Relevant Parts of Counter Models as Explanations for $\mathcal{EL}$ Non-Subsumptions (Extended Abstract)

Christian Alrabbaa, Willi Hieke, and Anni-Yasmin Turhan

Institute for Theoretical Computer Science
TU Dresden, Dresden, Germany
firstname.lastname@tu-dresden.de

Description logics (DLs) are a family of formal ontology languages that are fragments of first-order logic. The main building blocks of such ontologies are concepts, which are unary predicates, and roles, which are binary predicates. Concepts and roles can be combined to build concept axioms that can describe notions from an application domain. Concept axioms are collected in the TBox. DL ontologies are widely used as knowledge representation formalisms, since they are the basis for W3C standardized ontology languages comprised in the OWL 2 standard. The main asset of DLs is that they come with formally defined and well-investigated reasoning problems. Often, the corresponding reasoning procedures are implemented in highly-optimized reasoner systems [8]. DL TBoxes from practical applications can easily contain more than 10.000s of axioms. If an unexpected consequence holds w.r.t. such a TBox, it is often far from obvious why it holds. In cases where a user with little expertise in logic faces such an unexpected consequence, automated explanation services are needed.

A prominent reasoning problem for DL TBoxes is to decide whether a subsumption relationship holds between two given concepts w.r.t. the TBox, i.e., to decide whether the first concept is a specialization of the second taking all axioms in the TBox into account. Decision procedures for subsumption have been implemented in a range of DL reasoners [8]. In this extended abstract, we consider the setting where the unexpected consequence is a missing subsumption relationship between two concepts w.r.t. an $\mathcal{EL}$ TBox. The DL $\mathcal{EL}$ is computationally well-behaved, as deciding subsumption is tractable [5]. Furthermore, $\mathcal{EL}$ enjoys the canonical model property, which guarantees the existence of a particular model, which can be embedded into every model of the TBox. Reasoning in $\mathcal{EL}$ then amounts to computing the canonical model. In fact, many $\mathcal{EL}$ reasoners implement the computation of the canonical model [2,11,12]. Intuitively, the canonical model has for each concept from the TBox an element that represents this concept. Canonical models for TBoxes are node and edge labeled graphs. This justifies that we view such models as labeled graphs in the following.

To explain negative answers to a subsumption test is a form of explaining non-entailment which can be addressed by supplying counter examples to the expected entailment. Such counter examples could then either be displayed to the user or serve as a starting point for generating more user-friendly explanations, as, for instance, by verbalizing the graph [9]. A counter example for non-subsumption is the canonical model of the TBox itself. As this model contains the whole signature of the TBox, it can easily be too large for explanation purposes. A relevant part of the counter model can be much more comprehensible. We propose in this paper 4 kinds of relevant substructures of canonical models that can serve as (starting points for) explanations.

As a method for extracting relevant substructures from canonical models, we propose to use transductions. In general, transductions specify mappings on relational structures using logical formulae with free variables. Transductions are a versatile approach for graph transformations as they admit the expressivity of monadic second order logic in their formulae. In [10] we have argued that transductions can be suitable for model(-preserving) transformations for DLs. The relevant parts of a counter model can be obtained by transductions that prune the canonical model.

Now, for explaining positive answers to a subsumption test early explanation approaches are based on the computation of justifications, which are minimal axiom subsets of the TBox and that are "responsible" for the subsumption [4,14]. These methods often produce explanations on the granularity of axioms that appear in the KB and are thus to a certain extend syntactic approaches. Another approach to explaining positive subsumption results is to give a proof, i.e., a derivation of the subsumption by the calculus in use. Such methods were investigated for DLs, recently e.g. for $\mathcal{EL}$ in [1]. Proof-based methods often supply a rather procedural view on explanations. Our approach to explaining non-subsumption is more fine-grained than classical justification-based methods in the sense that it can address each consequence of the TBox, since it uses the semantics of the DL KB. Furthermore, unlike proof-based techniques, the outcome of our method is of declarative nature and thus well in line with the declarative formalism of DLs.

Defining Relevant Parts of Counter Models

We assume that the reader is familiar with the basic notions of DLs. These notions are, for instance, described in [3].

