Concept refinements and their applications in Description Logics and knowledge engineering

Nicolas Troquard

Free University of Bozen-Bolzano

with Roberto Confalonieri, Pietro Galliani, Maria Hedblom, Oliver Kutz, Rafael Peñaloza, Daniele Porello, Guendalina Righetti

KRDB Summer Online Seminars 2020 — 7 August

A bad ontology

1. $\mathsf{RaiseWages}(Switzerland)$
2. $\mathsf{TaxHighIncomes}(Sweden)$
3. $\mathsf{RaiseWelfare}(Switzerland)$
4. $\mathsf{RaiseWages}(Sweden)$
5. $\mathsf{TaxHighIncomes}(Switzerland)$
6. $\mathsf{RaiseWelfare}(France)$
7. $\mathsf{RaiseWelfare} \sqsubseteq \mathsf{LeftPolicy}$
8. $\mathsf{RaiseWelfare} \sqsubseteq \neg\mathsf{RaiseWages}$
9. $\mathsf{TaxHighIncomes} \sqsubseteq \mathsf{LeftPolicy}$
10. $\mathsf{LeftPolicy} \sqsubseteq \mathsf{RaiseWages} \sqcup \mathsf{RaiseWelfare} \sqcup \mathsf{TaxHighIncomes}$
11. $\mathsf{RaiseWages} \sqsubseteq \mathsf{LeftPolicy}$

voting

Sources of inconsistency:

- single-source error: after the addition of some axioms
- multi-source incompatibility:
  - during an automated construction (e.g., text mining, Yago, DBPedia, Open Street Map, ...)
  - ...
  - during a collaborative process aggregating opinions (e.g., multi-issue voting)

Example: an inconsistent ontology

Adapted from
[Lam, Sleeman, Pan, Vasconcelos: A Fine-Grained Approach to Resolving Unsatisfiable Ontologies. 2008].

1. $A(a)$
2. $A \equiv C \sqcap \forall R.B \sqcap D$
3. $C \equiv \exists R.\neg B \sqcap B$
4. $A \sqsubseteq G$
5. $F \sqsubseteq A$

Example: repair by axiom removing

Adapted from
[Lam, Sleeman, Pan, Vasconcelos: A Fine-Grained Approach to Resolving Unsatisfiable Ontologies. 2008].

Inconsistent $\mathcal{O}$:

1. $A(a)$
2. $A \sqsubseteq C \sqcap \forall R.B \sqcap D$
3. $C \sqcap \forall R.B \sqcap D \sqsubseteq A$
4. $C \sqsubseteq \exists R.\neg B \sqcap B$
5. $\exists R.\neg B \sqcap B \sqsubseteq C$
6. $A \sqsubseteq G$
7. $F \sqsubseteq A$

Solution: remove one axiom, e.g., axiom 2, 4, or 1.

Example: repair by removing parts of axioms

Inconsistent $\mathcal{O}$:

1. $A(a)$
2. $A \sqsubseteq C \sqcap \forall R.B \sqcap D$
3. $C \sqcap \forall R.B \sqcap D \sqsubseteq A$
4. $C \sqsubseteq \exists R.\neg B \sqcap B$
5. $\exists R.\neg B \sqcap B \sqsubseteq C$
6. $A \sqsubseteq G$
7. $F \sqsubseteq A$

Solution: remove parts of an axiom, e.g., conjunct C in axiom 2.

Example: repair by normalization and removing axioms

Solution (part 1): normalize $\mathcal{O}$:

1. $A(a)$
2. $A \sqsubseteq C$
3. $A \sqsubseteq \forall R.B$
4. $A \sqsubseteq D$
5. $C \sqcap \forall R.B \sqcap D \sqsubseteq A$
6. $C \sqsubseteq \exists R.\neg B$
7. $C \sqsubseteq B$
8. $\exists R.\neg B \sqcap B \sqsubseteq C$
9. $A \sqsubseteq G$
10. $F \sqsubseteq A$

Solution (part 2): remove, e.g., axiom 2, or 6.

Example: repair by axiom weakening

Inconsistent $\mathcal{O}$:

1. $A(a)$
2. $A \sqsubseteq C \sqcap \forall R.B \sqcap D$
3. $C \sqcap \forall R.B \sqcap D \sqsubseteq A$
4. $C \sqsubseteq \exists R.\neg B \sqcap B$
5. $\exists R.\neg B \sqcap B \sqsubseteq C$
6. $C \sqsubseteq G$
7. $F \sqsubseteq C$

Solution: replace, e.g., axiom 2, into $A \sqsubseteq \mathsf{G} \sqcap \forall R.B \sqcap D$; or axiom 4 into $\mathsf{F} \sqsubseteq \exists R.\neg B \sqcap B$; or axiom 1 into $\mathsf{G}(a)$.

Repairing Ontologies via Axiom Weakening

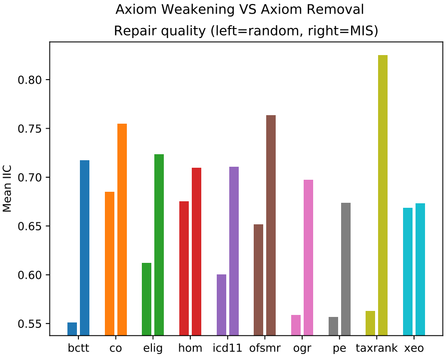
(Troquard, Confalonieri, Galliani, Peñaloza, Porello, Kutz — [AAAI 2018])

There is an increasing demand for methods to finely repair inconsistent ontologies.

