TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Steffen Breuer

Generating Proofs in the Description Logic ALCH

Dresden, August 3, 2023

Motivation

- Debugging of ontologies
- Understanding results

Description Logic $\mathcal{ALCH}$

- negation ($\neg$)
- concept intersection ($\sqcap$)
- concept union ($\sqcup$)
- value restriction ($\forall$)
- existential restriction ($\exists$)
- top concept ($\top$)
- bottom concept ($\bot$)
- gerneral concept inclusion ($C \sqsubseteq D$)
- role inclusion ($r \sqsubseteq s$)

Semantics

Let $A \in N_C, C, D \in N_{\mathcal{ALCH}}, r \in N_R$ then

$$
\begin{array}{lcl}
\top^I & := & \Delta^I \\
\bot^I & := & \varnothing \\
(\neg C)^I & := & C^I \\
(C \sqcap D)^I & := & C^I \cap D^I \\
(C \sqcup D)^I & := & C^I \cup D^I \\
(\exists r.C)^I & := & \{a \in \Delta^I \mid \exists b \in \Delta^I : (a, b) \in r^I \text{ and } b \in C^I\} \\
(\forall r.C)^I & := & \{a \in \Delta^I \mid \forall b \in \Delta^I : \text{if}(a, b) \in r^I, \text{then } b \in C^I\}
\end{array}
$$

Semantics

- For $C, D \in N_{\mathcal{ALCH}} : I \models C \sqsubseteq D :\iff C^I \subseteq D^I$
- For $r, s \in N_R : I \models r \sqsubseteq s :\iff r^I \subseteq s^I$

Proofs

- $G = (V, E, l)$ finite, acyclic, directed, labeled hypergraph
  - V set of nodes
  - $E \subseteq V^n$ set of hyperedges
  - $l : V \to \mathcal{L}$ labeling function such that $\{l(v) \mid v \in S\} \models l(d)$

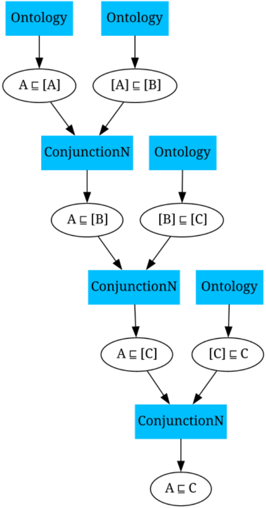

Figure: $\mathcal{O} \coloneqq \{A \sqsubseteq B, B \sqsubseteq C\}$, Goal: $A \sqsubseteq C$

The Algorithm

- Normalization
- Rule Application
- Proof Extraction

Normal Form

- $\bigsqcap A_i \sqsubseteq \bigsqcup B_j$
- $A \sqsubseteq \exists r.B$
- $\exists r.A \sqsubseteq A$
- $A \sqsubseteq \forall r.B$
- $r \sqsubseteq s$

Normalization

1. removal of axioms with negative $\forall r.C$ as subconcept
2. apply structural transformation
3. final normalization rules

Positive and Negative Occurrences Of Concepts

Let $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ some ontology and C some concept then we say that a concept C occurs positively (negatively) in $\mathcal{O}$ if C occurs positively (negatively) in some GCI of $\mathcal{T}$.

- C occurs positively in itself.
- If C occurs positively (negatively) in C′ then C occurs positively (negatively) in $C' \sqcap DC \sqcap D', C' \sqcup D, C \sqcup D', \exists r.C', \forall r.C', D \sqsubseteq C'$ and negatively (positively) in $\neg C', C' \sqsubseteq D$.

Structural Transformation

$$
\begin{array}{ll}
\mathrm{st}(A) := A & \mathrm{st}(C \sqcap D) := [C] \sqcap [D] \\
\mathrm{st}(\top) := \top & \mathrm{st}(C \sqcup D) := [C] \sqcup [D] \\
\mathrm{st}(\bot) := \top & \mathrm{st}(\exists r.C) := \exists r.[C] \\
\mathrm{st}(\neg C) := \neg[C] & \mathrm{st}(\exists r.C) := \exists r.[C]
\end{array}
$$