Let $\mathcal{T}$ be a TBox and $\phi \coloneqq A \sqsubseteq_{\mathcal{T}} B$ a subsumption query that uses w.l.o.g. named concepts (from $\mathcal{T}$). An interpretation $\mathcal{I}$ is called a *counter model for ϕ w.r.t. $\mathcal{T}$* iff $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \not\models \phi$. In order to be able to supply a succinct explanation of $\mathcal{T} \not\models \phi$, we aim to reduce the amount of information in the counter model in order to provide a concise explanation of the non-subsumption. We exemplify our method by a running example.

Example 1. Clinical differentiation between Parkinson's disease (PD) and progressive supranuclear palsy (PSP) can be challenging due to overlapping clinical features [13]. We model the characteristics of patients with the diseases by the

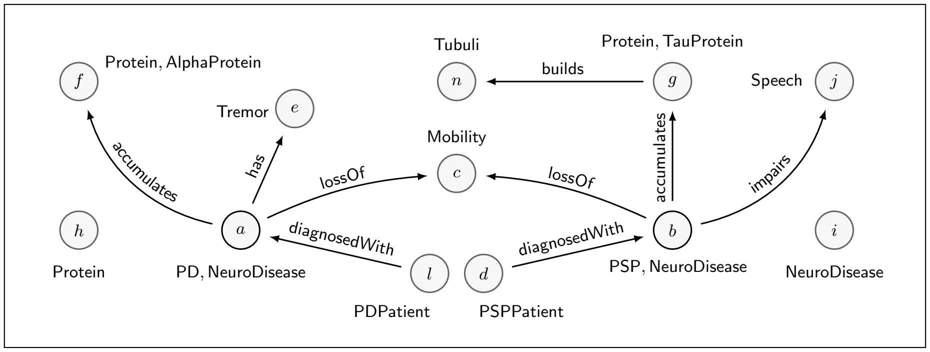


Fig. 1: Model $\mathcal{I}_{ex}$ of TBox $\mathcal{T}_{ex}$.

following TBox.

$$\begin{aligned}\mathcal{T}_{ex} := \Big\{ \qquad \mathsf{PD} &\sqsubseteq \mathsf{NeuroDisease} \sqcap \exists \mathsf{accumulates}.\mathsf{AlphaProtein} \sqcap \\ & \quad \exists \mathsf{lossOf}.\mathsf{Mobility} \sqcap \exists \mathsf{has}.\mathsf{Tremor}, \\ \mathsf{PSP} &\sqsubseteq \mathsf{NeuroDisease} \sqcap \exists \mathsf{accumulates}.\mathsf{TauProtein} \sqcap \\ & \quad \exists \mathsf{lossOf}.\mathsf{Mobility} \sqcap \exists \mathsf{impairs}.\mathsf{Speech}, \\ \mathsf{PDPatient} &\sqsubseteq \exists \mathsf{diagnosedWith}.\mathsf{PD}, \quad \mathsf{TauProtein} \sqsubseteq \mathsf{Protein} \sqcap \exists \mathsf{builds}.\mathsf{Tubuli}, \\ \mathsf{PSPPatient} &\sqsubseteq \exists \mathsf{diagnosedWith}.\mathsf{PSP}, \quad \mathsf{AlphaProtein} \sqsubseteq \mathsf{Protein} \quad \Big\}\end{aligned}$$

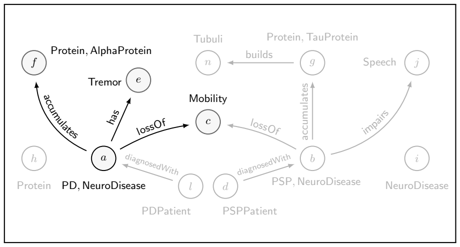
Our example subsumption query is $\phi_{ex} \coloneqq \mathsf{PD} \sqsubseteq_{\mathcal{T}_{ex}} \mathsf{PSP}$ which is not entailed by $\mathcal{T}_{ex}$ and for which we want to supply relevant parts from the canonical model of $\mathcal{T}_{ex}$ as this is our standard counter model. Figure 1 depicts the canonical model $\mathcal{I}_{ex}$ of $\mathcal{T}_{ex}$ using the obvious abbreviations for the names. $\mathcal{I}_{ex}$ contains element a as the representative for the concept PD and b for the concept PSP.