Most existing approaches are based on coarsely removing a few axioms from the ontology to regain consistency.

Contributions:

- novel concept refinement operators.
- theoretical computational complexity.
- axiom weakening.
- experimental evaluation of repairing by axiom removal vs. by axiom weakening.



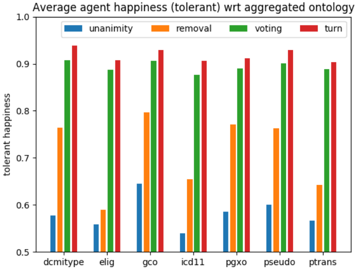
Two Approaches to Ontology Aggregation Based on Axiom Weakening

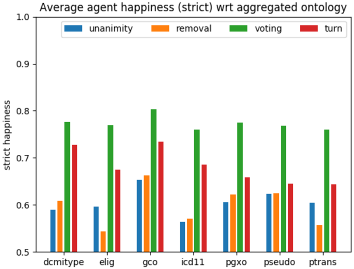
(Porello, Troquard, Peñaloza, Confalonieri, Galliani, Kutz — [IJCAI 2018])

Aggregating opinions about a domain is a traditional topic of social choice theory.

Contributions:

- new methods of aggregation.
- experimental evaluation vs. unanimity and axiom removal.





Towards even more irresistible axiom weakening

(Confalonieri, Galliani, Kutz, Porello, Righetti, Troquard — [DL 2020])

The work of [AAAI 2018] is limited to $\mathcal{EL}$ and $\mathcal{ALC}$ ontologies. The OWL 2 Web Ontology Language is based on $\mathcal{SROIQ}$.

Contributions:

- extension of the refinement operators to $\mathcal{SROIQ}$ concepts and roles.
- computational complexity.
- almost-sure termination of repair algorithms.

Outline

Description Logics

$\mathcal{ALC}$ concepts over N_C and N_R:

$$C \quad ::= \quad A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall R.C \mid \exists R.C \ ,$$

where A is a concept name in N_C and R is a role name in N_R.

$\mathcal{EL}$ is the restriction of $\mathcal{ALC}$ allowing only conjunctions and existential restrictions.

($\mathcal{SROIQ}$ also has nominals, inverse roles, qualified role restrictions, ...)

The set $\mathcal{L}(\mathcal{DL}, N_c, N_R)$ collects all the concepts of $\mathcal{DL}$ over N_C and N_R.

A TBox is a finite set of concept inclusions of the form $C \sqsubseteq D$ where C and D are concepts.

$$\mathsf{RaiseWages} \sqcup \mathsf{TaxHighIncomes} \sqsubseteq \neg\mathsf{RightPolicy}$$

An ABox is a finite set of formulas of the form $C(a)$ and $R(a, b)$, where a and b are individual names in N_I.

$$\mathsf{RaiseWages}(Switzerland), \quad \textit{ShareBorder}(Switzerland, France)$$

($\mathcal{SROIQ}$ also has an RBox, a finite set of role inclusions, and role constraints.)

Semantics

An interpretation is a tuple $I = (\Delta^I, \cdot^I)$, where Δ^I is a non-empty *domain*, and $\cdot^I$ is a function:

- $a \in N_I$, $a^I \in \Delta^I$.
- $C \in N_C$, $C^I \subseteq \Delta^I$.
- $R \in N_R$, $R^I \subseteq \Delta^I \times \Delta^I$.

It is extended to all $\mathcal{ALC}$ concepts as follows:

- $\top^I = \Delta^I$; $\quad \bot^I = \emptyset$; $\quad (\neg C)^I = \Delta^I \setminus C^I$
- $(C \sqcap D)^I = C^I \cap D^I$; $\quad (\exists R.C)^I = \{x \in \Delta^I \mid (x, y) \in R^I,$ and $y \in C^I\}$

The truth value of axioms is:

- $I \models C \sqsubseteq D$ iff $C^I \subseteq D^I$; $\quad I \models C(a)$ iff $a \in C^I$; $\quad I \models R(a, b)$ iff $(a^I, b^I) \in R^I$.

The interpretation I is a model of the ontology $\mathcal{O}$ if it satisfies all the axioms in $\mathcal{O}$. An ontology is consistent if it has a model.

Given two concepts C and D, we say that C is subsumed by D w.r.t. the ontology $\mathcal{O}$ ($C \sqsubseteq_{\mathcal{O}} D$) if $I \models C \sqsubseteq D$ for every model I of $\mathcal{O}$. We write $C \equiv_{\mathcal{O}} D$ when $C \sqsubseteq_{\mathcal{O}} D$ and $D \sqsubseteq_{\mathcal{O}} C$.

C is "strictly" subsumed by D w.r.t. $\mathcal{O}$ ($C \sqsubset_{\mathcal{O}} D$) if $C \sqsubseteq_{\mathcal{O}} D$ and $C \not\equiv_{\mathcal{O}} D$.

