Finding Proofs for Description Logic Entailments in Practice

Based on “Finding Small Proofs for Description Logic Entailments—Theory and Practice” (LPAR’20) // Explainable Logic-Based Knowledge Representation (XLoKR 2020), September 14, 2020
Description Logics and Ontologies

Syntax of DL $\mathcal{ALC}$

Concepts: $C ::= A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.C \mid \forall r.C$

Axioms: $\alpha ::= C \sqsubseteq C \mid C \equiv C$

Description Logics

- Well-established formalism for specifying terminological knowledge in Ontologies
- Used for many large-scale ontologies
  - SNOMED CT: over 300,000 concepts
  - BioPortal: repository of bio-medical ontologies, currently hosting 889 ontologies defining 12,084,317 terms
  - MOWLCorp: ontologies obtained by web-crawling, containing 21,000 ontologies
Description Logics and Ontologies

Syntax of DL $\mathcal{ALC}$

Concepts: $C ::= A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r. C \mid \forall r. C$

Axioms: $\alpha ::= C \sqsubseteq C \mid C \equiv C$

Description Logics

- Well-established formalism for specifying terminological knowledge in Ontologies
- Used for many large-scale ontologies
- With increasing complexity of the ontology, understanding entailments becomes both crucial and difficult
Description Logics and Ontologies

Syntax of DL $\mathcal{ALC}$

Concepts: $C ::= A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.C \mid \forall r.C$

Axioms: $\alpha ::= C \sqsubseteq C \mid C \equiv C$

Description Logics

- Well-established formalism for specifying terminological knowledge in Ontologies
- Used for many large-scale ontologies
- With increasing complexity of the ontology, understanding entailments becomes both crucial and difficult
- One typical reasoning task is classification
  - compute all entailed axioms of form $A \sqsubseteq B$
  - obtain concept hierarchy
Current Tool of Choice: Justifications

[Diagram showing a detailed ontology with classes such as AnthraxVaccine, AntidiureticAgent, BCGVaccine, BodySystemPhenomenon, AnatomicalSystemSign, CardiovascularSystemPhenomenon, CentralNervousSystemDisorder, DigestiveSystemPhenomenon, EndocrineSystemPhenomenon, ExaminingProcessInvolvesBodySystem, GenitoUrinarySystemPhenomenon, HaematologicalDisorder, HaematologicalSystemPathology, ImmuneSystemPhenomenon, LymphoreticularSystemPhenomenon, MentalDisease, MentalDisorder, MentalDisturbance, MentalIllness, MusculoSkeletalSystemPhenomenon, NervousSystemPhenomenon, ExaminingProcessInvolvesNervousSystem, NervousSystemSign, PathologicalProcessInvolvesNervousSystem, SignPhenomenonInvolvesNervousSystem, OroDentalSystemPhenomenon.]
Current Tool of Choice: Justifications
Current Tool of Choice: Justifications

Finding Proofs for Description Logic Entailments in Practice
Technische Universität Dresden // Christian Alrabbaa, Stefan Borgwardt, Patrick Koopmann, Alisa Kovtunova
Explainable Logic-Based Knowledge Representation (XLoKR 2020), September 14, 2020
Current Tool of Choice: Justifications
Justifications

**Justifications:** Minimal subsets entailing given subsumption

In practice often insufficient:
- can be large
- inferences often not obvious

Showing *how* to obtain the inference would be better:
- simple reasoning steps leading to conclusion
- generally known as proof
Proofs for ELK in Protege
Proofs for ELK in Protege
Proofs for ELK in Protege
Proofs for ELK in Protege

Slide 4 of 25
Proofs for ELK in Protege

Finding Proofs for Description Logic Entailments in Practice
Technische Universität Dresden // Christian Alrabbaa, Stefan Borgwardt, Patrick Koopmann, Alisa Kovtunova
Explainable Logic-Based Knowledge Representation (XLoKR 2020), September 14, 2020
Proofs using Evonne (work in progress)

Proofs with ELK

- ELK using a consequence-based reasoning procedure
  ⇒ inferences performed using a calculus

\[
\begin{align*}
R_0 & \quad C \sqsubseteq C \\
R_T & \quad C \sqsubseteq T \\
R_\cap & \quad C \sqsubseteq D \sqcap E \\
R_\sqcap & \quad C \sqsubseteq \exists r.D \\
R_\sqcup & \quad C \sqsubseteq D ; E \\
R_\circ & \quad C_0 \sqsubseteq \exists r_1.C_1, C_1 \sqsubseteq \exists r_2.C_2 \ldots C_{n-1} \sqsubseteq \exists r_n.C_n : r_1 \circ \ldots \circ r_n \sqsubseteq r \in O
\end{align*}
\]