Structural Transformation

- $\mathrm{st}(C) \sqsubseteq [C]$ for every negative C in $\mathcal{O}$
- $[D] \sqsubseteq \mathrm{st}(D)$ for every positive D in $\mathcal{O}$
- $[C] \sqsubseteq [D]$ for every axiom $C \sqsubseteq D$ occurring in$\mathcal{O}$

Normalization Rules

- $$\frac{[C \sqcap D] \sqsubseteq [C] \sqcap [D]}{[C \sqcap D] \sqsubseteq [C], \quad [C \sqcap D] \sqsubseteq [D]}$$
- $$\frac{[C] \sqcup [D] \sqsubseteq [C \sqcup D]}{[C] \sqsubseteq [C \sqcup D], \quad [D] \sqsubseteq [C \sqcup D]}$$
- $$\frac{[\neg C] \sqsubseteq \neg [C]}{[\neg C] \sqcap [C] \sqsubseteq \bot}$$
- $$\frac{\neg [C] \sqsubseteq [\neg C]}{\top \sqsubseteq [\neg C] \sqcap [C]}$$

Rule Application

Procedure main(ontology, goal)

1. Normalizer.normalize(ontology)
2. proofHandler ← new ProofHandler(goal)
3. rules ← getRules(ontology, proofHandler)
4. **while** notFinished(proofHandler) **do**
   5. **for** (rule in rules) **do**
      6. setNewRule(rule.apply())

Proof Handler

- Active Axioms
- Active Concepts
- Inference Handling
- Goal Checking
- Proof Extraction

Rules

$$\mathbf{R}_{\mathbf{A}}^{+}\ \frac{}{H \sqsubseteq A} : A \in H \tag{1}$$

$$\mathbf{R}_{\mathbf{A}}^{-}\ \frac{H \sqsubseteq N \sqcup A}{H \sqsubseteq N} : \neg A \in H \tag{2}$$

$$\mathbf{R}_{\sqcap}^{\mathbf{n}}\ \frac{\{H \sqsubseteq N_i \sqcup A_i\}_{i=1}^{n}}{H \sqsubseteq \bigsqcup_{i=1}^{n} N_i \sqcup M} : \prod_{i=1}^{n} A_i \sqsubseteq M \in \mathcal{O} \tag{3}$$

$$\mathbf{R}_{\exists}^{+}\ \frac{H \sqsubseteq N \sqcup A}{H \sqsubseteq N \sqcup \exists r.B} : A \sqsubseteq \exists r.B \in \mathcal{O} \tag{4}$$

$$\mathbf{R}_{\exists}^{-}\ \frac{H \sqsubseteq M \sqcup \exists r.K, \quad K \sqsubseteq N \sqcup A}{H \sqsubseteq M \sqcup B \sqcup \exists r.(K \sqcap \neg A)} : \exists s.A \sqsubseteq B \in \mathcal{O} \quad r \sqsubseteq_{\mathcal{O}} s \tag{5}$$

$$\mathbf{R}_{\exists}^{\perp}\ \frac{H \sqsubseteq M \sqcup \exists r.K, \quad K \sqsubseteq \perp}{H \sqsubseteq M} \tag{6}$$

$$\mathbf{R}_{\forall}\ \frac{H \sqsubseteq M \sqcup \exists r.K, \quad H \sqsubseteq N \sqcup A}{H \sqsubseteq M \sqcup N \sqcup \exists r.(K \sqcap B)} : A \sqsubseteq \forall s.B \in \mathcal{O} \quad r \sqsubseteq_{\mathcal{O}} s \tag{7}$$

Consequence-Based Reasoning beyond Horn Ontologies [4]

$$\mathbf{R}_{\mathbf{A}}^{+} := \frac{}{H \sqsubseteq A} : A \in H$$

- ontology independent
- uses active concepts
- active concepts can be removed

$$\mathbf{R}_{\mathbf{A}}^{-} := \frac{H \sqsubseteq N \sqcup A}{H \sqsubseteq N} : \neg A \in H$$

- ontology independent
- uses active axioms
- Search
  1. search for axiom with $\neg A \in H$ (find all)
  2. check whether A in superconcept

$$\mathbf{R}_{\sqcap}^{\mathbf{n}} := \frac{\{H \sqsubseteq N_i \sqcup A_i\}_{i=1}^{n}}{H \sqsubseteq \bigsqcup_{i=1}^{n} N_i \sqcup M} : \bigsqcap_{i=1}^{n} A_i \sqsubseteq M \in \mathcal{O}$$