Refinement operators

A generalisation refinement operator (wrt. an ontology $\mathcal{O}$) is a function:

$$\gamma_{\mathcal{O}}(C) \subseteq \{C' \in \mathcal{L}(\mathcal{DL}, N_c, N_R) \mid C \sqsubseteq_{\mathcal{O}} C'\} \ .$$

A specialisation refinement operator is a function:

$$\rho_{\mathcal{O}}(C) \subseteq \{C' \in \mathcal{L}(\mathcal{DL}, N_c, N_R) \mid C' \sqsubseteq_{\mathcal{O}} C\} \ .$$

Expected properties:

1. **generalisation** if $D \in \gamma_{\mathcal{O}}(C)$ then $C \sqsubseteq_{\mathcal{O}} D$
2. **specialisation** if $D \in \rho_{\mathcal{O}}(C)$ then $D \sqsubseteq_{\mathcal{O}} C$
3. **trivial generalisability** $\top \in \gamma^*_{\mathcal{O}}(C)$
4. **falsehood specialisability** $\bot \in \rho^*_{\mathcal{O}}(C)$
5. **generalisation finiteness** $\gamma_{\mathcal{O}}(C)$ is finite
6. **specialisation finiteness** $\rho_{\mathcal{O}}(C)$ is finite.

In [AAAI'18], we propose refinement operators with these properties.

details

Complexity

The refinement operators proposed in the paper are "efficient".

Definition

Given an ontology $\mathcal{O}$ and concepts C, D, the problems $\gamma_{\mathcal{O}}$-membership and $\rho_{\mathcal{O}}$-membership ask whether $D \in \gamma_{\mathcal{O}}(C)$ and $D \in \rho_{\mathcal{O}}(C)$, respectively.

	$\mathcal{ALC}$	$\mathcal{EL}$	$\mathcal{SROIQ}$ [DL 2020]
$\gamma_{\mathcal{O}}$-membership	ExpTime-complete	PTime-complete	N2ExpTime-complete
$\rho_{\mathcal{O}}$-membership	ExpTime-complete	PTime-complete	N2ExpTime-complete

details

Hence: the problems are no harder than concept subsumption.

Axiom Weakening

Definition (Axiom weakening)

Given a concept inclusion $C \sqsubseteq D$, the set of (least) weakenings of $C \sqsubseteq D$ w.r.t. $\mathcal{O}$, denoted by $g_{\mathcal{O}}(C \sqsubseteq D)$ is the set of all axioms $C' \sqsubseteq D'$ such that

$$C' \in \rho_{\mathcal{O}}(C) \text{ and } D' = D$$

or

$$C' = C \text{ and } D' \in \gamma_{\mathcal{O}}(D) \quad .$$

Given an assertional axiom $C(a)$, the set of (least) weakenings of $C(a)$, denoted $g_{\mathcal{O}}(C(a))$ is the set of all axioms $C'(a)$ such that

$$C' \in \gamma_{\mathcal{O}}(C).$$

Lemma

For every axiom φ, if $\varphi' \in g_{\mathcal{O}}(\varphi)$, then $\varphi \models_{\mathcal{O}} \varphi'$.

Special care with infinite chains of refinements

In general, refinement operators are reflexive: $C \in \gamma_{\mathcal{O}}(C)$, and $C \in \rho_{\mathcal{O}}(C)$.

There may also be infinite chains of strict refinements.

Example: Reference ontology $\mathcal{O}$:

- $A \sqsubseteq \exists R.A$

We have:

$$\exists R.A \in \gamma_{\mathcal{O}}(A)$$

So:

- $\exists R.A \in \gamma_{\mathcal{O}}(A)$
- $\exists R.\exists R.A \in \gamma_{\mathcal{O}}(\exists R.A)$
- $\exists R.\exists R.\exists R.A \in \gamma_{\mathcal{O}}(\exists R.\exists R.A)$
- $\ldots$
- $(\exists R.)^k A \in \gamma_{\mathcal{O}}^k(A)$

Almost-sure reachability of $\top$ and $\bot$

See [DL 2020].

A simple procedure to reach $\top$: Input concept C; Iteratively generalize it until $\top$ is reached.

- while $C \neq \top$:
- uniformly at random choose $C' \in \gamma_{\mathcal{O}}(C)$
- $C = C'$

Non-termination.
Almost-sure termination.

Theorem

Let $\mathcal{O}$ be a $\mathcal{DL} \in \{\mathcal{EL}, \mathcal{ALC}, \mathcal{SROIQ}\}$ ontology, let C be a concept, and let $(C_i)_{i \in \mathbb{N}}$ be a sequence of concepts such that $C_0 = C$ and each C_{i+1} is chosen uniformly at random in $\gamma_{\mathcal{O}}(C_i)$. Then, with probability 1, there exists some $i \in \mathbb{N}$ such that $C_i = \top$.

- The rate of growth in size of generalizations and of the set of generalizations is "small".
  - if $C' \in \gamma_{\mathcal{O}}(C) \cup \rho_{\mathcal{O}}(C)$, then $|C'|$ is linear in $|C| + |\mathcal{O}|$.
  - $\mathbf{card}(\gamma_{\mathcal{O}}(C))$ and $\mathbf{card}(\rho_{\mathcal{O}}(C))$ is linear in $|\mathcal{O}|$.

Outline

Needs for a reference ontology

Reference ontology: any ontology $\mathcal{O}^{\mathsf{ref}}$ with which one can make "useful" inferences.

The purpose of $\mathcal{O}^{\mathsf{ref}}$ is to make "useful" generalisations of a concept C with:

$$\gamma_{\mathcal{O}^{\mathsf{ref}}}(C) \quad ,$$

"useful" specialisations of a concept C with:

$$\rho_{\mathcal{O}^{\mathsf{ref}}}(C) \quad ,$$

and sensible weakening of axioms with

$$g_{\mathcal{O}^{\mathsf{ref}}}(\phi) \quad .$$

ChooseAxiom

ChooseAxiom(O) is critical.