In general, we want to identify relevant substructures of a counter model by requiring that these substructures to be models of sets of implications that follow from the TBox $\mathcal{T}$, and hence, preserving parts of the model. To this end, we define these sets of implications. With $\mathcal{EL}(\mathsf{sig}(\mathcal{T}))$ denoting $\mathcal{EL}$ concepts written in the signature of $\mathcal{T}$, we define $\mathsf{Subsumers}_{\mathcal{T}}(C) \coloneqq \{H \mid C \sqsubseteq_{\mathcal{T}} H, H \in \mathcal{EL}(\mathsf{sig}(\mathcal{T}))\}$.

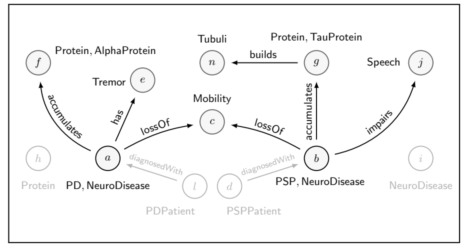
Definition 2 (Relevant Implication Sets). *Let $\mathcal{T}$ be a TBox and $A, B \in \mathsf{sig}(\mathcal{T})$ be concept names. The* relevant implication sets *of $\mathcal{T}$ w.r.t. A and B are:*

$$\begin{aligned}\mathrm{S}_{\mathcal{T}}(A) &\coloneqq \{A \sqsubseteq H \mid H \in \mathit{Subsumers}_{\mathcal{T}}(A)\}, \\ \mathrm{C}_{\mathcal{T}}(A,B) &\coloneqq \{G \sqsubseteq H \mid H \in \mathit{Subsumers}_{\mathcal{T}}(A) \cap \mathit{Subsumers}_{\mathcal{T}}(B), G \in \{A,B\}\}, \\ \bar{\mathrm{S}}_{\mathcal{T}}(A,B) &\coloneqq \{B \sqsubseteq H \mid H \in \mathit{Subsumers}_{\mathcal{T}}(B), H[{}^{M}/{}_{\top}, {}^{\exists r.\top}/{}_{\top}] \in \mathit{Subsumers}_{\mathcal{T}}(A) \\ & \qquad \textit{for all concept names } M \textit{ and all role names } r \textit{ in } \mathsf{sig}(H)\}.\end{aligned}$$

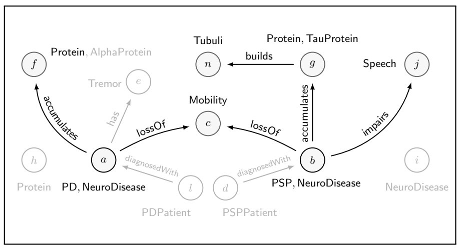
Intuitively, $\mathrm{S}_{\mathcal{T}}(A)$ contains all implications that preserve the information from $\mathcal{T}$ on the instances of A. The set $\mathrm{C}_{\mathcal{T}}(A,B)$ contains all implications that preserve



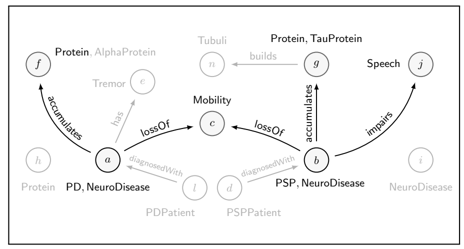
(a) exemplify-A-relevant part



(b) exemplify-A&B-relevant part



(c) diff-relevant part



(d) flat-diff-relevant part

Fig. 2: The different kinds of relevant parts of the canonical model $\mathcal{I}_{ex}$ of $\mathcal{T}_{ex}$.

for A and for B the information on the commonalities of A and B from $\mathcal{T}$. The set $\bar{\mathrm{S}}_{\mathcal{T}}(A, B)$ contains all implications that preserve for B some commonalities of A and B that follow from $\mathcal{T}$. These commonalities are restricted to those subsumers of A that remain subsumers, if all concept names are removed from them and the role-depth of each nested existential restriction is reduced by 1.

