (18), choice 2)
(21) y: ¬FishTopping ((8)-(20))
(22) y: ¬PizzaTopping ⊔ MeatTopping ⊔ ¬$\overline{FishTopping}$ (unfold FishTopping and normalize)
(23) z: ¬FishTopping ((13)-(20))
(24) z: ¬PizzaTopping ⊔ MeatTopping ⊔ ¬$\overline{FishTopping}$ (unfold FishTopping and normalize)
(26) y: MeatTopping (⊔-rule (22), choice 2)
(27) y: PizzaTopping ⊓ $\overline{MeatTopping}$ (unfold MeatTopping)
(28) y: $\overline{MeatTopping}$ (⊓-rule (27))
(30) z: MeatTopping (⊔-rule (24), choice 2)
(31) z: PizzaTopping ⊓$\overline{MeatTopping}$ (unfold MeatTopping)
(32) z: $\overline{MeatTopping}$ (⊓-rule (31))

x: MyPizza
x: ¬FishyMeatyPizza
x: Pizza

$\dot{\sqsubseteq}$ MyPizza $\dot{\sqsubseteq}$ FishyMeatyPizza
$\dot{\sqsubseteq}$ Pizza $\dot{\sqsubseteq}$ FishyMeatyPizza

$R_{\mathcal{A}} = \{$

Basic algorithm - example

- Repairing actions for a missing is-a relation created by choosing one element from each set R_A
- Example – 5 open ABoxes

$$
\begin{array}{l}
\{\text{MyPizza} \dot{\sqsubseteq} \text{FishyMeatyPizza}\} \\
\{\text{Pizza} \dot{\sqsubseteq} \text{FishyMeatyPizza}\}^* \\
\{\text{AnchoviesTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{AnchoviesTopping} \dot{\sqsubseteq} \text{MeatTopping}\}^* \\
\{\text{PizzaTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{AnchoviesTopping} \dot{\sqsubseteq} \text{MeatTopping}\}^* \\
\{\text{ParmaHamTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{AnchoviesTopping} \dot{\sqsubseteq} \text{MeatTopping}\} \\
\{\text{AnchoviesTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{PizzaTopping} \dot{\sqsubseteq} \text{MeatTopping}\}^* \\
\{\text{PizzaTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{PizzaTopping} \dot{\sqsubseteq} \text{MeatTopping}\}^* \\
\{\text{ParmaHamTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{PizzaTopping} \dot{\sqsubseteq} \text{MeatTopping}\}^* \\
\{\text{AnchoviesTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{ParmaHamTopping} \dot{\sqsubseteq} \text{MeatTopping}\} \\
\{\text{PizzaTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{ParmaHamTopping} \dot{\sqsubseteq} \text{MeatTopping}\}^* \\
\{\text{ParmaHamTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{ParmaHamTopping} \dot{\sqsubseteq} \text{MeatTopping}\}^*
\end{array}
$$

Basic algorithm

- Same process repeated for other missing is-a relations
- Repairing actions for all missing is-a relations created by combining repairing actions for the individual missing is-a relations

Example – multiple missing is-a relations

Missing is-a relations

$$\{\text{MyPizza} \dot{\sqsubseteq} \text{FishyMeatyPizza}, \text{MyFruttiDiMare} \dot{\sqsubseteq} \text{NonVegetarianPizza}\}$$

$$\text{MyPizza} \dot{\sqsubseteq} \text{FishyMeatyPizza}$$

$$\{\text{MyPizza} \dot{\sqsubseteq} \text{FishyMeatyPizza}\}$$
$$\{\text{AnchoviesTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{ParmaHamTopping} \dot{\sqsubseteq} \text{MeatTopping}\}$$
$$\{\text{ParmaHamTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{AnchoviesTopping} \dot{\sqsubseteq} \text{MeatTopping}\}$$

$$\text{MyFruttiDiMare} \dot{\sqsubseteq} \text{NonVegetarianPizza}$$

$$\{\text{MyFruttiDiMare} \dot{\sqsubseteq} \text{NonVegetarianPizza}\}$$
$$\{\text{AnchoviesTopping} \dot{\sqsubseteq} \text{FishTopping}\}$$
$$\{\text{AnchoviesTopping} \dot{\sqsubseteq} \text{MeatTopping}\}$$