- filter ontology
  - list of tuples $(\{A_1, \dots A_n\}, \bigsqcap_{i=1}^{n} A_i \sqsubseteq M)$
- preprocess active axioms
  - $m(H, A) := \langle(\{N_c \mid N_c \in N\}, H \sqsubseteq N \sqcup A)\rangle$
- search
  1. check for each H if $\{A \mid m(H, A) \text{ defined}\} \subseteq \{A_1, \dots, A_n\}$
  2. create all $n^{\sum_{i=1}^{n} |m(H, A_i)|}$ combinations of $H \sqsubseteq \bigsqcup_{i=1}^{n} N_i \sqcup M$

$$\mathbf{R}_{\exists}^{+} := \frac{H \sqsubseteq N \sqcup A}{H \sqsubseteq N \sqcup \exists r.B} : A \sqsubseteq \exists r.B \in \mathcal{O}$$

- filter ontology for $A \sqsubseteq \exists r.B$
- for each A in ontology search for $H \sqsubseteq N \sqcup A$

$$\mathbf{R}_{\exists}^{-} := \frac{H \sqsubseteq M \sqcup \exists r.K, \quad K \sqsubseteq N \sqcup A}{H \sqsubseteq M \sqcup B \sqcup \exists r.(K \sqcap \neg A)} : \quad A \sqsubseteq \forall s.B \in \mathcal{O} \quad r \sqsubseteq_{\mathcal{O}} s$$

- filter ontology for r and A combinations
- search $K \sqsubseteq N \sqcup A$ for some A
- for found K search for $H \sqsubseteq M \sqcup \exists r.K$

$$\mathbf{R}_{\exists}^{\bot} := \frac{H \sqsubseteq M \sqcup \exists r.K, \quad K \sqsubseteq \bot}{H \sqsubseteq M}$$

- ontology independent
- first search for K then for H

$$\mathbf{R}_\forall := \frac{H \sqsubseteq M \sqcup \exists r.K, \quad H \sqsubseteq N \sqcup A}{H \sqsubseteq M \sqcup N \sqcup \exists r.(K \sqcap B)} :$$

$$A \sqsubseteq \forall s.B \in \mathcal{O} \quad r \sqsubseteq_{\mathcal{O}} s$$

- filter ontology for r and A combinations
- for each (A, r) search for $H \sqsubseteq N \sqcup A$
- based on H search for $H \sqsubseteq M \sqcup \exists r.K$

Comparison to Lethe

- LetheBasedALCHProofGenerator
- elimination based
- different calculus [3]
- also uses EVEE [2]
  - https://github.com/de-tu-dresden-inf-lat/evee
- tasks [1]

Task 1

- $\mathcal{O} \coloneqq \{A \sqsubseteq C, C \sqsubseteq D, D \sqsubseteq E, B \equiv (F \sqcap \exists s.\exists r.G), A \sqsubseteq \exists s.\exists r.G, E \equiv ((D \sqcup H \sqcup I) \sqcap F)\}$
- Goal: $A \sqsubseteq B$