⇒ proofs generated as part of reasoning process
Proofs For More Expressive DLs

- Currently, ELK is the only DL reasoner supporting proof generation
- ELK supports only a limited fragment of OWL, OWL EL
- More expressive reasoner often use other reasoning procedures
  - for a long time prominent: tableau reasoning
  - less convenient for understanding entailments
- Existing consequence-based reasoning for expressive DLs
  - often involved in complex systems
  - often combined with other reasoning paradigms
  - may use normal forms requiring different syntax
  - generation of proofs not obvious
- Can we generate proofs without a calculus?
Justification Based Proofs

Justification-Based Proofs (Matthew Horridge 2011)

- derive intermediate steps between conclusion and justification
- consider all axioms of some predefined shapes
- justification-relation between allows to construct a proof
- involved ranking function allows to select axioms to be used in proof
Justification Based Proofs

Justification-Based Proofs (Matthew Horridge 2011)

- derive intermediate steps between conclusion and justification
- consider all axioms of some predefined shapes
- justification-relation between allows to construct a proof
- involved ranking function allows to select axioms to be used in proof
Justification Based Proofs

Justification-Based Proofs (Matthew Horridge 2011)

- derive intermediate steps between conclusion and justification
- consider all axioms of some predefined shapes
- justification-relation between allows to construct a proof
- involved ranking function allows to select axioms to be used in proof
Justification Based Proofs

Justification-Based Proofs (Matthew Horridge 2011)

- derive intermediate steps between conclusion and justification
- consider all axioms of some predefined shapes
- justification-relation between allows to construct a proof
- involved ranking function allows to select axioms to be used in proof
Justification Based Proofs

Justification-Based Proofs (Matthew Horridge 2011)

- derive intermediate steps between conclusion and justification
- consider all axioms of some predefined shapes
- justification-relation between allows to construct a proof
- involved ranking function allows to select axioms to be used in proof
Justification Based Proofs

Justification-Based Proofs (Matthew Horridge 2011)

- derive intermediate steps between conclusion and justification
- consider all axioms of some predefined shapes
- justification-relation between allows to construct a proof
- involved ranking function allows to select axioms to be used in proof
Justification Based Proofs

Justification-Based Proofs (Matthew Horridge 2011)

- derive intermediate steps between conclusion and justification
- consider all axioms of some predefined shapes
- justification-relation between allows to construct a proof
- involved ranking function allows to select axioms to be used in proof
Justification Based Proofs

Justification-Based Proofs (Matthew Horridge 2011)

- derive intermediate steps between conclusion and justification
- consider all axioms of some predefined shapes
- justification-relation between allows to construct a proof
- involved ranking function allows to select axioms to be used in proof
Justification Based Proofs

Justification-Based Proofs (Matthew Horridge 2011)

- derive intermediate steps between conclusion and justification
- consider all axioms of some predefined shapes
- justification-relation between allows to construct a proof
- involved ranking function allows to select axioms to be used in proof

Advantage of approach:

- generates best proof according to ranking

Disadvantage of approach:

- no clear inference principle
- hard to implement
- strongly depends on ranking function
Forgetting-Based Proofs

Forgetting Based Proofs

- Idea from propositional resolution:

\[
\frac{q_1 \lor p \quad q_2 \lor \neg p}{p_1 \lor p_2}
\]

→ inference through elimination of \( p \)
Forgetting-Based Proofs

- Idea from propositional resolution:

\[
\frac{q_1 \lor p \quad q_2 \lor \neg p}{p_1 \lor p_2}
\]

\[\Rightarrow\] inference through elimination of \( p \)

- Decide satisfiability by eliminating names one after the other:

\[
\begin{align*}
    b & \quad a \lor b \\
    \neg b \lor c & \quad \neg b \lor \neg c \\
    \neg a \lor c
\end{align*}
\]
Forgetting-Based Proofs

- Idea from propositional resolution:

\[
\frac{q_1 \lor p \quad q_2 \lor \neg p}{p_1 \lor p_2}
\]

⇒ inference through elimination of \( p \)

- Decide satisfiability by eliminating names one after the other:

\[
\begin{align*}
    b & \quad a \lor b & \quad \neg b \lor c & \quad \neg b \lor \neg c & \quad \neg a \lor c
\end{align*}
\]
Forgetting-Based Proofs