Example – multiple missing is-a relations

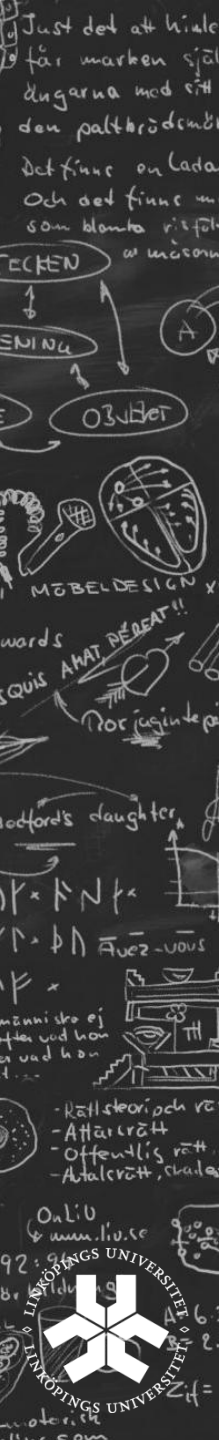
$\{\text{MyPizza} \dot{\sqsubseteq} \text{FishyMeatyPizza}\}$
$\{\text{AnchoviesTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{ParmaHamTopping} \dot{\sqsubseteq} \text{MeatTopping}\}$
$\{\text{ParmaHamTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{AnchoviesTopping} \dot{\sqsubseteq} \text{MeatTopping}\}$

$\{\text{MyFruttiDiMare} \dot{\sqsubseteq} \text{NonVegetarianPizza}\}$
$\{\text{AnchoviesTopping} \dot{\sqsubseteq} \text{FishTopping}\}$
$\{\text{AnchoviesTopping} \dot{\sqsubseteq} \text{MeatTopping}\}$

$\{\text{MyPizza} \dot{\sqsubseteq} \text{FishyMeatyPizza}, \text{MyFruttiDiMare} \dot{\sqsubseteq} \text{NonVegetarianPizza}\}$
$\{\text{AnchoviesTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{ParmaHamTopping} \dot{\sqsubseteq} \text{MeatTopping}\}$
$\{\text{ParmaHamTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{AnchoviesTopping} \dot{\sqsubseteq} \text{MeatTopping}\}$
$\{\text{MyPizza} \dot{\sqsubseteq} \text{FishyMeatyPizza}, \text{AnchoviesTopping} \dot{\sqsubseteq} \text{FishTopping}\}$
$\{\text{MyPizza} \dot{\sqsubseteq} \text{FishyMeatyPizza}, \text{AnchoviesTopping} \dot{\sqsubseteq} \text{MeatTopping}\}$

Algorithm - properties

- Sound
- Minimal solutions
- Solutions do not introduce incoherence



Algorithm - extension

- Finding additional solutions

  Example:

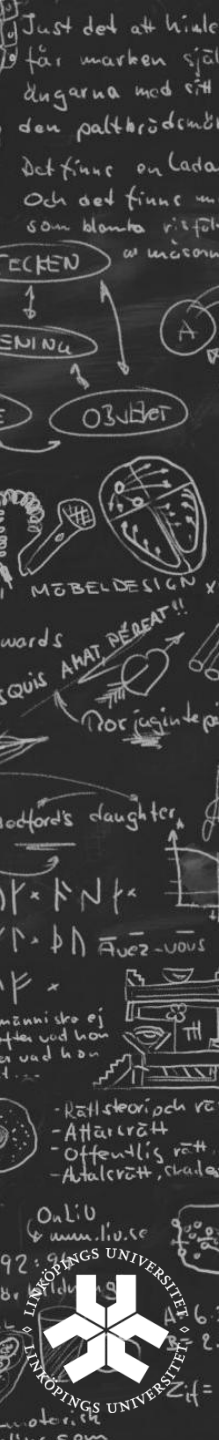
  $\{\text{AnchoviesTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{ParmaHamTopping} \dot{\sqsubseteq} \text{MeatTopping}\}$

  Given: $\text{HamTopping} \dot{\sqsubseteq} \text{MeatTopping}$

  $\{\text{AnchoviesTopping} \dot{\sqsubseteq} \text{FishTopping}, \text{ParmaHamTopping} \dot{\sqsubseteq} \text{HamTopping}\}$

- Is-a relation $A \sqsubseteq B$ in a repairing action can be replaced with $P \sqsubseteq Q$ where P is a super-concept of A and Q is a sub-concept of B
- Source and Target sets