For the experimental evaluation, we consider:

- Random: take an axiom at random;
- MIS: take an axiom occurring the most often in the set of minimally inconsistent sets at random.

In practice:

- ask the user to choose an axiom, possibly in a MIS.

WeakenAxiom

WeakenAxiom(ϕ, $\mathcal{O}^{\mathsf{ref}}$).

For evaluation:

- choose an axiom uniformly at random in $g_{\mathcal{O}^{\mathsf{ref}}}(\phi)$.

In practice:

- ask the user to choose an axiom in $g_{\mathcal{O}^{\mathsf{ref}}}(\phi)$.

Repair strategies

Algorithm 1 RepairOntologyRemove(O)

---

while O is inconsistent **do**
  BadAx $\leftarrow$ ChooseAxiom(O)
  $O \leftarrow O \setminus \{\text{BadAx}\}$
end while
return O

Algorithm 2 RepairOntologyWeaken(O)

$\mathcal{O}^{\text{ref}} \leftarrow$ MaximallyConsistent(O)
while O is inconsistent **do**
  BadAx $\leftarrow$ ChooseAxiom(O)
  WeakerAx $\leftarrow$ WeakenAxiom(BadAx, $\mathcal{O}^{\text{ref}}$)
  $O \leftarrow O \setminus \{\text{BadAx}\} \cup \{\text{WeakerAx}\}$
end while
return O

Comparing two repairs: relative quality

When is one of two consistent repairs O_1 and O_2 of an inconsistent ontology O preferable to the other?

Here, we compare their information content, in the inferred class hierarchies.

$$\mathsf{Inf}(O_i) = \{A \sqsubseteq B \mid A, B \in N_C \cap \mathsf{sub}(O_i), O_i \models A \sqsubseteq B\} \ .$$

Measure to compare the inferable information content of two ontologies:

Definition

Let O_1 and O_2 be two consistent ontologies. If $\mathsf{Inf}(O_1) \neq \mathsf{Inf}(O_2)$, we define the inferable information content $\mathsf{IIC}(O_1, O_2)$ of O_1 w.r.t. O_2 as $\mathsf{IIC}(O_1, O_2) =$

$$\frac{\mathbf{card}(\mathsf{Inf}(O_1) \setminus \mathsf{Inf}(O_2))}{\mathbf{card}(\mathsf{Inf}(O_1) \setminus \mathsf{Inf}(O_2)) + \mathbf{card}(\mathsf{Inf}(O_2) \setminus \mathsf{Inf}(O_1))} \ ;$$

if instead $\mathsf{Inf}(O_1) = \mathsf{Inf}(O_2)$, we set $\mathsf{IIC}(O_1, O_2) = 0.5$.

Properties of IIC and interpretation

1. $\mathsf{IIC}(O_1, O_2) \in [0, 1]$;
2. $\mathsf{IIC}(O_1, O_2) = 1 - \mathsf{IIC}(O_2, O_1)$;
3. $\mathsf{IIC}(O_1, O_2) = 0.5$ if and only if $\mathbf{card}(\mathsf{Inf}(O_1)) = \mathbf{card}(\mathsf{Inf}(O_2))$;
4. $\mathsf{IIC}(O_1, O_2) = 1$ if and only if $\mathsf{Inf}(O_2) \subset \mathsf{Inf}(O_1)$;
5. $\mathsf{IIC}(O_1, O_2) > 0.5$ if and only if $\mathbf{card}(\mathsf{Inf}(O_1) \setminus \mathsf{Inf}(O_2)) > \mathbf{card}(\mathsf{Inf}(O_2) \setminus \mathsf{Inf}(O_1))$.

Applied to two repairs O_1 and O_2 of the same inconsistent ontology, we interpret

$$\mathsf{IIC}(O_1, O_2) > 0.5$$

as O_1 being a 'better' repair than O_2.

Results on ontologies from the NCBO BioPortal

Results over 100 repairing runs on 10 ontologies from the BioPortal made inconsistent.

	Random	MIS
bctt	0.55 (0.35)	**0.72 (0.36)**
co-wheat	**0.69 (0.29)**	**0.76 (0.31)**
elig	**0.61 (0.30)**	**0.72 (0.27)**
hom	**0.68 (0.26)**	**0.71 (0.31)**
icd11	0.60 (0.30)	**0.71 (0.40)**
ofsmr	**0.65 (0.31)**	**0.76 (0.29)**
ogr	0.56 (0.32)	**0.70 (0.35)**
pe	0.56 (0.33)	**0.67 (0.41)**
taxrank	0.56 (0.31)	**0.82 (0.36)**
xeo	**0.67 (0.29)**	**0.67 (0.34)**

Table: Mean and standard deviation (in parentheses) of IIC between RepairOntologyWeaken and RepairOntologyRemove, both when choosing axioms at random (left column) and by sampling minimally inconsistent sets (right). Bolded values are significant $(p < 0.05)$ with respect to both Wilcoxon and T-test with Holm-Bonferroni correction; non-bolded values were not significant for either.