Definition 3 (Relevant Parts of Counter Models). *Let $\mathcal{T}$ be a TBox, $\phi \coloneqq A \sqsubseteq_{\mathcal{T}} B$ a subsumption query, $\mathcal{I}$ a counter model of ϕ w.r.t. $\mathcal{T}$. Let $a, b \in \Delta^{\mathcal{I}}$ s.t. $a \in A^{\mathcal{I}} \setminus B^{\mathcal{I}}$, and $b \in B^{\mathcal{I}}$ if $B^{\mathcal{I}} \neq \emptyset$. An interpretation $\mathcal{I}'$ is called* {*exemplify-A*, *exemplify-A&B*, *diff*, *flat-diff*}-relevant part of $\mathcal{I}$ w.r.t. ϕ and $\mathcal{T}$ *iff $\mathcal{I}'$ is one of the smallest substructures of $\mathcal{I}$, s.t. $\mathcal{I}' \not\models \phi$, and either*

- $a \in A^{\mathcal{I}'}$ *and* $\mathcal{I}' \models \mathrm{S}_{\mathcal{T}}(A)$; (exemplify-A)
- $a \in A^{\mathcal{I}'}, b \in B^{\mathcal{I}'}$ *and* $\mathcal{I}' \models \mathrm{S}_{\mathcal{T}}(A) \cup \mathrm{S}_{\mathcal{T}}(B)$; (exemplify-A&B)
- $a \in A^{\mathcal{I}'}, b \in B^{\mathcal{I}'}$ *and* $\mathcal{I}' \models \mathrm{C}_{\mathcal{T}}(A, B) \cup \mathrm{S}_{\mathcal{T}}(B)$; (diff)
- $a \in A^{\mathcal{I}'}, b \in B^{\mathcal{I}'}$ *and* $\mathcal{I}' \models \mathrm{C}_{\mathcal{T}}(A, B) \cup \bar{\mathrm{S}}_{\mathcal{T}}(A, B)$. (flat-diff)

We explain the intuition and the purpose of the four kinds of relevant parts and refer to Figure 2, where the corresponding parts of the canonical model $\mathcal{I}_{ex}$ from our running example are depicted. The exemplify-A-relevant part illustrates the (implicit) information $\mathcal{T}$ has on A and thus can be used to display a "full" example for the conditions for instances of A from the subsumption query. Figure 2a displays this part of $\mathcal{I}_{ex}$ and concept PD.

The exemplify-A&B-part follows the same idea, but does so for both concepts from the subsumption query. Thus the "full" information from both query concepts can get displayed to the user. The corresponding part from our example is displayed in Figure 2b. Now, these two kinds of relevant parts give, in some sense, the full descriptions of the involved concepts.

Explaining non-entailment, usually is regarded as a kind of abduction problem. In our case of non-subsumption, the corresponding abduction problem would need to infer what A lacks to become a B. This consideration motivates the other kinds of relevant parts that we suggest. Intuitively, a diff-relevant part demonstrates the difference of A and B and does so by preserving the information on the commonalities of both concepts at a and gives full information on B at b. Thereby it highlights which parts of B are not entailed for A. The diff-relevant part in our running example is displayed in Figure 2c. The $\mathsf{flat\text{-}diff}$-relevant part follows the same idea, but illustrates a flattened form of difference as it preserves only those parts from B up to the smallest depth where a difference to A occurs. The $\mathsf{flat\text{-}diff}$-relevant part in our running example is displayed in Figure 2d. It prunes the relational structure of b in comparison to the diff-relevant part. This is depicted in Figure 2d by only showing that b accumulates a tau protein and omitting that tau proteins build tubuli. Generally, a $\mathsf{flat\text{-}diff}$-relevant part is more succinct in highlighting the differences between A and B, than a diff-relevant part.

Towards Extracting Relevant Parts of Counter Models

In order to obtain the relevant part of a counter models, we use monadic second-order (MSO) transductions as a formalism to describe model transformations [7], which are tailored to DL interpretations in [10]. We consider transductions since they are well-defined and such transductions are computable since MSO model checking is PSPACE-complete [15]. Furthermore, several transductions could easily be combined and be applied after one another like procedures to obtain the relevant parts. One can devise four transductions, one for each kind of the relevant parts, and prove their correctness for canonical models of general $\mathcal{EL}$ TBoxes. Since the canonical models of $\mathcal{EL}$ TBoxes are always finite, transductions can be applied. To make the transductions precise is part of our future work.

