Repairing Networks of $\mathcal{EL}_\bot$ Ontologies using Weakening and Completing

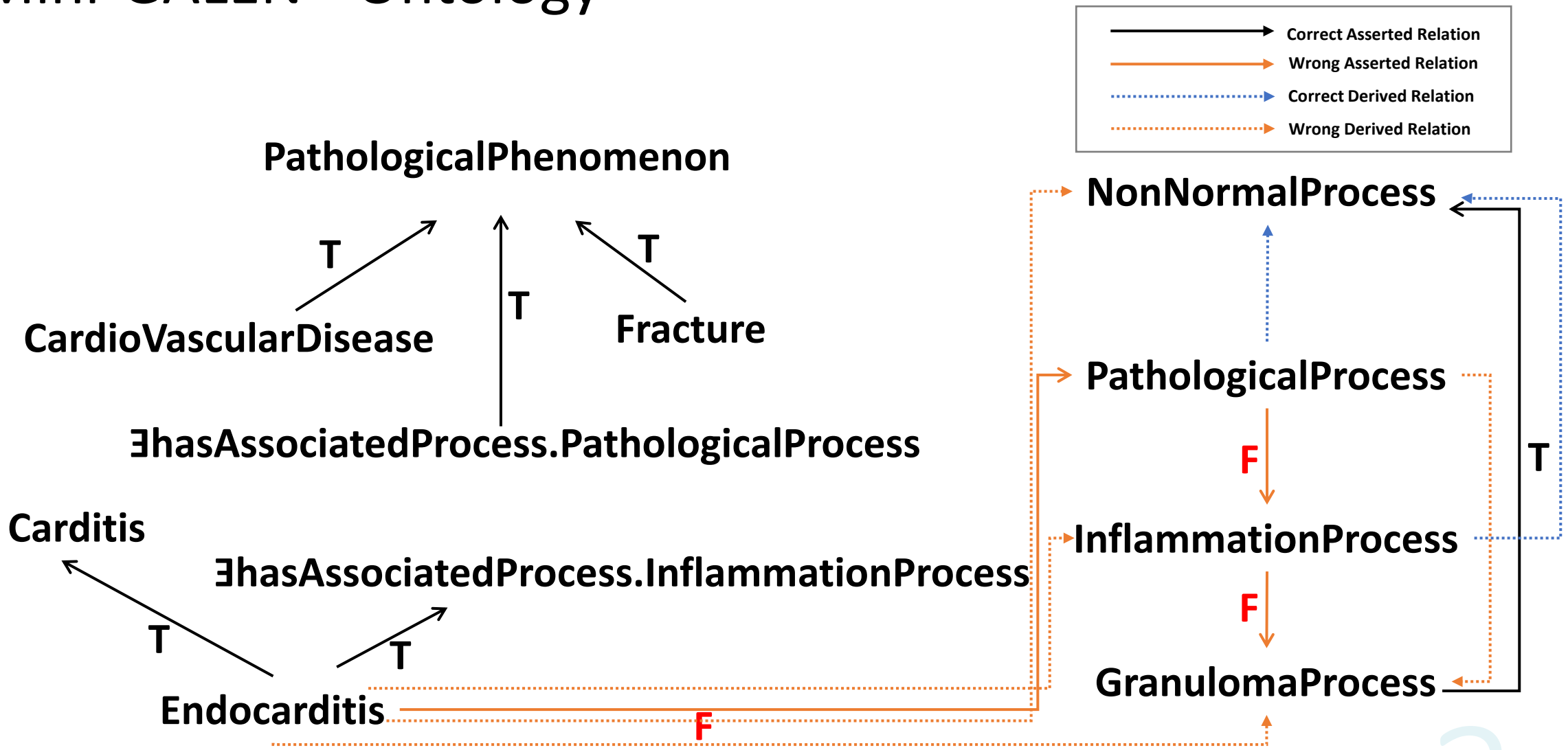
Ying Li, Patrick Lambrix

ISWC 2024

LINKÖPING UNIVERSITY

Motivation

Mini-GALEN[1] Ontology



Motivation

Preserve as much as possible correct knowledge when removing wrong axioms from the ontology/ontology network.

A domain expert (Oracle)
will validate the results

We propose an interactive repairing approach to mitigate these effects of removing wrong axioms by, in addition to removing those axioms, also adding correct knowledge.

Completeness:
Weakening
Completion

Correctness:
Debugging
Removing

Preliminaries

Ontology network

Ontologies are represented using DL TBoxes.

An alignment between two ontologies is a set of mappings between the ontologies.

Definition 1. Let $\mathcal{T}_1, ..., \mathcal{T}_n$ be TBoxes representing ontologies $\mathcal{O}_1, ..., \mathcal{O}_n$, respectively. For $i, j \in [1..n]$ with $i < j$, let $\mathcal{A}_{ij}$ be an alignment between ontology $\mathcal{O}_i$ and $\mathcal{O}_j$. The network of the ontologies and their alignments is then represented by TBox $\mathcal{T} = (\bigcup_{i=1..n} \mathcal{T}_i) \cup (\bigcup_{i,j=1..n,i<j} \mathcal{A}_{ij})$.

Problem Formulation

An ontology/ontology network represented by TBox T

Domain expert

Definition 4. (Repair) Let TBox $\mathcal{T} = (\bigcup_{i=1..n} \mathcal{T}_i) \cup (\bigcup_{i,j=1..n,i<j} \mathcal{A}_{ij})$ represent a network of ontologies $\mathcal{O}_i$ represented by TBoxes $\mathcal{T}_i$, and their alignments $\mathcal{A}_{ij}$. Let Or be an oracle that given a TBox axiom returns true or false. Let W be a finite set of TBox axioms in $\mathcal{T}$ such that $\forall\, \psi \in W$: $Or(\psi) =$ false. Then, a repair for Debug-Problem $\mathrm{DP}(\mathcal{T}, Or, W)$ is a tuple (A, D) where A and D are finite sets of TBox axioms such that

(i) $\forall\, \psi \in A$: $Or(\psi) =$ true;
(ii) D is a finite set of *asserted* axioms in $\mathcal{T}$;
(iii) $\forall\, \psi \in D$: $Or(\psi) =$ false;
(iv) $\forall\, \psi \in W$: $(\mathcal{T} \cup A) \setminus D \not\models \psi$.

W is a set of the wrong axioms to remove from the ontology/ontology network.