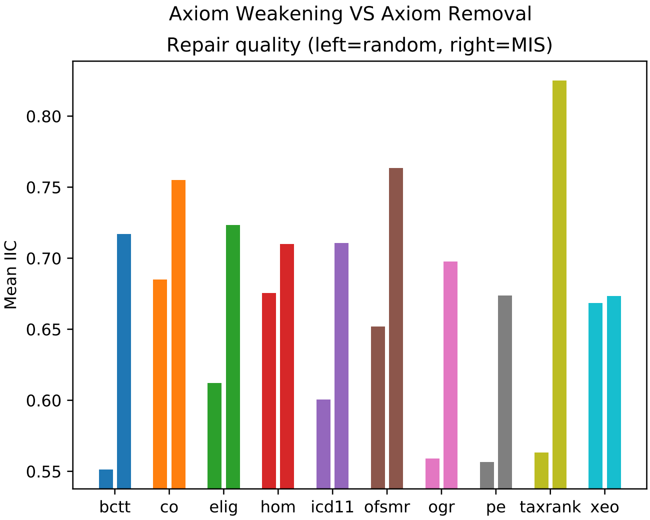


Figure: Comparing weakening-based ontology repair with removal-based ontology repair. Mean IIC of weakening-based against removal-based repair for each ontology, when choosing axioms at random (left) or by sampling minimally inconsistent sets (right).

Implementations

`https://bitbucket.org/troquard/ontologyutils/` (src/main/java/www/ontologyutils/apps)

- AppAutomatedRepairWeakening
- AppInteractiveRepair
- AppInteractiveReferenceOntologyAndRepair

Outline

Social choice: discursive dilemma

- P: the defendant did a certain action;
- Q: the defendant had a contractual obligation not to do that action;
- C: the defendant is liable.

	P?	Q?	$C \equiv P \wedge Q$?	C?
Juror 1	yes	yes	yes	yes
Juror 2	no	yes	yes	no
Juror 3	yes	no	yes	no
Majority	yes	yes	yes	no

Each Juror makes a consistent judgement; the Majority is inconsistent.

Repairing collective ontologies

agenda

A discursive dilemma.

	1	2	3	4	5	6	7	8	9	10	11
Expert 1	1	0	1	0	1	0	0	0	0	0	0
Expert 2	0	0	1	0	0	0	1	1	1	1	1
Expert 3	1	1	0	0	1	0	1	1	0	0	1
Majority	1	0	1	0	1	0	1	1	0	0	1

$\{1, 3, 8\}$ is an inconsistent set of axioms:

- RaiseWages$(Switzerland)$
- RaiseWelfare$(Switzerland)$
- RaiseWelfare $\sqsubseteq \neg$RaiseWages

A principled reference ontology

Idea:

1. Preference $<_i$ over the set of axioms $\mathcal{O}$ for every agent i.
2. Lexicographic preference over the subsets of axioms.
3. Collective preference.
4. $<$-optimal MCS: most collectively preferred axioms are added iteratively, as long as they cause no inconsistency.
5. Reference ontology as $<$-optimal MCS.

Proposition

If ontology consistency is in the complexity class $\mathfrak{C}$, then the problem of finding the $<$-optimal MCS is in the class $\mathrm{NP}^{\mathfrak{C}}$.

A majority voting procedure

Parameters:

- $\mathcal{O}$ is an (inconsistent) ontology.
- each O_i is a consistent subset of $\mathcal{O}$.
- each $<_i$ is a preference over $\mathcal{O}$.

Algorithm 3 VoteBasedCollectiveOntology($\mathcal{O}, (<_i)_i, (O_i)_i$)

$\mathcal{O}^{\text{ref}} \leftarrow$ ReferenceOntology($\mathcal{O}, (<_i)_i$)
$R \leftarrow$ Majority($(O_i)_i$)
while R is inconsistent **do**
    BadAx $\leftarrow$ ChooseAxiom(R)
    WeakerAx $\leftarrow$ WeakenAxiom(BadAx, $\mathcal{O}^{\text{ref}}$)
    $R \leftarrow R \backslash \{\text{BadAx}\} \cup \{\text{WeakerAx}\}$
end while
return R

A turn-based procedure

Algorithm 4 TurnBasedCollectiveOntology($\mathcal{O}, (<_i)_i, (O_i)_i$)

```
𝒪^ref ← ReferenceOntology(𝒪, (<_i)_i)
R ← ∅
ConsideredAxioms ← ∅
FinishedAgents ← ∅
Agent ← 1
while FinishedAgents ≠ Agents do
    if O_Agent ⊆ ConsideredAxioms then
        FinishedAgents ← FinishedAgents ∪{ Agent }
    else
        Ax ← FavoriteNonConsideredAxiom(<_Agent, O_Agent)
        ConsideredAxioms ← ConsideredAxioms ∪{Ax}
        while R ∪ {Ax} is inconsistent do
            Ax ← WeakenAxiom(Ax, 𝒪^ref)
        end while
        R ← R ∪ {Ax}
    end if
    Agent ← (Agent (mod # Agents )) + 1
end while
return R
```

Agents' happiness with an ontology

- **Tolerant Happiness:** an agent is tolerantly happy when their chosen axioms are in (or follow from) the collective repaired ontology, regardless of whether this ontology also entails further statements.
- **Strict Happiness:** an agent is strictly happy when their chosen axioms are in (or follow from) the collective ontology and everything that follows from the collective ontology also follows from their own chosen axioms.