Idea from propositional resolution:

\[
\frac{q_1 \lor p \quad q_2 \lor \neg p}{p_1 \lor p_2}
\]

\(\Rightarrow\) inference through elimination of \(p\)

Decide satisfiability by eliminating names one after the other:

\[b \quad \neg b \lor c \quad \neg b \lor \neg c \quad b \lor c\]
Forgetting-Based Proofs

Idea from propositional resolution:

\[
\frac{q_1 \lor p \qquad q_2 \lor \neg p}{p_1 \lor p_2}
\]

⇒ inference through elimination of \( p \)

- Decide satisfiability by eliminating names one after the other:

\[
\begin{align*}
  b & \quad \neg b \lor c & \quad \neg b \lor \neg c & \quad b \lor c \\
\end{align*}
\]
Forgetting-Based Proofs

Forgetting Based Proofs

- Idea from propositional resolution:

  \[
  \frac{q_1 \lor p \quad q_2 \lor \neg p}{p_1 \lor p_2}
  \]

  \(\Rightarrow\) inference through elimination of \(p\)

- Decide satisfiability by eliminating names one after the other:

  \[c \quad \neg c\]
Forgetting-Based Proofs

Idea from propositional resolution:

\[ \frac{q_1 \lor p \quad q_2 \lor \neg p}{p_1 \lor p_2} \]

\[ \rightarrow \text{inference through elimination of } p \]

Decide satisfiability by eliminating names one after the other:

\[ c \quad \neg c \]
Forgetting-Based Proofs

Idea from propositional resolution:

\[
\frac{q_1 \lor p \quad q_2 \lor \neg p}{p_1 \lor p_2}
\]

inference through elimination of \( p \)

Decide satisfiability by eliminating names one after the other:

\[\bot\]
Forgetting-Based Proofs

- Idea from propositional resolution:
  \[
  \begin{array}{c|c}
  q_1 \lor p & q_2 \lor \neg p \\
  \hline
  p_1 \lor p_2
  \end{array}
  \]

\[\Rightarrow\] inference through elimination of \( p \)
- Decide satisfiability by eliminating names one after the other:

\[\bot\]

- Idea: Use similar approach to prove axioms of form \( A \sqsubseteq B \)
Forgetting-Based Proofs

Forgetting Based Proofs

- Idea from propositional resolution:
  \[ q_1 \lor p \quad q_2 \lor \neg p \]
  \[
  \frac{p_1 \lor p_2}{p_1 \lor p_2} 
  \]

\[ \Rightarrow \] inference through elimination of \( p \)

- Decide satisfiability by eliminating names one after the other:

\[ \bot \]

- Idea: Use similar approach to prove axioms of form \( A \sqsubseteq B \)

\[ A \sqsubseteq C \quad C \sqsubseteq \exists r.D \quad \exists r.\top \sqsubseteq B \]
Forgetting-Based Proofs

- Idea from propositional resolution:
  \[
  \frac{q_1 \lor p}{p_1 \lor p_2}, \quad \frac{q_2 \lor \neg p}{p_1 \lor p_2}
  \]

  \[\Rightarrow\] inference through elimination of \( p \)

- Decide satisfiability by eliminating names one after the other:
  \[\bot\]

- Idea: Use similar approach to prove axioms of form \( A \sqsubseteq B \)

  \[A \sqsubseteq C, \quad C \sqsubseteq \exists r.D, \quad \exists r.\top \sqsubseteq B\]
Forgetting-Based Proofs

Forgetting Based Proofs

- Idea from propositional resolution:

\[
\frac{q_1 \lor p \quad q_2 \lor \neg p}{p_1 \lor p_2}
\]

\[\Rightarrow\] inference through elimination of \( p \)

- Decide satisfiability by eliminating names one after the other:

\[\bot\]

- Idea: Use similar approach to prove axioms of form \( A \sqsubseteq B \)

\[
A \sqsubseteq C \quad C \sqsubseteq \exists r.T \quad \exists r.T \sqsubseteq B
\]
Forgetting-Based Proofs

Idea from propositional resolution:

\[
\begin{align*}
q_1 \lor p & \quad q_2 \lor \neg p \\
p_1 \lor p_2
\end{align*}
\]

⇒ inference through elimination of \( p \)

Decide satisfiability by eliminating names one after the other:

\[ \bot \]

Idea: Use similar approach to prove axioms of form \( A \sqsubseteq B \)

\[
\begin{align*}
A \sqsubseteq C & \quad C \sqsubseteq \exists r. \top & \quad \exists r. \top \sqsubseteq B
\end{align*}
\]
Forgetting-Based Proofs