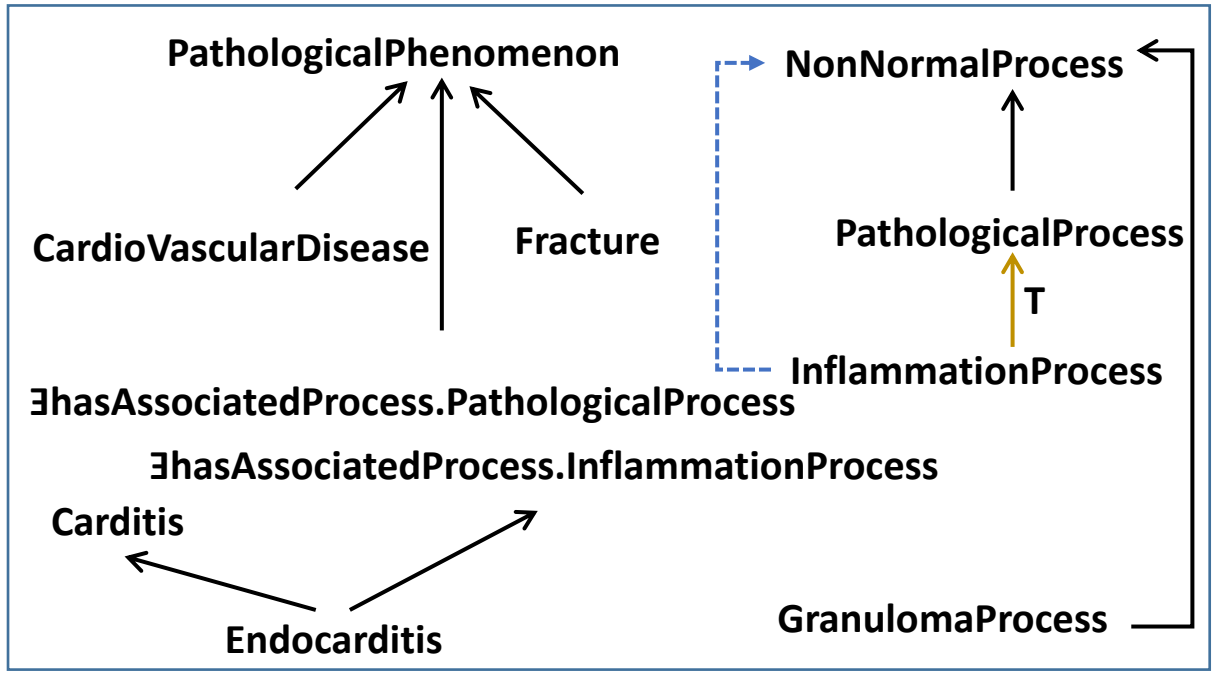
A repair (A,D) is a tuple containing two sets:
A: a set of correct axioms to add to the TBox
D: a set of wrong asserted axioms to remove from the Tbox.

When the axioms in D are removed and the axioms in A are added, the wrong axioms in W cannot be derived anymore.

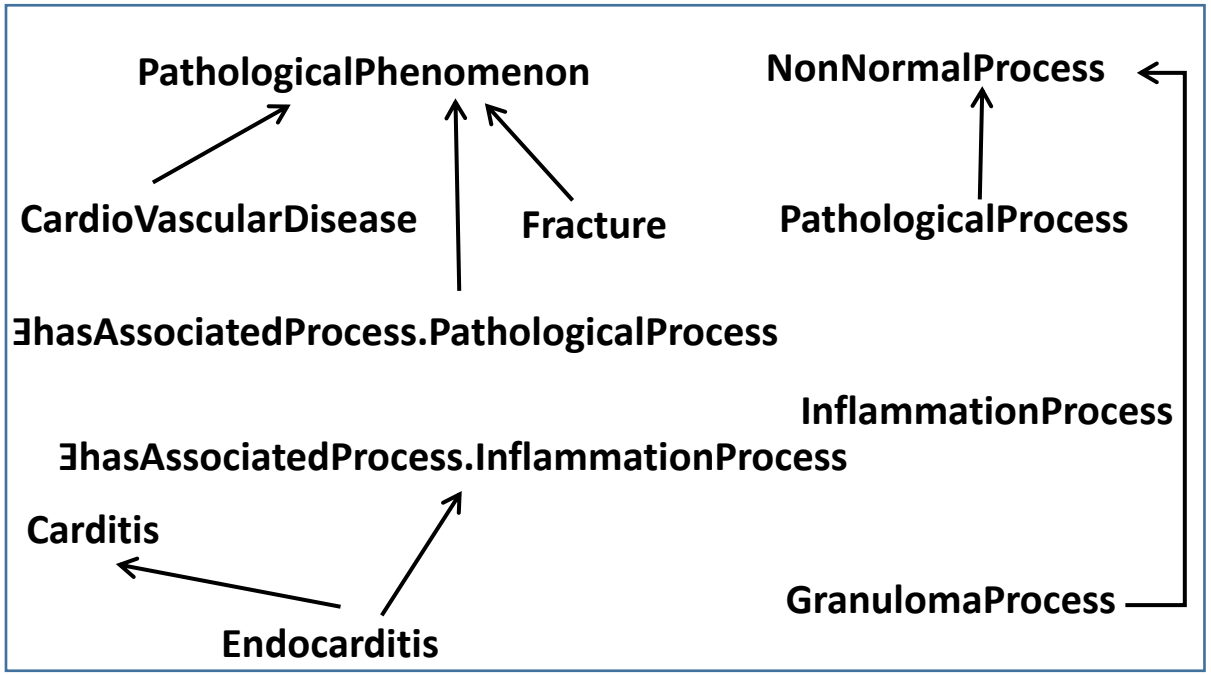
More complete TBoxes

Definition 2. TBox T_1 is more complete than *TBox T_2* iff

1) all correct knowledge in T_2 can also be derived in T_1.
2) there is correct knowledge in T_1 that cannot be derived in T_2.