Agents' happiness, formally

Definition

Let O_i be the set of axioms chosen by an agent i, and O be a consistent collective ontology, we define:

- the tolerant agent happiness $\textbf{TolH}(O|O_i)$ as

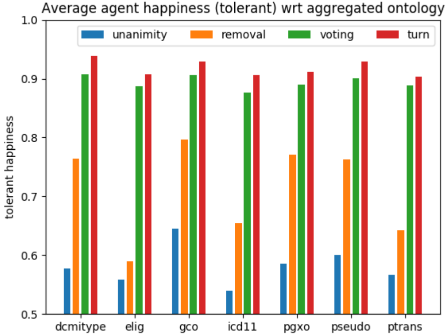
$$\frac{|\{\varphi \in O_i \cup \mathsf{Inf}(O_i) \text{ s.t. } O \models \varphi\}|}{|O_i \cup \mathsf{Inf}(O_i)|} ;$$

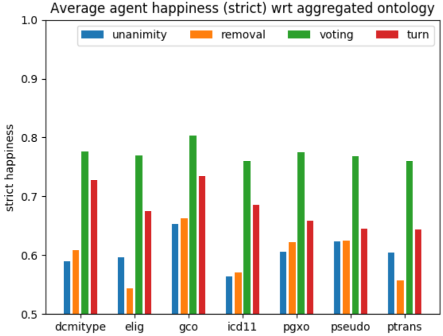
- the strict agent happiness $\textbf{StrH}(O|O_i)$ as

$$\frac{|\{\varphi \in O_i \cup \mathsf{Inf}(O_i) \cup O \cup \mathsf{Inf}(O) \text{ s.t. } O \models \varphi \text{ and } O_i \models \varphi\}|}{|O_i \cup \mathsf{Inf}(O_i) \cup O \cup \mathsf{Inf}(O)|} .$$

Results

We used 7 ontologies from BioPortal. Made the ontologies inconsistent 250 times through the addition of random axioms.
Each time and for each agent i, we randomly generated a preference order $<_i$, and chose the individual ontology O_i to be a consistent set of the best axioms of the agenda w.r.t. $<_i$.





Average agent happiness using our two methods (voting, turn) and two baselines (unanimity, removal).

`https://bitbucket.org/troquard/ontologyutils/` (src/main/java/www/ontologyutils/apps)

- AppTurnBasedMechanism

Outline

Conclusions

We proposed "efficient" refinement operators for $\mathcal{EL}$, $\mathcal{ALC}$, and recently for $\mathcal{SROIQ}$.

We proposed and implemented algorithms to repair inconsistent ontologies by weakening axioms.

Of course, weakening an axiom will preserve at least as much information as removing it would.

We quantified it.

We found that it is significant only if one can pinpoint reliably to the culprits for inconsistency.

We used axiom weakening in two approaches for ontology aggregation.

We found that the turn-based mechanism is preferable for tolerant-happiness; the voting mechanism is preferable for strict-happiness.

Further work

Implementation:

- Extend to $\mathcal{SROIQ}$.
- Plugin for Protégé.

Ontology evaluation criteria:

- What makes an ontology good?
- Here: quantity of information
- [Fox & Tenenbaum]: generality, efficiency, perspicuity, transformability, extensibility, granularity, scalability, competence.
- [Gruber]: clarity, coherence, extensibility, minimal encoding, minimal ontological commitment.
- ...

Need for inconsistent ontology benchmarks
(or "good" methods to render an existing ontology inconsistent):

- What makes an ontology badly hurt?
- Hard to repair ontologies?
- Hard to perform fault-tolerant reasoning?
- ...

Concept refinements and their applications in Description Logics and knowledge engineering

Nicolas Troquard

Free University of Bozen-Bolzano

with Roberto Confalonieri, Pietro Galliani, Maria Hedblom, Oliver Kutz, Rafael Peñaloza, Daniele Porello, Guendalina Righetti

KRDB Summer Online Seminars 2020 — 7 August

Outline

Subcconcepts

Definition

Let $\mathcal{O}$ be a $\mathcal{DL}$ ontology. The set of *subconcepts* of $\mathcal{O}$ is given by

$$\mathsf{sub}(\mathcal{O}) = \{\top, \bot\} \cup \bigcup_{C(a)\in\mathcal{O}} \mathsf{sub}(C) \cup \bigcup_{C\sqsubseteq D\in\mathcal{O}} \mathsf{sub}(C) \cup \mathsf{sub}(D) \ ,$$

where $\mathsf{sub}(C)$ is the set of subconcepts in C.

back

Upward and Downward cover sets of concepts

The upward cover of the concept C collects the most specific subconcepts of the ontology $\mathcal{O}$ that subsume C.

The downward cover of C collects the most general subconcepts from $\mathcal{O}$ subsumed by C.

Definition

Let $\mathcal{O}$ be a $\mathcal{DL}$ ontology and C a concept. The upward cover and downward cover of C w.r.t. $\mathcal{O}$ are:

$$\begin{aligned}
\mathsf{UpCov}_{\mathcal{O}}(C) := \{D \in \mathsf{sub}(\mathcal{O}) \mid\; & C \sqsubseteq_{\mathcal{O}} D \text{ and} \\
& \nexists . D' \in \mathsf{sub}(\mathcal{O}) \text{ with } C \sqsubset_{\mathcal{O}} D' \sqsubset_{\mathcal{O}} D\}, \\
\mathsf{DownCov}_{\mathcal{O}}(C) := \{D \in \mathsf{sub}(\mathcal{O}) \mid\; & D \sqsubseteq_{\mathcal{O}} C \text{ and} \\
& \nexists . D' \in \mathsf{sub}(\mathcal{O}) \text{ with } D \sqsubset_{\mathcal{O}} D' \sqsubset_{\mathcal{O}} C\}.
\end{aligned}$$

back

Example: Limits of considering only subconcepts of the ontology

$\mathsf{UpCov}_{\mathcal{O}}$ and $\mathsf{DownCov}_{\mathcal{O}}$ miss interesting refinements.