Conclusions and Future Work

We have introduced several notions of informative relevant parts from counter examples for explaining non-subsumptions of $\mathcal{EL}$ TBoxes. We are currently implementing a system for providing explanations of $\mathcal{EL}$ non-subsumptions based on our kinds of relevant parts, to be evaluated on application ontologies. An obvious extension to this work is to extend our approach to DL knowledge bases also containing data and to more expressive logics that also have the canonical model property—such as Horn DLs or even some types of existential rules. We would also like to consider different types of reasoning tasks such as explaining missing answers to conjunctive queries—as it was done for DL-Lite in [6].

Acknowledgements This work was supported by the DFG in grant 389792660 as part of TRR 248 (https://perspicuous-computing.science), and QuantLA, GRK 1763 (https://lat.inf.tu-dresden.de/quantla).

References

1. Alrabbaa, C., Baader, F., Borgwardt, S., Koopmann, P., Kovtunova, A.: Finding small proofs for description logic entailments: Theory and practice. In: Albert, E., Kovacs, L. (eds.) LPAR-23: 23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning. EPiC Series in Computing, vol. 73, pp. 32–67. EasyChair (2020)
2. Baader, F., Lutz, C., Suntisrivaraporn, B.: CEL — a polynomial-time reasoner for life science ontologies. In: Furbach, U., Shankar, N. (eds.) Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR-06). Lecture Notes in Artificial Intelligence, vol. 4130, pp. 287–291. Springer-Verlag (2006)
3. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
4. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic $\mathcal{EL}^{+}$. In: Hertzberg, J., Beetz, M., Englert, R. (eds.) Proceedings of the 30th German Conference on Artificial Intelligence (KI-07). Lecture Notes in Artificial Intelligence, vol. 4667, pp. 52–67. Springer-Verlag (2007)
5. Brandt, S.: Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In: de Mantáras, R.L., Saitta, L. (eds.) Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-2004). pp. 298–302. IOS Press (2004)
6. Calvanese, D., Ortiz, M., Simkus, M., Stefanoni, G.: Reasoning about explanations for negative query answers in DL-lite. Journal of Artificial Intelligence Research **48**, 635–669 (2013)
7. Courcelle, B., Engelfriet, J.: Graph Structure and Monadic Second-Order Logic - A Language-Theoretic Approach. Cambridge University Press (2012)
8. Dentler, K., Cornet, R., ten Teije, A., de Keizer, N.: Comparison of reasoners for large ontologies in the OWL 2 $\mathcal{EL}$ profile. Semantic Web **2**(2), 71–87 (2011)
9. Ell, B., Harth, A.: A language-independent method for the extraction of RDF verbalization templates. In: Mitchell, M., McCoy, K.F., McDonald, D., Cahill, A. (eds.) INLG 2014 - Proceedings of the Eighth International Natural Language Generation Conference, Including Proceedings of the INLG and SIGDIAL. pp. 26–34. The Association for Computer Linguistics (2014)
10. Hieke, W., Turhan, A.: Towards model transformation in description logics — investigating the case of transductions. In: Beierle, C., Ragni, M., Stolzenburg, F., Thimm, M. (eds.) Proceedings of the 6th Workshop on Formal and Cognitive Reasoning (FCR) co-located with 43rd German Conference on Artificial Intelligence (KI-20). CEUR, vol. 2680, pp. 69–82. CEUR-WS.org (2020)
11. Kazakov, Y., Krötzsch, M., Simancik, F.: The incredible ELK — from polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies. Journal of Automated Reasoning **53**(1), 1–61 (2014)
12. Lawley, M.J., Bousquet, C.: Fast classification in protégé: Snorocket as an owl 2 el reasoner. In: Taylor, K., Meyer, T., Orgun, M. (eds.) Proceedings of the 6th Australasian Ontology Workshop (IAOA-10). Conferences in Research and Practice in Information Technology. vol. 122, pp. 45–49. Australian Computer Society Inc. (2010)
13. Lee, Y.E.C., Williams, D.R., Anderson, J.F.: Frontal deficits differentiate progressive supranuclear palsy from parkinson's disease. Journal of Neuropsychology **10**(1), 1–14 (2016)

14. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Gottlob, G., Walsh, T. (eds.) Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03). pp. 355–362. Morgan Kaufmann (2003)
15. Vardi, M.Y.: The complexity of relational query languages. In: Lewis, H.R., Simons, B.B., Burkhard, W.A., Landweber, L.H. (eds.) Proceedings of the 14th Annual ACM Symposium on Theory of Computing. pp. 137–146. ACM (1982)