T1



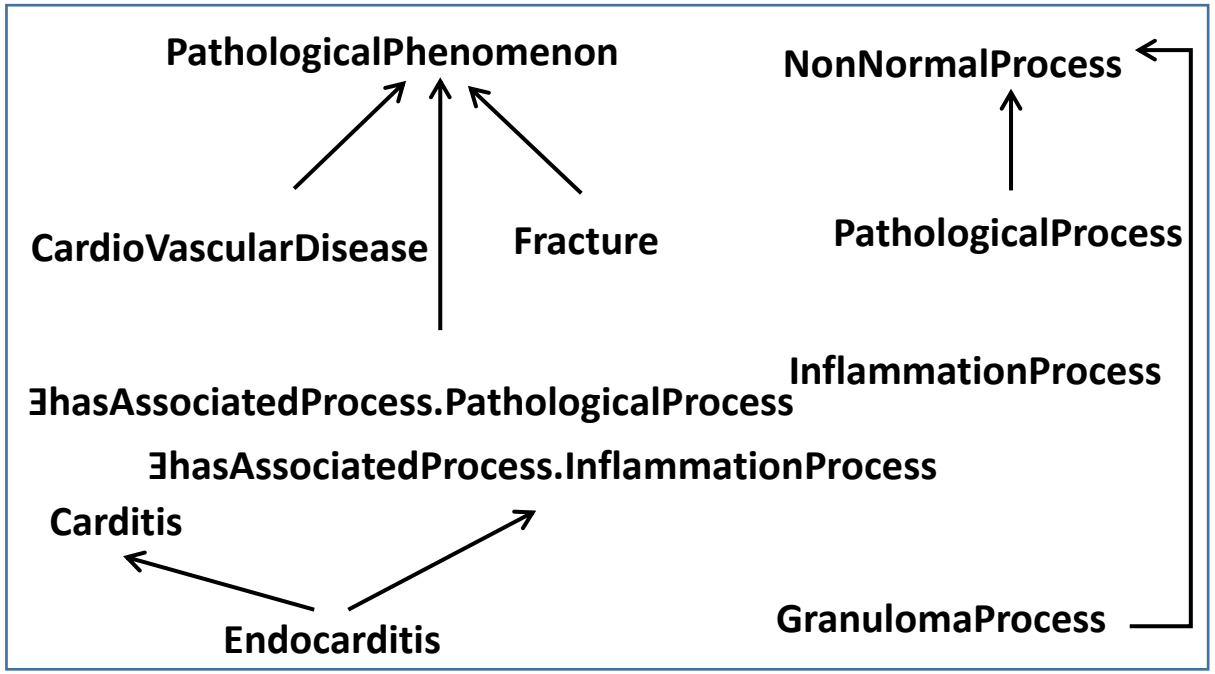
T2

Or(InflammationProcess ⊑ PathologicalProcess) = true
T_1 is more complete than T_2

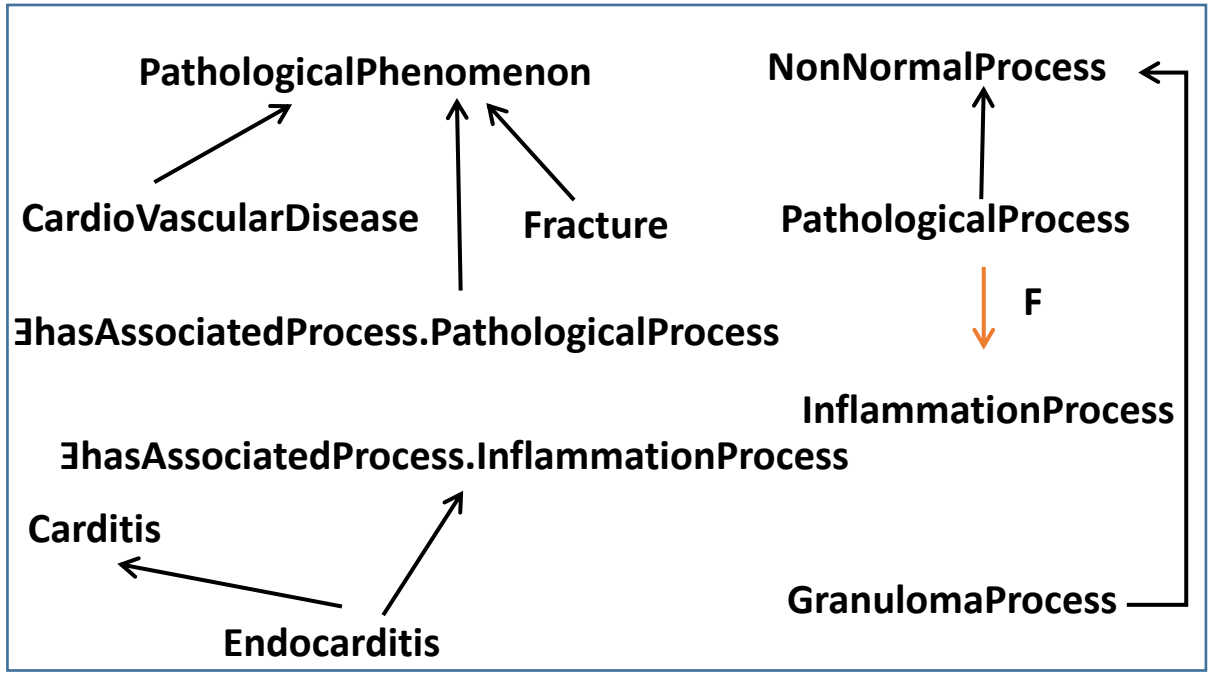
Less incorrect TBoxes

Definition 3. TBox T_1 is less incorrect than *TBox T_2* iff

1) all incorrect knowledge in T_1 can also be derived in T_2.
2) there is incorrect knowledge in T_2 that cannot be derived in T_1.



T1

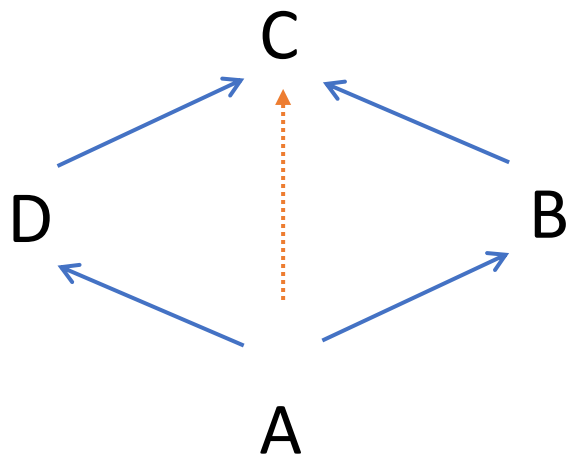


T2

Or(PathologicalProcess ⊑ InflammationProcess) = false

T_1 is less incorrect than T_2

Debugging and Removing- dealing with wrong axioms

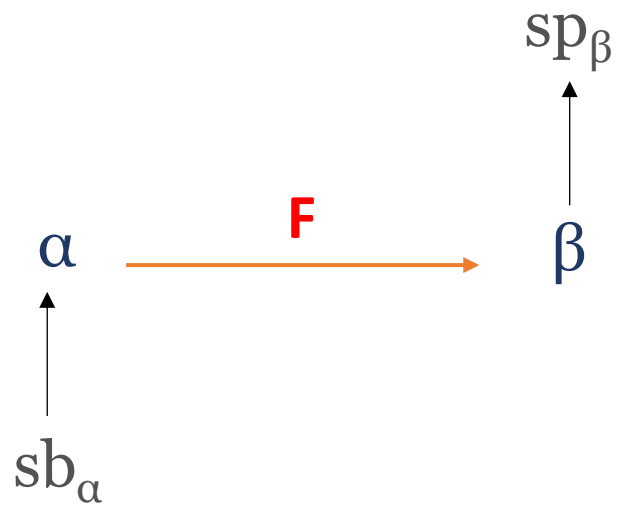


Debugging and removing lead to less incorrect ontologies.