Example

Let $N_C = \{A, B, C\}$ and $\mathcal{O} = \{A \sqsubseteq B\}$.

- We have $\mathsf{UpCov}_{\mathcal{O}}(A \sqcap C) = \{A\}$.
- Iterating: $\mathsf{UpCov}_{\mathcal{O}}(A) = \{A, B\}$ and $\mathsf{UpCov}_{\mathcal{O}}(B) = \{B, \top\}$.
- $B \sqcap C$ is missed by the iterated application of $\mathsf{UpCov}_{\mathcal{O}}$ from $A \sqcap C$.
- Similarly, $\mathsf{UpCov}_{\mathcal{O}}(\exists R.A) = \{\top\}$, while we can expect $\exists R.B$ to be a generalisation of $\exists R.A$.

back

Abstract refinement, generalisation, and specialisation operator

The abstract refinement operator $\zeta_{\uparrow,\downarrow}$ is defined by induction on the structure of concept descriptions.

$$
\begin{aligned}
\zeta_{\uparrow,\downarrow}(\top) &= \uparrow(\top)\\
\zeta_{\uparrow,\downarrow}(\bot) &= \uparrow(\bot)\\
\zeta_{\uparrow,\downarrow}(A) &= \uparrow(A), \quad A \in N_C\\
\zeta_{\uparrow,\downarrow}(\neg C) &= \{\neg C' \mid C' \in \zeta_{\downarrow,\uparrow}(C)\} \cup \uparrow(\neg C)\\
\zeta_{\uparrow,\downarrow}(C \sqcap D) &= \{C' \sqcap D \mid C' \in \zeta_{\uparrow,\downarrow}(C)\} \cup\\
&\quad \{C \sqcap D' \mid D' \in \zeta_{\uparrow,\downarrow}(D)\} \cup \uparrow(C \sqcap D)\\
\zeta_{\uparrow,\downarrow}(C \sqcup D) &= \{C' \sqcup D \mid C' \in \zeta_{\uparrow,\downarrow}(C)\} \cup\\
&\quad \{C \sqcup D' \mid D' \in \zeta_{\uparrow,\downarrow}(D)\} \cup \uparrow(C \sqcup D)\\
\zeta_{\uparrow,\downarrow}(\forall R.C) &= \{\forall R.C' \mid C' \in \zeta_{\uparrow,\downarrow}(C)\} \cup \uparrow(\forall R.C)\\
\zeta_{\uparrow,\downarrow}(\exists R.C) &= \{\exists R.C' \mid C' \in \zeta_{\uparrow,\downarrow}(C)\} \cup \uparrow(\exists R.C)
\end{aligned}
$$

The generalisation operator and specialisation operator are defined, respectively, as

$$
\begin{aligned}
\gamma_{\mathcal{O}} &= \zeta_{\mathsf{UpCov}_{\mathcal{O}},\mathsf{DownCov}_{\mathcal{O}}} \quad \text{, and}\\
\rho_{\mathcal{O}} &= \zeta_{\mathsf{DowCov}_{\mathcal{O}},\mathsf{UpCov}_{\mathcal{O}}} \quad .
\end{aligned}
$$

back

Example (continued)

- $N_C = \{A, B, C\}$.
- $\mathcal{O} = \{A \sqsubseteq B\}$.
- $B \sqcap C$ is not in the iterated $\mathsf{UpCov}_{\mathcal{O}}$ from $A \sqcap C$.
- But $\gamma_{\mathcal{O}}(A \sqcap C) = \{A \sqcap C, B \sqcap C, A \sqcap \top, A\}$.

Details:

$$\gamma_{\mathcal{O}}(A \sqcap C) = \{A' \sqcap C \mid A' \in \gamma_{\mathcal{O}}(A)\} \cup \{A \sqcap C' \mid C' \in \gamma_{\mathcal{O}}(C)\} \cup \mathsf{UpCov}_{\mathcal{O}}(A \sqcap C)$$

- $\gamma_{\mathcal{O}}(A) = \mathsf{UpCov}_{\mathcal{O}}(A)$
- $\gamma_{\mathcal{O}}(C) = \mathsf{UpCov}_{\mathcal{O}}(C)$
- $\mathsf{UpCov}_{\mathcal{O}}(A) = \{A, B\}$
- $\mathsf{UpCov}_{\mathcal{O}}(C) = \{\top\}$
- $\mathsf{UpCov}_{\mathcal{O}}(A \sqcap C) = \{A\}$

back

Complexity (case of $\gamma_{\mathcal{O}}$ with an $\mathcal{EL}$ ontology)

Lemma

When $\mathcal{DL} = \mathcal{EL}$, deciding whether $D \in \gamma_{\mathcal{O}}(C)$ is PTime-hard.

- Deciding whether $D \in \mathsf{UpCov}_{\mathcal{O}}(C)$ is as hard as atomic subsumption.
- Deciding whether $D \in \gamma_{\mathcal{O}}(C)$ is as hard as deciding whether $C' \in \mathsf{DownCov}_{\mathcal{O}}(C)$.

Lemma

When $\mathcal{DL} = \mathcal{EL}$, $\mathsf{UpCov}_{\mathcal{O}}(C)$ is computable in polynomial time.