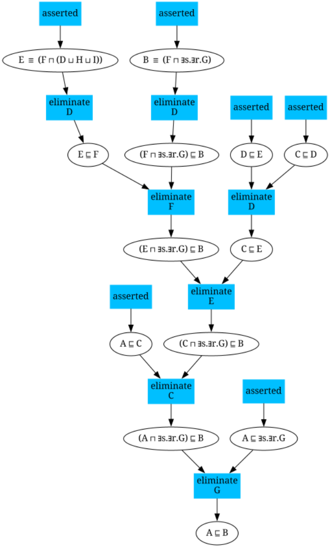

Figure: Lethe Proof for Task 1

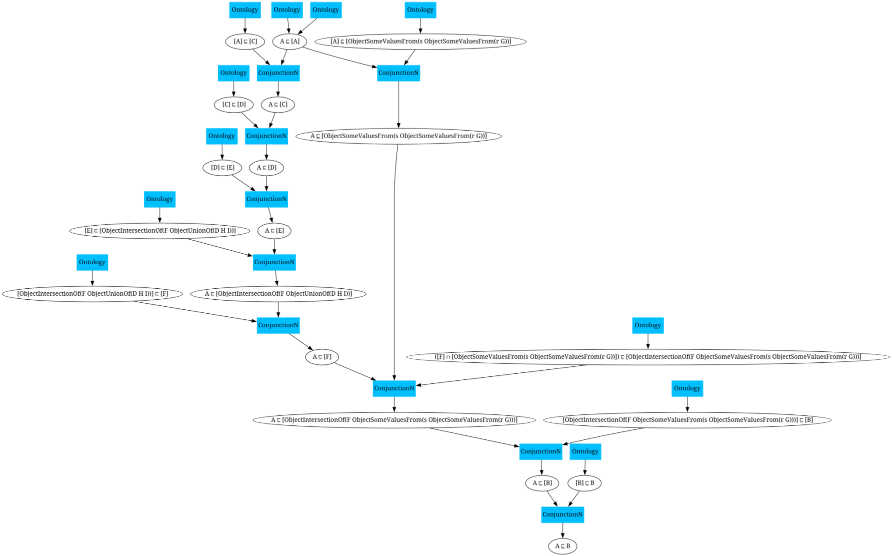

Figure: ALCH-Reasoner Proof for Task 1

Task 3

- $\mathcal{O} \coloneqq \{A \equiv (C \sqcap \exists r.(D \sqcap \forall s.E)), B \equiv (\exists r.D \sqcap C)\}$
- Goal: $A \sqsubseteq B$

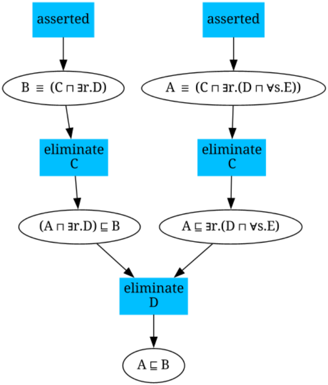

Figure: Lethe Proof for Task 3

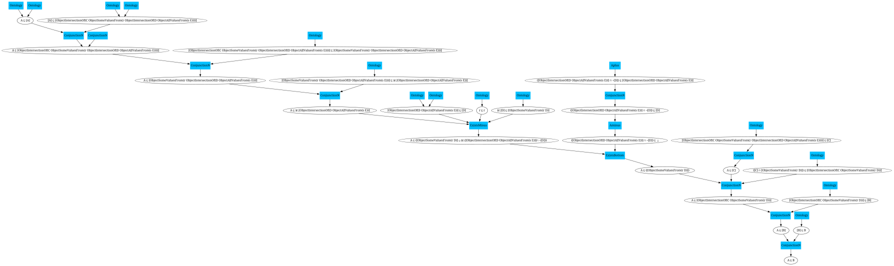

Figure: ALCH-Reasoner Proof for Task 3

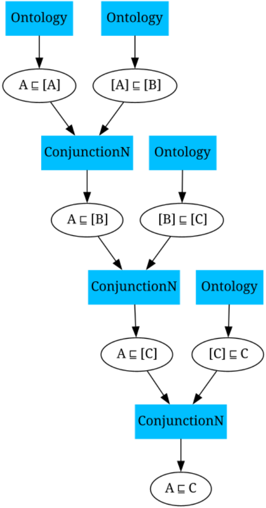

Figure: $\mathcal{O} \coloneqq \{A \sqsubseteq B, B \sqsubseteq C\}$, Goal: $A \sqsubseteq C$

ALCH-Reasoner Performance

Task	Time (ms)	#Axioms	Size largest Premise	#RuleApplications
00001	192	19	3	20
00003	5913	23	4	27
00008	133	15	3	18
00009	119	25	4	27
00012	24	7	2	7

Lethe Performance

Task	Time (ms)	#Axioms	Size largest Premise	#RuleApplications
00001	2786	13	2	13
00003	502	5	2	5
00008	2728	8	2	8
00009	1348	11	2	11
00012	159	2	1	2

Conclusion

- Removal of normalization-caused steps
- Less readable due to normalization
- Fast but can be optimized

Thank you for your attention!