Many algorithms exist.

Justifications($A \sqsubseteq C$)={ {$A \sqsubseteq B$, $B \sqsubseteq C$}, [$A \sqsubseteq D$, $D \sqsubseteq C$} }

Weakening - finding correct weaker axioms



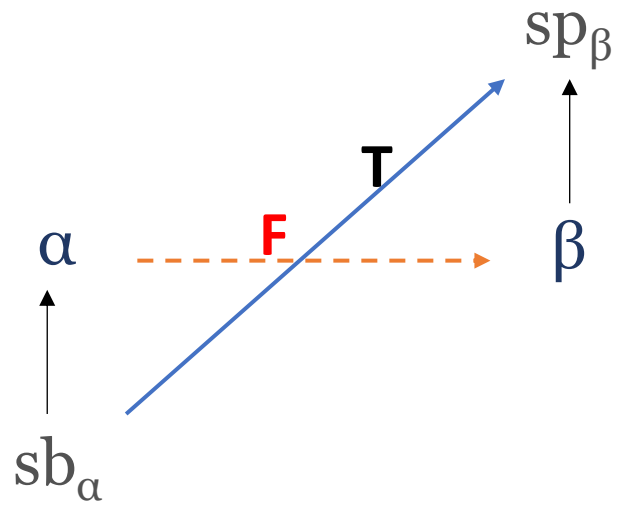
Weakening - finding correct weaker axioms

Replace wrong $\alpha \sqsubseteq \beta$

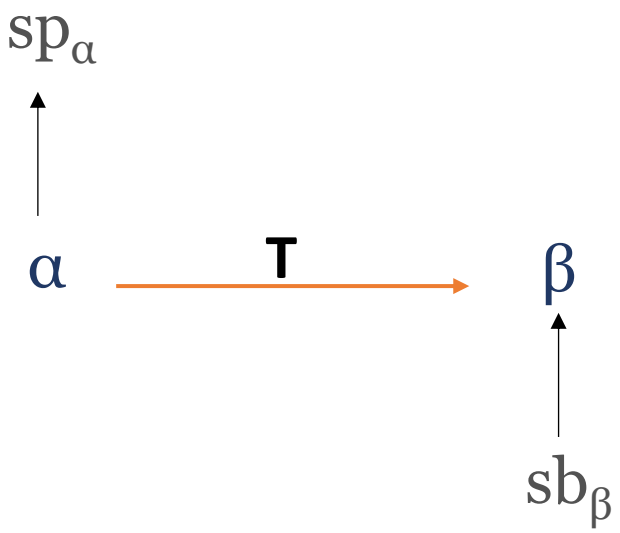
with correct $sb_\alpha \sqsubseteq sp_\beta$

Weakening leads to more complete ontologies.

Some algorithms exist.



Completing - finding correct stronger axioms



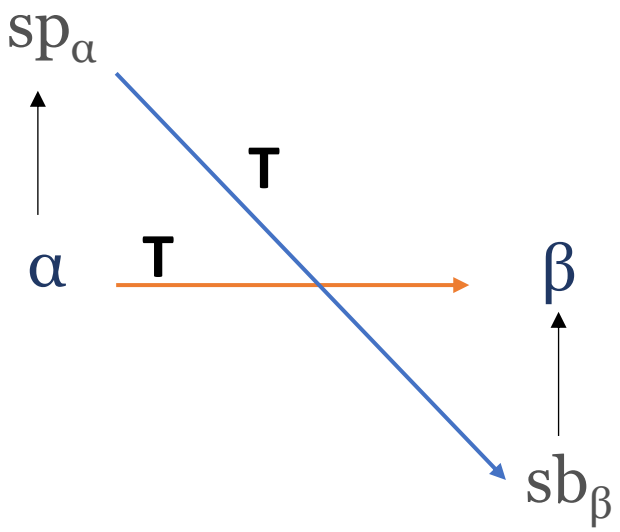
Completing - finding correct stronger axioms

Replace correct $\alpha \sqsubseteq \beta$

with correct $sp_\alpha \sqsubseteq sb_\beta$

Completing leads to more complete ontologies.

Few algorithms exist.



Ontologies

LINKÖPINGS UNIVERSITET

Basic operations

- Debugging: find the wrong asserted axioms in the ontology
- Removing: remove the wrong asserted axioms from the ontology

Remove wrong

- Weakening: Find correct weakened axioms of the wrong axioms

(Mitigate the negative effect of removing wrong axioms)

- Completing: Find correct completed axioms of the weakened axioms

(Make the ontology more complete)

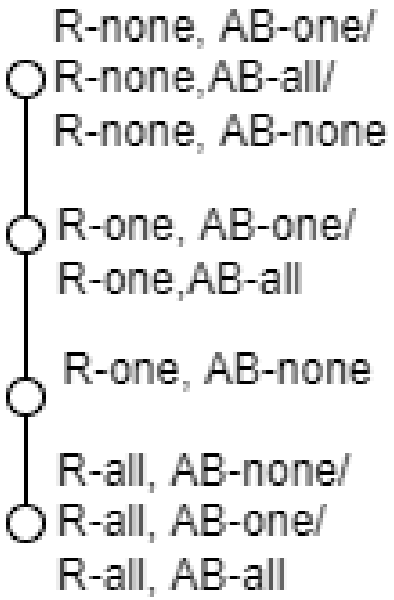
Add correct

Combinations of basic operations

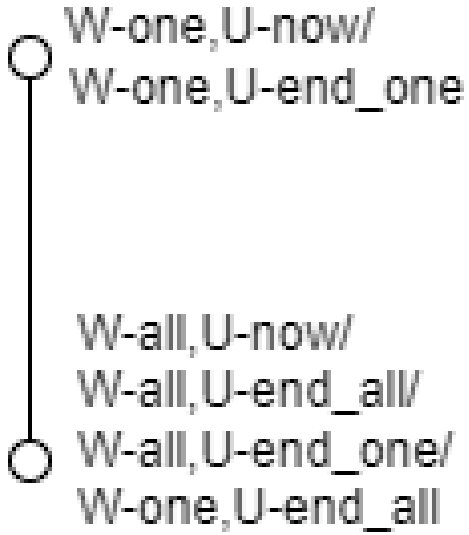
Choices regarding

- In which order to perform the operations
- Performing computations for all axioms at once or one at the time
- When to update the ontology (as soon as correct axioms are found or at the end)
- …