- $\mathbf{card}(\mathsf{sub}(\mathcal{O}))$ is linear in $|\mathcal{O}|$.
- Deciding whether $D \in \mathsf{UpCov}_{\mathcal{O}}(C)$ requires at most $1 + 4 \times \mathbf{card}(\mathsf{sub}(\mathcal{O}))$ calls to the subroutine for $\mathcal{DL}$ concept subsumption.
- It suffices to check for every $D \in \mathsf{sub}(\mathcal{O})$ whether $D \in \mathsf{UpCov}_{\mathcal{O}}(C)$ and collect those concepts for which the answer is positive.

Lemma

When $\mathcal{DL} = \mathcal{EL}$, deciding whether $D \in \gamma_{\mathcal{O}}(C)$ is in PTime.

- $\mathbf{card}(\gamma_{\mathcal{O}}(C)) \leq (|\mathcal{O}| + 2) \times |C|$.
- We can decide whether $\gamma_{\mathcal{O}}(C)$ contains a particular concept by computing only a linear number of times $\mathsf{UpCov}_{\mathcal{O}}(C')$, where $|C'|$ is linearly bounded by $|C'| + |\mathcal{O}|$.

back

$\mathcal{SROIQ}$ upward and downward covers

Let $\mathcal{O} = \mathcal{T} \cup \mathcal{R} \cup \mathcal{A}$ be an ontology.

Let C be a concept, the *upward cover* and *downward cover* of C wrt. $\mathcal{O}$ are as for $\mathcal{ALC}$.

Let r be a role *name*, the *upward cover* and *downward cover* of r wrt. $\mathcal{O}$ (where $N_R^- = \{r^- \mid r \in N_R\}$):

$$\begin{aligned}
\mathsf{UpCov}_{\mathcal{O}}(r) := \{ & s \in N_R \cup N_R^- \cup \{E, U\} \mid r \sqsubseteq_{\mathcal{O}} s \text{ and} \\
& \nexists . s' \in N_R \cup N_R^- \cup \{E, U\} \text{ with } r \sqsubset_{\mathcal{O}} s' \sqsubset_{\mathcal{O}} s \text{ and} \\
& s \text{ is simple in } \mathcal{R}\}. \\
\mathsf{DownCov}_{\mathcal{O}}(r) := \{ & s \in N_R \cup N_R^- \cup \{E, U\} \mid s \sqsubseteq_{\mathcal{O}} r \text{ and} \\
& \nexists . s' \in N_R \cup N_R^- \cup \{E, U\} \text{ with } r \sqsubset_{\mathcal{O}} s' \sqsubset_{\mathcal{O}} s \text{ and} \\
& s \text{ is simple in } \mathcal{R}\}.
\end{aligned}$$

Let n be a non-negative integer:

$$\mathsf{UpCov}_{\mathcal{O}}(n) := \{n, n+1\},$$

$$\mathsf{DownCov}_{\mathcal{O}}(n) := \begin{cases} \{n-1, n\} & \text{when } n > 1 \\ \{n\} & \text{when } n = 0. \end{cases}$$

$\mathcal{SROIQ}$ concepts and roles refinement operators

Role restrictions are upgraded from $\mathcal{ALC}$ with role refinements:

$$\zeta_{\uparrow,\downarrow}(\forall R.C) = \{\forall R'.C \mid R' \in \downarrow(R)\} \cup \{\forall R.C' \mid C' \in \zeta_{\uparrow,\downarrow}(C)\} \cup \uparrow(\forall R.C)$$

$$\zeta_{\uparrow,\downarrow}(\exists R.C) = \{\exists R'.C \mid R' \in \uparrow(R)\} \cup \{\exists R.C' \mid C' \in \zeta_{\uparrow,\downarrow}(C)\} \cup \uparrow(\exists R.C)$$

$\mathcal{SROIQ}$ concepts:

$$\zeta_{\uparrow,\downarrow}(\exists R.\mathit{Self}) = \{\exists R'.\mathit{Self} \mid R' \in \uparrow(R)\} \cup \uparrow(\exists R.\mathit{Self})$$

$$\zeta_{\uparrow,\downarrow}(\{i\}) = \uparrow(\{i\})$$

$$\begin{aligned}\zeta_{\uparrow,\downarrow}(\leq n\ R.C) = {} & \{\leq m\ R.C \mid m \in \uparrow(n)\} \cup \{\leq n\ R'.C \mid R' \in \downarrow(R)\} \cup \\ & \{\leq n\ R.C' \mid C' \in \zeta_{\downarrow,\uparrow}(C)\} \cup \uparrow(\leq n\ R.C)\end{aligned}$$

$$\begin{aligned}\zeta_{\uparrow,\downarrow}(\geq n\ R.C) = {} & \{\geq m\ R.C \mid m \in \downarrow(n)\} \cup \{\geq n\ R'.C \mid R' \in \uparrow(R)\} \cup \\ & \{\geq n\ R.C' \mid C' \in \zeta_{\uparrow,\downarrow}(C)\} \cup \uparrow(\geq n\ R.C)\end{aligned}$$

$\mathcal{SROIQ}$ roles:

$$\zeta_{\uparrow,\downarrow}(r) = \uparrow(r)$$

$$\zeta_{\uparrow,\downarrow}(r^-) = \{s^- \mid s \in \uparrow(r),\ s \in N_R\} \cup \{s \mid s^- \in \uparrow(r),\ s^- \in N_R^-\}$$

$$\zeta_{\uparrow,\downarrow}(U) = \uparrow(U)$$

$$\zeta_{\uparrow,\downarrow}(E) = \uparrow(E)$$