Freie Universität Bozen
Libera Università di Bolzano
Università Liedia de Bulsan

Fakultät für Ingenieurwesen
Facoltà di Ingegneria
Faculty of Engineering

Bachelor Thesis

Knowledge Refinement in Expressive Description Logics

Candidate Roland Bernard

Supervisors Oliver Kutz
Nicolas Troquard

July 2023

Description Logics

- Family of logics used to represent knowledge
  $\rightarrow$ aiming for favorable trade-offs between complexity and expressivity

- Individuals, e.g., $roland \quad oliver \quad nicolas \quad unibz \quad blue$
  Concepts, e.g., $Student \quad Professor \quad Person \quad University \quad Color$
  Roles, e.g., $studiesAt \quad supervisedBy \quad hasColor$

- Complex concepts, e.g., $\neg Person \quad Person \sqcap Student \quad \exists studiesAt.University$

- Axioms, e.g., $Student \sqsubseteq Person \quad Student(roland) \quad studiesAt(roland, unibz)$

Knowledge Refinement in Description Logics

- Process of iteratively modifying and improving the ontology

- Using two refinement operators
  - → specialization operator, e.g., $\mathit{Student}$ is a specialization of Person
  - → generalization operator, e.g., Person is a generalization of $\mathit{Student}$

- Using an axiom weakening operator
  - → uses the two refinement operators
  - → generates axioms that are less restrictive

Applications of Knowledge Refinement

Repairing ontologies, e.g.,

→ making inconsistent ontologies consistent

→ removing unintended consequences

Combination of conflicting knowledge

→ also for computational concept combination

Machine learning

→ learning axioms from data

Weakening in Expressive Description Logics

Extend axiom weakening to the description logic $\mathcal{SROIQ}$

Only simple roles can be used in every context

→ using a non-simple role in some places is forbidden

The graph formed by role inclusions must match some conditions

→ adding new role inclusions can cause violations

These problems have been prevented by ensuring that

→ all simple roles remain simple after refinement

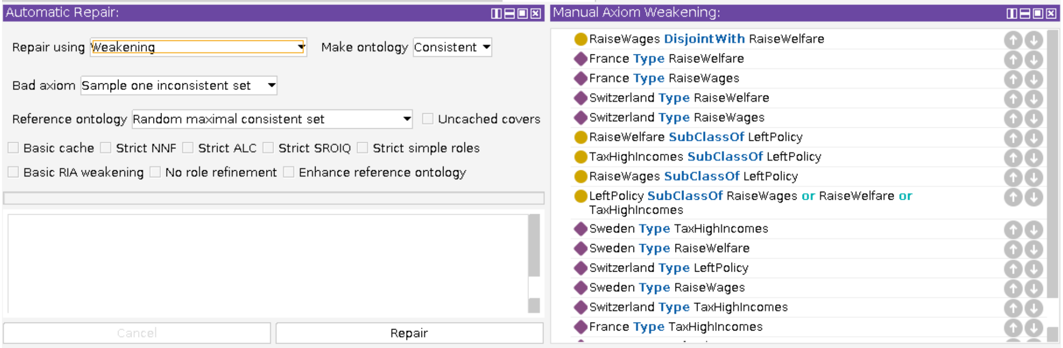
→ only simple roles are used during the refinement

A Protégé Plugin supporting Axiom Weakening

Protégé plugin for axiom weakening

→ allow computing weakening for specific axioms

→ enable automatic ontology repair



Evaluating Axiom Weakening for Ontology Repair

Repaired once using axiom weakening and once using removal

$\rightarrow$ the quality of the resulting repairs is compared

Deciding which repair is “better” is not well-defined

$\rightarrow$ we want to retain as many consequences as possible

$\rightarrow$ we focus only on subsumption between simple concepts

For comparing two repairs we define the IIC of $\mathcal{O}_1$ w.r.t $\mathcal{O}_2$

$\rightarrow$ value close to 1 for when $\mathcal{O}_1$ is better

$\rightarrow$ 0.5 if both repairs are equally “good”

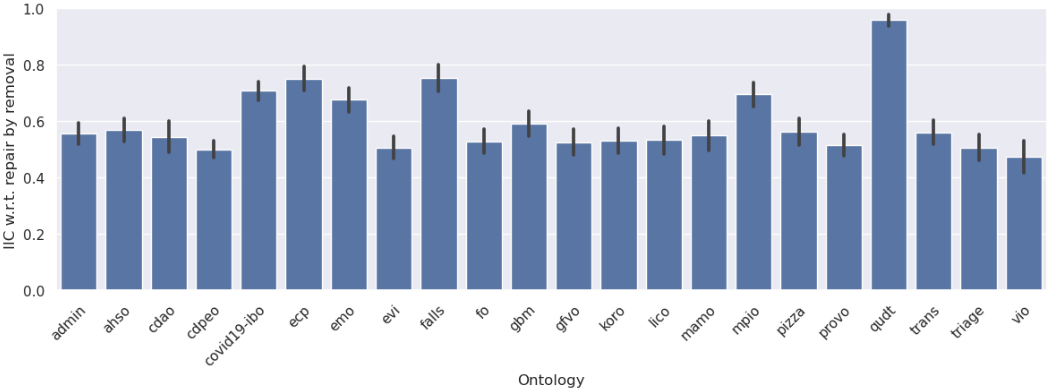
$\rightarrow$ value close to 0 for when $\mathcal{O}_2$ is better

Evaluation Results

Comparison between using axiom weakening and using removal

→ significantly better for some ontologies

→ in many cases only minor or no improvement



Outcomes of the Thesis

Extended the axiom weakening operator to $\mathcal{SROIQ}$

→ and showed that the proposed approach maintains the necessary constraints

Developed a Protégé plugin for applying these techniques

→ allowing users to easily repair ontologies and weaken axioms

Evaluated the proposed approach on real-world ontologies

→ showing that axiom weakening can outperform removal

Expressive Description Logics

- Additional kinds of concept expressions and axioms, e.g.,
  - → role inclusions, e.g., $\mathit{motherOf} \sqsubseteq \mathit{parentOf}$
  - → cardinality constraints, e.g., $\leq 1\ \mathit{studiesAt}.\mathit{University}$ $\quad \geq 2\ \mathit{supervisedBy}.\top$

- Require additional rules to guarantee decidability
  - → separation into simple and non-simple roles
  - → limits on the graph formed by role inclusions

◎ Focus on $\mathcal{SROIQ}$ and the Web Ontology Language

Implementation of Axiom Weakening

Implemented in Java using the OWL API

→ a library providing a uniform way of interfacing with reasoners

Using of-the-shelf reasoners for the Web Ontology Language

→ requires mapping between Web Ontology Language and $\mathcal{SROIQ}$

Tests to ensure correct behavior of the implementation

→ manual tests of expected operator results

→ automatically generated tests asserting general properties and invariants

Future Outlook

Study the possibility of loosening the restriction

→ refine with non-simple roles in some cases

→ more permissive weakening of role inclusions

Study better ways of guiding the repair process

→ using better heuristics, maybe domain specific

→ using user input to guide the repairs

Find better measures for comparing the quality of repairs

Study other possible applications of axiom weakening