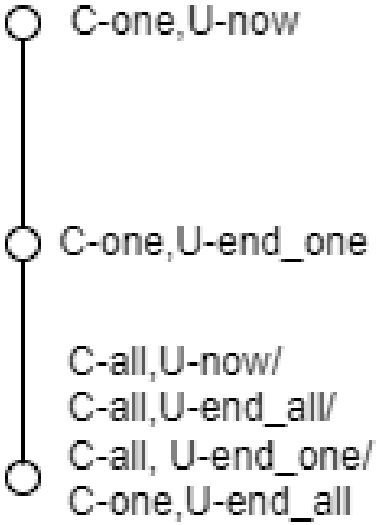
Combination operators



R: remove wrong axioms (one at a time/ all at once/none)
AB: add back wrong axioms (none/one/all)



W: weaken (one at a time/all at once)
U: update the ontology



C: complete (one at a time/all at once)
U: update the ontology

Using operators higher up in the diagrams leads to more complete ontologies and more validation work.

Li Y, Lambrix P, Repairing EL Ontologies Using Weakening and Completing, ESWC 2023.

Compare algorithms using the Hasse diagrams

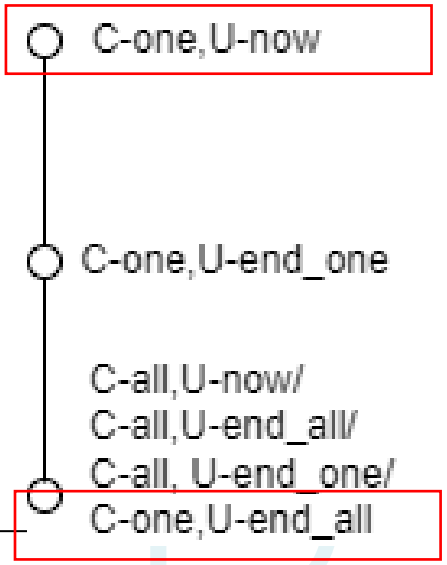
Alg 1: Weaken one at a time, remove all wrong, complete one at a time, then add completed axiom sets at end

C-one, U-end_all

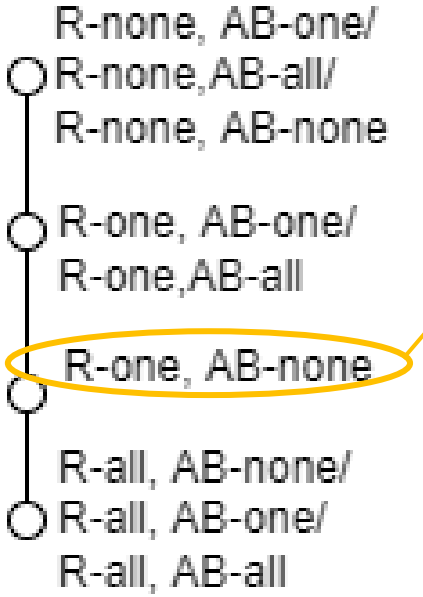
Alg 2: Weaken one at a time, remove all wrong, complete/add completed axiom sets one at a time

C-one, U-now

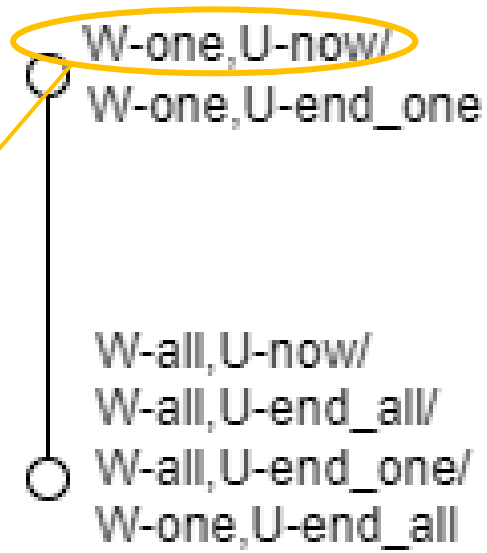
→ Alg 2 leads to more complete ontologies than Alg 1



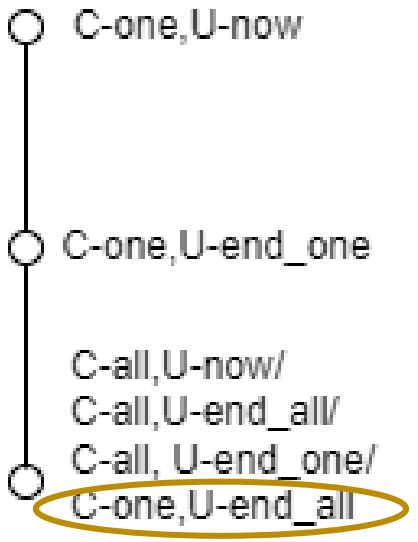
Compare algorithms using the Hasse diagrams



R: remove wrong axioms (one at a time/ all at once/none)
AB: add back wrong axioms (none/one/all)



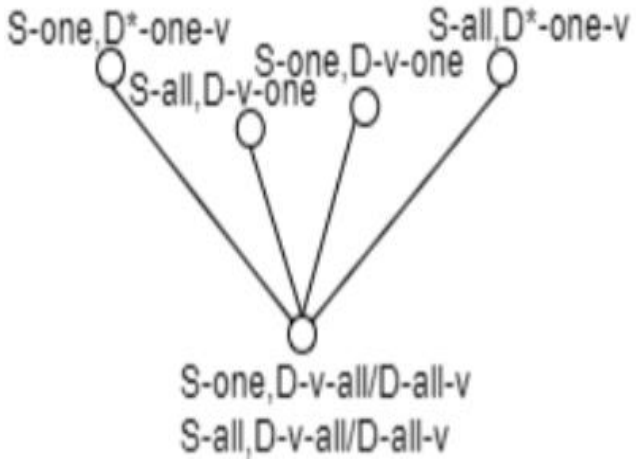
W: weaken (one at a time/all at once)
U: update the ontology



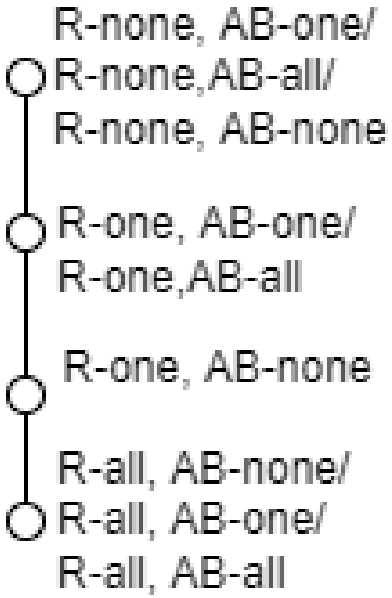
C: complete (one at a time/all at once)
U: update the ontology

Using operators higher up in the diagrams leads to more complete ontologies and more validation work.