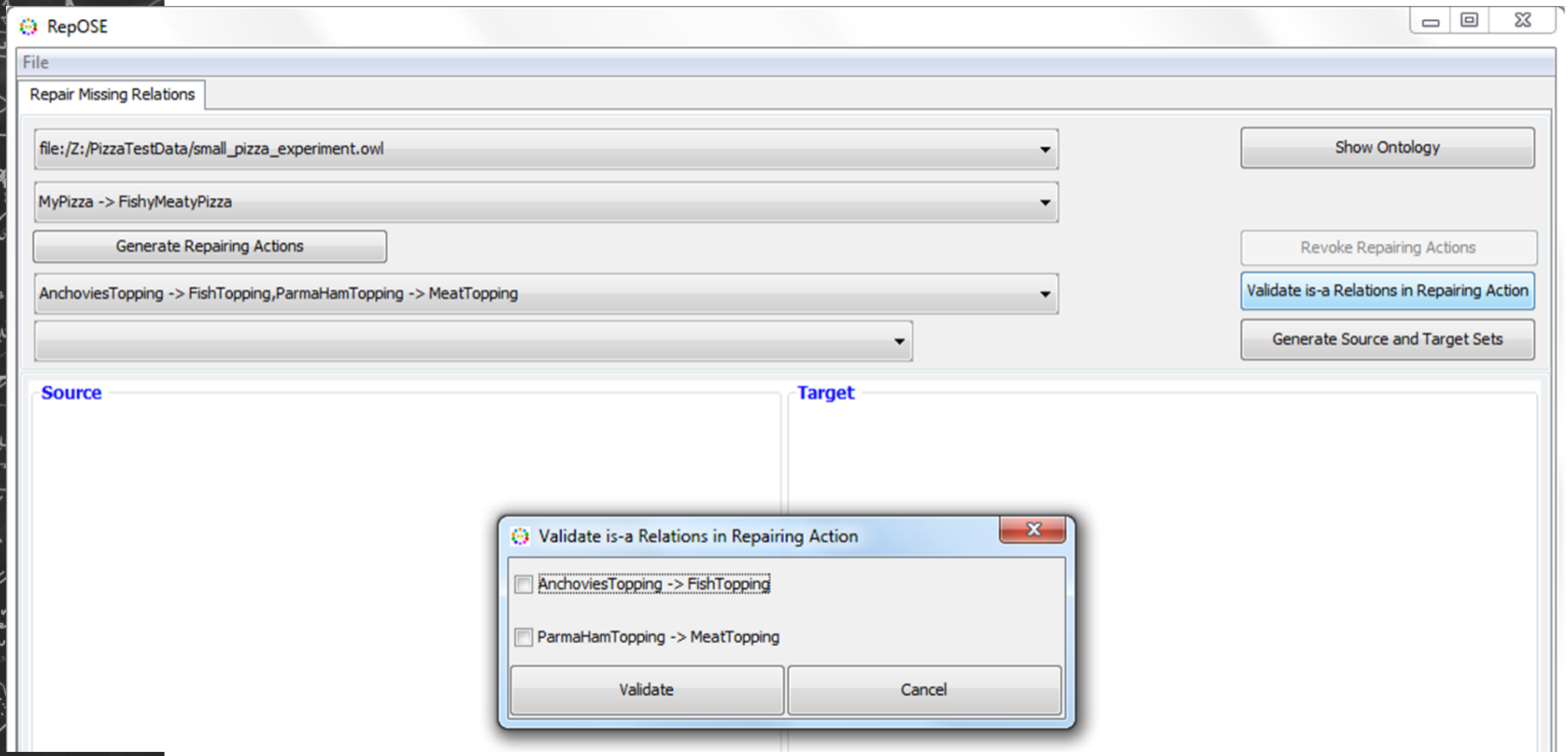
Outline

- Definitions
- Approach
- Implementation
- Future work

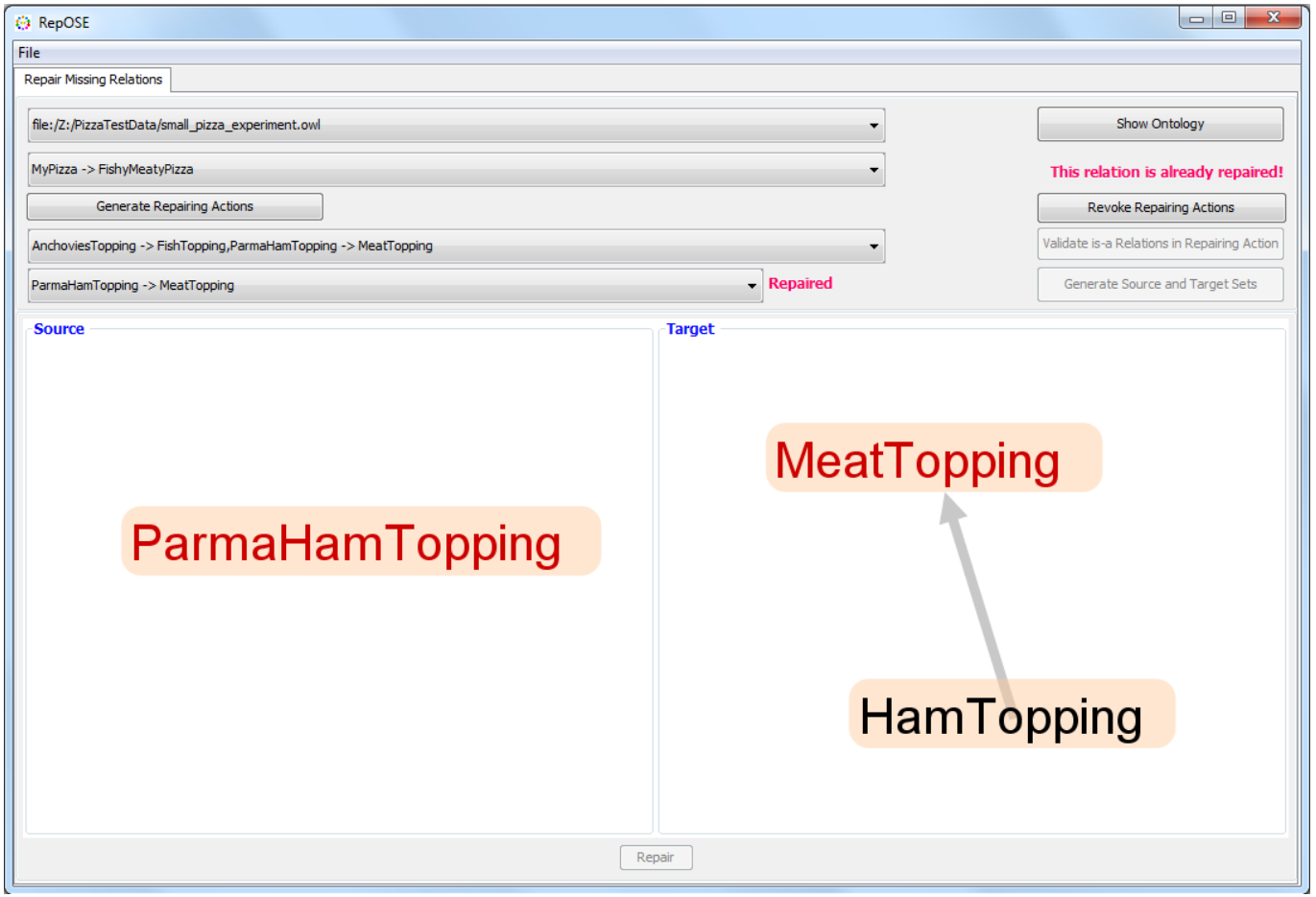
Implementation

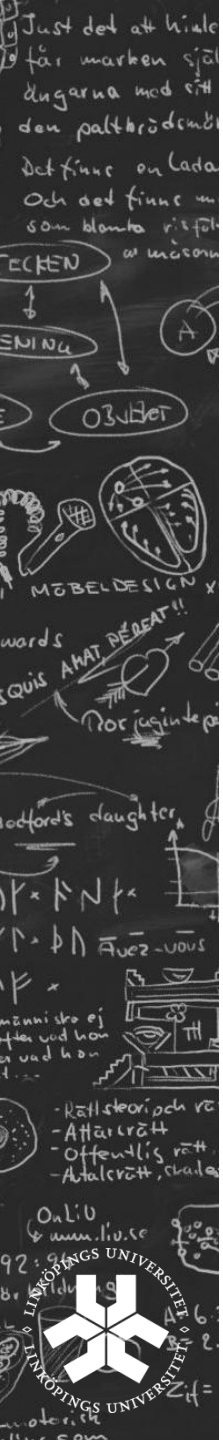
- System for repairing missing is-a relations in ALC acyclic terminologies
- Implemented in Java
- Uses Pellet
- Pellet modified to extract the completion graph

Implementation



Implementation





Outline

- Definitions
- Approach
- Implementation
- Future work

Future work

- Detecting missing is-a relations
- Debugging wrong is-a relations
- Debugging is-a relations and mappings in ontology networks