Forgetting Based Proofs

- Idea from propositional resolution:
  \[
  \frac{q_1 \lor p \quad q_2 \lor \neg p}{p_1 \lor p_2}
  \]

  ⇒ inference through elimination of \( p \)

  Decide satisfiability by eliminating names one after the other:

\[\bot\]

- Idea: Use similar approach to prove axioms of form \( A \sqsubseteq B \)
  \[
  A \sqsubseteq C \quad C \sqsubseteq B
  \]
Forgetting-Based Proofs

Idea from propositional resolution:
\[
\frac{q_1 \lor p}{p_1 \lor p_2} \quad \frac{q_2 \lor \neg p}{p_1 \lor p_2}
\]

⇒ inference through elimination of \( p \)

Decide satisfiability by eliminating names one after the other:
\[
\bot
\]

Idea: Use similar approach to prove axioms of form \( A \sqsubseteq B \)

\[
A \sqsubseteq C \quad C \sqsubseteq B
\]
Forgetting-Based Proofs

Idea from propositional resolution:

\[
\begin{array}{c}
q_1 \lor p \\
q_2 \lor \neg p \\
\hline
p_1 \lor p_2
\end{array}
\]

\[\Rightarrow \text{inference through elimination of } p\]

Decide satisfiability by eliminating names one after the other:

\[\bot\]

Idea: Use similar approach to prove axioms of form \(A \sqsubseteq B\)

\[A \sqsubseteq B\]
Forgetting

<table>
<thead>
<tr>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let $\mathcal{O}$ be an ontology and $X$ a predicate name. Then, $\mathcal{O}^{-X}$ is a result of forgetting $X$ iff</td>
</tr>
<tr>
<td>- $X$ does not occur in $\mathcal{O}^{-X}$</td>
</tr>
<tr>
<td>- for every axiom $\alpha$ in which $X$ does not occur, $\mathcal{O} \models \alpha$ iff $\mathcal{O}^{-X} \models \alpha$</td>
</tr>
</tbody>
</table>

⇒ strongest ontology without $X$ entailed by $\mathcal{O}$ |
| - which $\alpha$ to preserve also depends on underlying DL |
Forgetting based proofs

- Use forgetting to produce sequence of ontologies

\[ O_0 \quad \rightarrow \quad \neg X_0 \quad \rightarrow \quad O_1 \quad \rightarrow \quad \neg X_1 \quad \rightarrow \quad O_2 \quad \rightarrow \quad \neg X_2 \quad \rightarrow \quad O_3 \quad \rightarrow \quad \neg X_3 \quad \rightarrow \quad O_4 \]

In each step, recompute justification for \( A \sqsubseteq B \)

Finally, reconstruct proof using justifications
Forgetting based proofs

- Use forgetting to produce sequence of ontologies

\[ \mathcal{O}_0 \rightarrow \mathcal{O}_1 \rightarrow \mathcal{O}_2 \rightarrow \mathcal{O}_3 \rightarrow \mathcal{O}_4 \]

- In each step, recompute justification for \( A \sqsubseteq B \)
Forgetting based proofs

- Use forgetting to produce sequence of ontologies

\[ \mathcal{O}_0 \rightarrow \mathcal{O}_1 \rightarrow \mathcal{O}_2 \rightarrow \mathcal{O}_3 \rightarrow \mathcal{O}_4 \]

- In each step, recompute justification for $A \sqsubseteq B$
- Finally, reconstruct proof using justifications
Forgetting based proofs

- Use forgetting to produce sequence of ontologies

- In each step, recompute justification for $A \subseteq B$
- Finally, reconstruct proof using justifications, skipping steps if it makes sense
Forgetting Based Proof

\[(C) \quad A \sqsubseteq C  \quad (D) \quad C \sqsubseteq \exists r. D \quad (r) \quad A \sqsubseteq \exists r. T \quad \exists r. T \sqsubseteq B \]

\
\[
A \sqsubseteq B
\]
Forgetting Order

Forgetting order, as well as selection of justification, affects proof

- Forgetting $D$ first:

  \[
  \frac{A \sqsubset C}{A \sqsubset \exists r. T}
  \]

- Forgetting $r$ first:

  \[
  \frac{A \sqsubset C}{A \sqsubset B}
  \]
Forgetting Order

To obtain nicer proofs *practically*, we process names using the following heuristics:

- role with non-trivial fillers last:
  - otherwise may hide most of inference:

\[
\begin{align*}
(r) & \quad A \sqsubseteq \exists r.B \\
& & \quad A \sqsubseteq \forall r.C \sqcup D \\
& & \quad B \sqsubseteq \exists s.D \\
& & \quad C \sqsubseteq \forall s.\neg D \\
\end{align*}
\]

\[
A \sqsubseteq D
\]

- unnested names first
  - delay complex inferences

- less frequent names first
  - delay expensive forgetting operations
  - (also used by existing forgetting procedures)
Evaluation

- Implemented approach in modular fashion
  - easy exchange of different forgetting procedures, provided they produce OWL ontologies
  - easy comparison with proofs generated by ELK
    - Dijkstra-based search to extract shortest proof
  - use 2 forgetting tools in the evaluation
    - $ALCH$ variant of LETHE 0.6
    - $ALCOI$ variant of FAME 1.0

\[1\text{there is a much improved version FAME 2.0, but it often creates ontologies outside of the OWL-standard}\]
Evaluation: Corpus

- Focus on proofs in $\mathcal{ELH}$
  - to be able to compare with ELK
  - easier extraction of justification patterns (see below)
- Use ontologies from the OWL Reasoner Evaluation 2015, OWL EL Classification Track
  - well-balanced mix of ontologies from different repositories
- Extracted 1,573 justification patterns
  - all entailments of form $A \sqsubseteq B$ or $A \equiv B$
  - all justifications for these entailments
  - abstract away concept and role names
  - remove resulting duplicates
Evaluation: Metrics

- **Hypergraph-size**
  - number of distinct axioms used in the proof

- **Tree-size**
  - sub-proofs count as often as they are used

- **Justification Complexity**
  - Matthew et al. 2013: “Toward cognitive support for OWL justifications”
  - attempt to measure cognitive complexity of justification
  - provides value for each proof step
  - we measured maximum and sum for each proof
Evaluation: Proof size

![Graphs showing tree size and hypergraph size](image-url)
Evaluation: ELK vs. Forgetting-Based Proofs

- Top Superclass
  - B ⊆ T
    - Asserted Conclusion
      - A ⊆ ∃r.B
        - Existential Filler Expansion
          - ∃r.B ⊆ ∃r.T
            - Class Hierarchy
              - A ⊆ ∃r.T
                - Property Domain Translation
                  - ∃r.T ⊆ B
                    - Class Hierarchy
                      - A ⊆ B

- Asserted Conclusion
  - domain(r) = B

- Asserted Conclusion
  - asserted
    - A ⊆ ∃r.B
      - forgetting r
        - domain(r) = B
          - A ⊆ B
Evaluation: ELK Proof

**Diagram:**
- **Assumed Conclusion:**
  - $A \equiv (E \cap \exists r.D \cap \exists s.C \cap \exists t.F)$
  - $B \equiv (E \cap \exists r.D \cap \exists s.C \cap \exists t.F)$

- **Equivalent Classes Decomposition:**
  - $A \subseteq (E \cap \exists r.D \cap \exists s.C \cap \exists t.F)$
  - $(E \cap \exists r.D \cap \exists s.C \cap \exists t.F) \subseteq A$
  - $(E \cap \exists r.D \cap \exists s.C \cap \exists t.F) \subseteq B$
  - $B \subseteq (E \cap \exists r.D \cap \exists s.C \cap \exists t.F)$

- **Class Inclusion Cycle:**
  - $A \subseteq B$
  - $B \subseteq A$

- **Class Hierarchy:**
  - $A \equiv B$
Evaluation: Justification Complexity

Just. complexity (sum)

- LETHE (101) vs. FAME (462)
- ELK (411) vs. FAME (214)
- ELK (669) vs. LETHE (603)

Just. complexity (max)

- FAME (107) vs. LETHE (37)
- FAME (189) vs. ELK (374)
- LETHE (719) vs. ELK (493)
Evaluation: Complexity of Inferences

- Both LETHE and FAME may use logical operators outside of $\mathcal{EL}$

- Large number of distinct “inference rules” was used:
  - LETHE: 362 different rules
  - FAME: 281 different rules
Conclusion

- New proof generation procedure based on forgetting
- Generate proof by repeated use of forgetting and justification
- Proofs for expressive DLs without calculus
- Can sometimes even compete with ELK
- Several possibilities to improve:
  - better heuristics on forgetting order or when to skip steps
  - dynamic selection of forgetting order
  - use learned “rules” to shorten proof computation times
  - integrate newer version of FAME