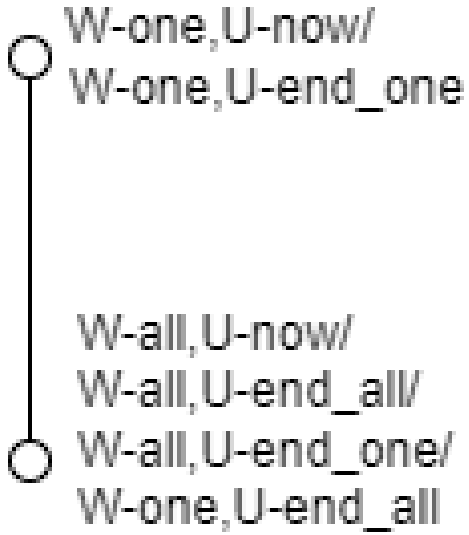
Compare algorithms using the Hasse diagrams



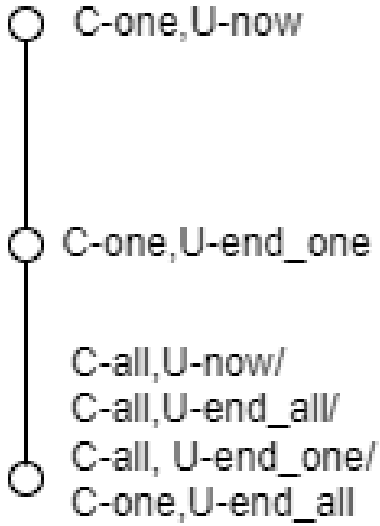
S: Compute the justifications for (one at a time/ all at once)
D: Validate the axioms (all asserted axioms/one hitting set)



R: remove wrong axioms (one at a time/ all at once/none)
AB: add back wrong axioms (none/one/all)



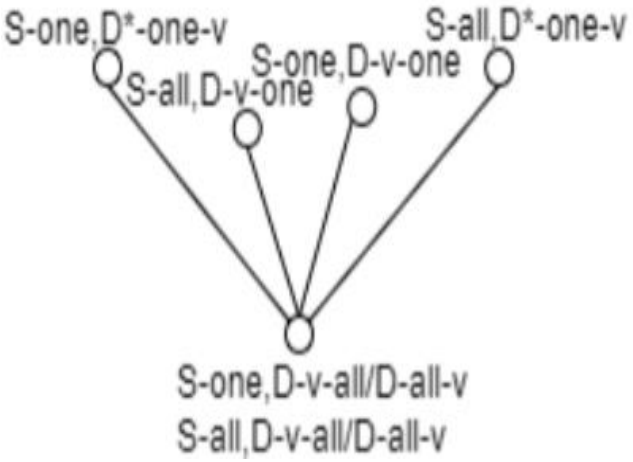
W: weaken (one at a time/all at once)
U: update the ontology

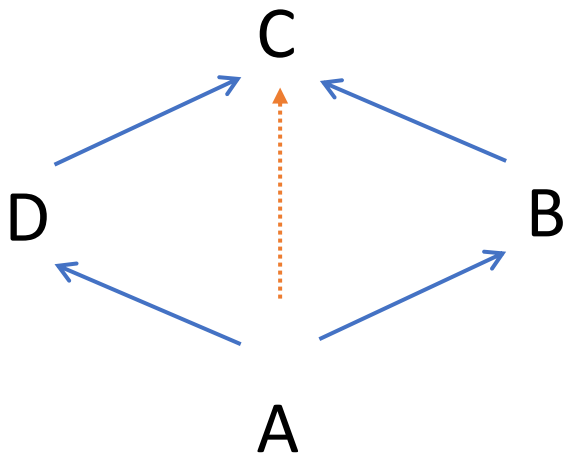


C: complete (one at a time/all at once)
U: update the ontology

Using operators higher up in the diagrams leads to more complete, more incorrect ontologies.
Higher up leads to more (less) validation work for weaking/completing (debugging).

Compare algorithms using the Hasse diagrams



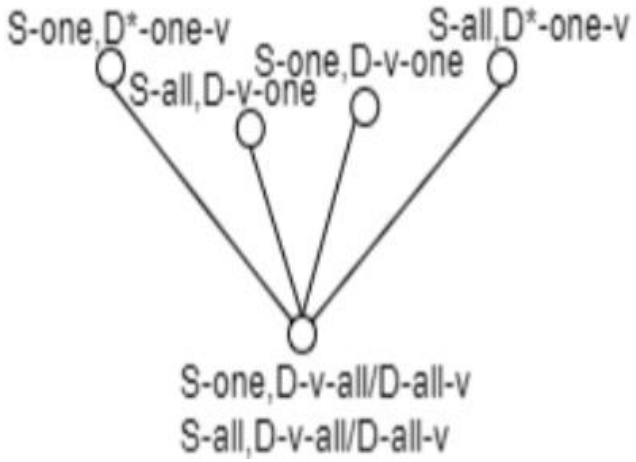


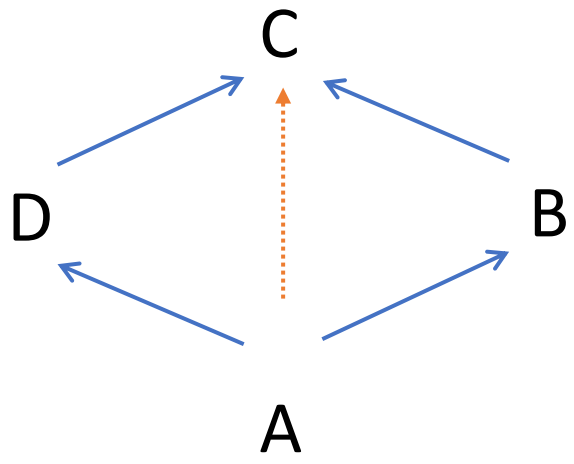
S: Compute the justifications for (one at a time/ all at once)
D: Validate the axioms (all asserted axioms/one hitting set)

Justifications(A ⊑ C)={ {A ⊑ B, B ⊑ C}, [A ⊑ D, D ⊑ C} }

Using operators higher up in the diagrams leads to more complete, more incorrect ontologies.
Higher up leads to more (less) validation work for weaking/completing (debugging).

Compare algorithms using the Hasse diagrams



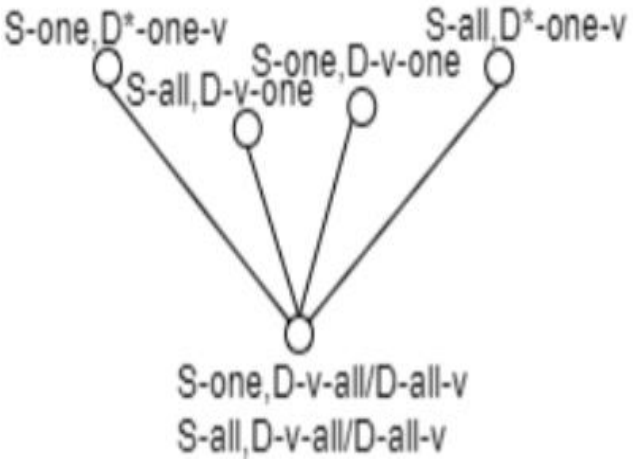


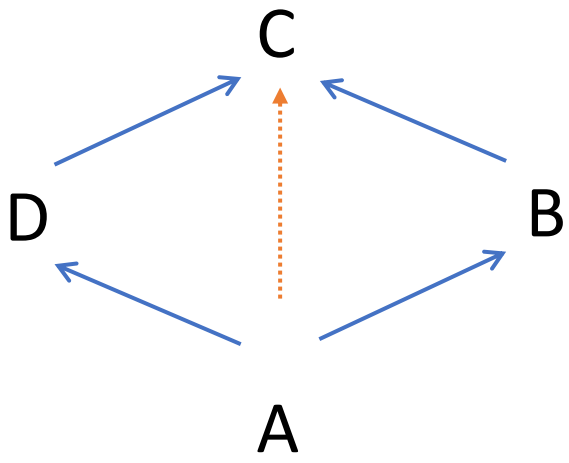
S: Compute the justifications for (one at a time/ all at once)
D: Validate the axioms (all asserted axioms/one hitting set)

Justifications($A \sqsubseteq C$)={ {$A \sqsubseteq B$, $B \sqsubseteq C$}, [$A \sqsubseteq D$, $D \sqsubseteq C$} }

Using operators higher up in the diagrams leads to more complete, more incorrect ontologies.
Higher up leads to more (less) validation work for weaking/completing (debugging).

Compare algorithms using the Hasse diagrams





Justifications(A ⊑ C)={ {A ⊑ B, B ⊑ C}, [A ⊑ D, D ⊑ C} }

S: Compute the justifications for (one at a time/ all at once)
D: Validate the axioms (all asserted axioms/one hitting set)

Using operators higher up in the diagrams leads to more complete, more incorrect ontologies.
Higher up leads to more (less) validation work for weaking/completing (debugging).

Ontology networks

LINKÖPINGS UNIVERSITET

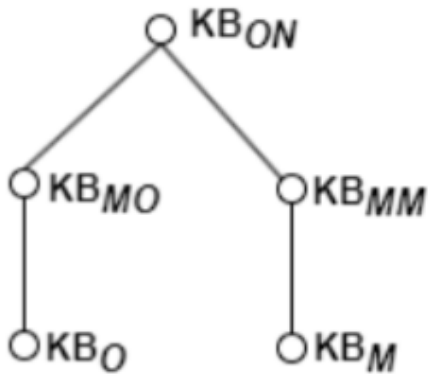
Ontology vs. Ontology network

There are also choices regarding the autonomy level of the ontologies and alignments in the ontology network, which reflects the policies of the ontology and alignment owners regarding updating and computing for their ontologies and alignments.

- O (ontology) / M(mappings)
- MO (materialized ontology) / MM(materialized mappings)
- ON (ontology network)

Ontology network

During the repairing process different levels of autonomy can be used at different stages:



Choices regarding computation

More complete

Choices regarding final answer

Using as much knowledge as possible may lead to more complete networks, but also more validation work.

Extreme case 1

➢Ontology is completely autonomous

- The set of wrong axioms W contains only axioms in the ontology (O).
- Only the axioms within the ontology can be used for the computation of repairs.

  ⟶ KB_O
- Solutions only include axioms in the ontology.

  ⟶ AS_O

Extreme case 2

➢ON (ontology network) --- ontologies and alignments as integral parts of the network

- The set of wrong axioms W contain ontology axioms and mappings.
- The whole network is used for the computation of repairs.

  ⟶ KB_{ON}

- Solutions contain ontology axioms and mappings.

  ⟶ AS_{ON}

Example

Asserted axioms in Ontologies O_1 and O_2

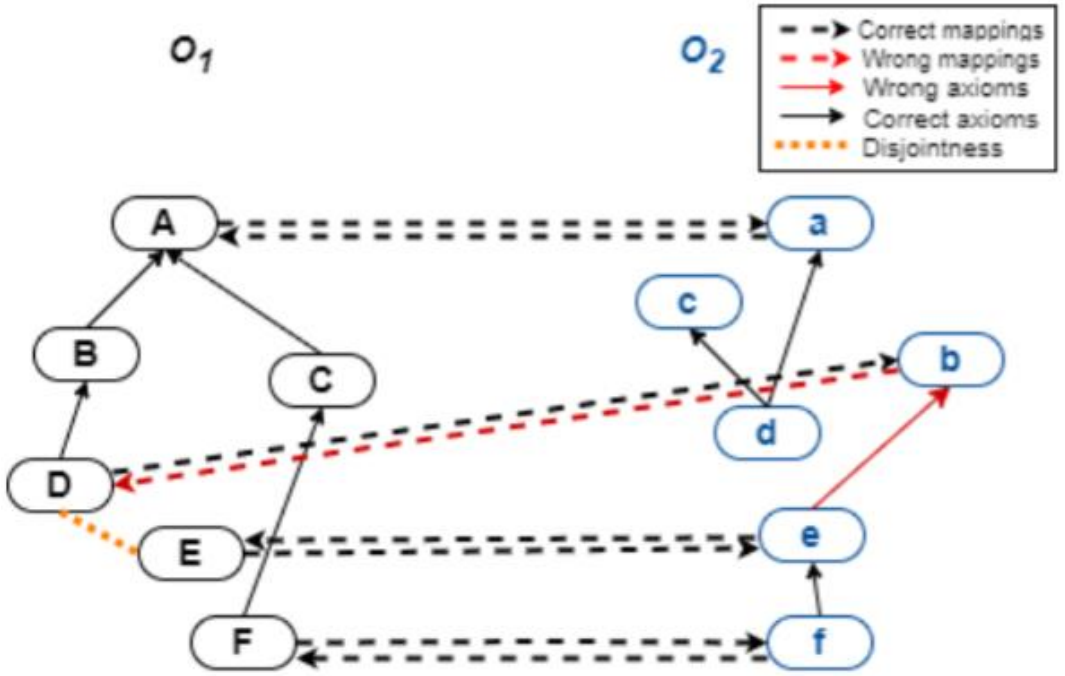
O_1: B ⊑ A, D ⊑B, C ⊑ A, D ⊑ B, F ⊑ C

O_2: d ⊑ c, d ⊑ a, e ⊑ b, f ⊑ e

Mappings between O_1 and O_2

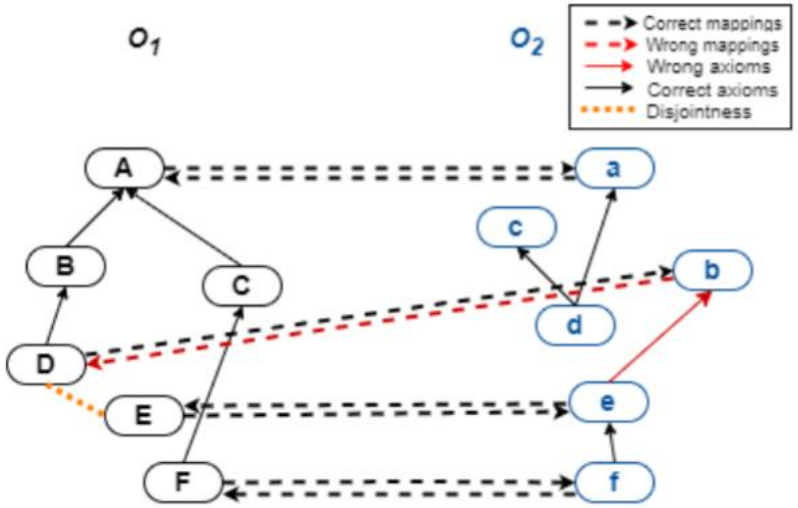
a ⊑ A, D ⊑ b, E ⊑ e, F ⊑ f

A ⊑ a, b ⊑ D, e ⊑ E, f ⊑ F



We have used 'ON' in the debugging step where we validate all axioms in the justifications. This results in a wrong axiom e ⊑ b and a wrong mapping b ⊑ D.

Weakening

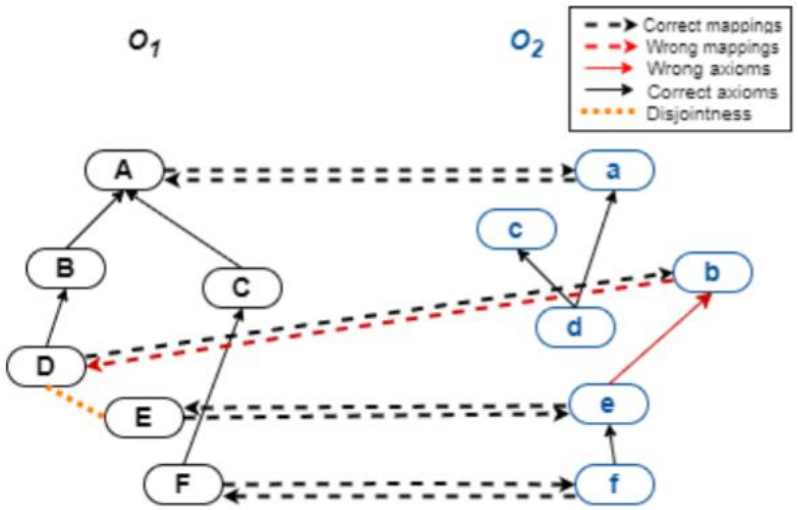


2, 4, 20 candidates for ‘O’, ’MO’ and ’ON’

Background knowledge base	KB_O $e \sqsubseteq b$	KB_{MO} $e \sqsubseteq b$	KB_{ON} $e \sqsubseteq b$		KB_M $b \sqsubseteq D$	KB_{MM} $b \sqsubseteq D$	KB_{ON} $b \sqsubseteq D$
$\lvert Sub(\alpha,\mathcal{T})\rvert$	2	2	4		2	6	6
$\lvert Sup(\beta,\mathcal{T})\rvert$	1	2	5		2	5	5
Weakened	$f \sqsubseteq b$	$f \sqsubseteq b$, $e \sqsubseteq a$	$f \sqsubseteq b$, $e \sqsubseteq a$, $E \sqsubseteq A$			$b \sqsubseteq B$	$b \sqsubseteq B$, $e \sqsubseteq a$

’ON’ leads to the most complete network

Completing



Background knowledge base	KB_O e ⊑ b	KB_{MO} e ⊑ b	KB_{ON} e ⊑ b		KB_M b ⊑ D	KB_{MM} b ⊑ D	KB_{ON} b ⊑ D
$\|Sub(\alpha,\mathcal{T})\|$ $\|Sup(\beta,\mathcal{T})\|$	2 1	2 2	4 5		2 2	6 5	6 5
Weakened	f ⊑ b	f ⊑ b, e ⊑ a	f ⊑ b, e ⊑ a, E ⊑ A			b ⊑ B	b ⊑ B, e ⊑ a
$\|Sup(\alpha,\mathcal{T})\|$ $\|Sub(\beta,\mathcal{T})\|$	2 1	4 3 3 5	10 7 7 6 11 11			5 7	5 7 7 11
Completed	f ⊑ b	f ⊑ b, e ⊑ d	f ⊑ b, e ⊑ d, E ⊑ C			b ⊑ B, B ⊑ b	b ⊑ B, B ⊑ b

’ON’ leads to the most complete network

The network is repaired by removing the wrong axioms and adding the completed axioms.

Conclusion

Conclusion

- Interactive approach to mitigate the negative effects of removing unwanted axioms from an ontology network.
- Combination operators reflecting choices and policies.
- Trade-offs for different combination strategies involving correctness, completeness and validation work.
- Implemented system.
- Our framework provides a blueprint for extending previous work and systems.