Explaining Non-Entailment by Model Transformation for the Description Logic $\mathcal{EL}$

Christian Alrabbaa
christian.alrabbaa@tu-dresden.de
Technische Universität Dresden
Dresden, Germany

Willi Hieke
willi.hieke@tu-dresden.de
Technische Universität Dresden
Dresden, Germany

ABSTRACT

Reasoning results computed by description logic systems can be hard to comprehend. When an ontology does not entail an expected subsumption relationship, generating an explanation of this non-entailment becomes necessary. In this paper, we use countermodels to explain non-entailments. More precisely, we devise relevant parts of canonical models of $\mathcal{EL}$ ontologies that serve as explanations and discuss the computational complexity of extracting these parts by means of model transformations. Furthermore, we provide an implementation of these transformations and evaluate it using real ontologies.

CCS CONCEPTS

• **Computing methodologies → Description logics**.

KEYWORDS

Description Logics, Explainable AI, Model Transformation

ACM Reference Format:
Christian Alrabbaa and Willi Hieke. 2022. Explaining Non-Entailment by Model Transformation for the Description Logic $\mathcal{EL}$. In *Proceedings of 11th International Joint Conference On Knowledge Graphs (IJCKG).* ACM, New York, NY, USA, 9 pages.

1 INTRODUCTION

DLs are formal knowledge representation languages that formalize knowledge in terms of concepts and relations between them. DL systems possess knowledge in form of sets containing logical statements about a specific domain of interest, e.g. medical sciences. These sets are usually referred to as ontologies or TBoxes[1]. A DL TBox typically speaks about subsumption relationships of concepts, which are defined as unary predicates using the specific DL of the TBox. In this paper, we consider the DL $\mathcal{EL}$— a language with existential quantification and conjunction. DLs are widely used for knowledge representation as they constitute the foundation of W3C standardized ontologies comprised in the OWL 2 standard.

[1]TBox abbreviates terminological knowledge.

IJCKG, October 27–29, 2022, Hangzhou, China

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00

The logical statements in $\mathcal{EL}$ TBoxes are assertions about subsumption relationships between two $\mathcal{EL}$ concepts. For instance, a sentence of an $\mathcal{EL}$ TBox could state that "every Cat is a Mammal", where Cat and Mammal are $\mathcal{EL}$ concepts of the respective TBox. DL systems are used to perform various reasoning services using the TBox knowledge. Standard reasoning problems for DLs are well-investigated and often implemented in highly optimized reasoner systems [14]. However, these reasoning results can be hard to comprehend for users of the DL systems, which is the reason for the development of various syntax as well as semantics-based approaches to explaining logic-based AI reasoning [1, 7, 8, 11, 21, 26]. A common reasoning problem for $\mathcal{EL}$ TBoxes is to decide whether a concept is subsumed by another one w.r.t. the given TBox. Sometimes, a user might face the situation in which the subsumption relationship cannot be deduced from the TBox — and the reason for this non-entailment can be far from obvious.

Non-entailments can be explained by using countermodels of the subsumption relationship in question w.r.t. the TBox in use. Now, not every countermodel is suitable for the purpose of explanation since these models might contain a lot of irrelevant information as the TBox can easily consist of tens of thousands of statements. In order to provide concise explanations to users of $\mathcal{EL}$ systems, we define logically relevant parts of countermodels for the user to understand the non-entailment. More precisely, we provide four notions of relevant parts of countermodels, which are built upon each other. We then define model transformations to extract these relevant parts from canonical models of $\mathcal{EL}$ TBoxes — these are canonically constructed models that every $\mathcal{EL}$ TBox admits. In fact, many $\mathcal{EL}$ reasoners implement the computation of canonical models [6, 20, 22]. As formalism for extracting relevant substructures from canonical models, we use graph transductions [13]. In general, transductions specify mappings over graph (or higher arity) structures using logical formulae with free variables.

In Section 3, we provide the definitions of the relevant parts of countermodels and explain the idea using an example from the medical sciences. In Section 4, we define the model transformations, show their soundness w.r.t. the definition of relevant parts of countermodels using canonical models as input, and discuss their computational complexity. In Section 5 then, we present an implementation of these transformations paired with a run time evaluation using TBoxes from practical applications. The definitions of relevant parts of countermodels and first models transformations have been introduced in [3]. This paper extends [3] by adding further model transformations and by showing complexity results for the defined transductions as well as by providing an evaluated implementation.

2 PRELIMINARIES

DLs are decidable knowledge representation languages that can model structures over unary and binary predicates. Unary predicates are called *concepts* and binary predicates are called *roles*. Concepts for the description logic $\mathcal{EL}$ are built inductively from a set of concept names $\mathsf{N_C}$ and a set of role names $\mathsf{N_R}$. Let $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$, then, an $\mathcal{EL}$ concept C is constructed by the syntax:

$$C ::= A \mid C \sqcap C \mid \exists r.C \mid \top.$$

We assume that the sets of concept names and role names are disjoint and we denote concepts by upper case and roles with lower case letters. A signature Σ is the union of two finite sets $\Sigma_\mathsf{C} \subset \mathsf{N_C}$ and $\Sigma_\mathsf{R} \subset \mathsf{N_R}$. By $\mathcal{EL}(\Sigma)$ we denote the set of $\mathcal{EL}$ concepts using only signature symbols from Σ. An interpretation $\mathcal{I} = (\mathcal{D}^\mathcal{I}, \cdot^\mathcal{I})$ over a signature Σ consists of a non-empty set $\mathcal{D}^\mathcal{I}$ called the *interpretation domain* and an *interpretation function* $\cdot^\mathcal{I}$ that maps every concept name in Σ_C, written $A^\mathcal{I}$ for a concept name A, to a subset of $\mathcal{D}^\mathcal{I}$ and every role name in Σ_R, written $r^\mathcal{I}$ for a role name r, to a subset of $\mathcal{D}^\mathcal{I} \times \mathcal{D}^\mathcal{I}$. The mapping $\cdot^\mathcal{I}$ extends to concepts as follows:

- $(C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$,
- $(\exists r.C)^\mathcal{I} = \{d \in \mathcal{D}^\mathcal{I} \mid \text{there is an } e \in C^\mathcal{I} \ s.t. \ (d,e) \in r^\mathcal{I}\}$,
- and $\top^\mathcal{I} = \mathcal{D}^\mathcal{I}$.

Knowledge for DL systems is stored as logical formulae in so called TBoxes (terminological knowledge), which sometimes are referred to as ontologies. An $\mathcal{EL}$ *TBox* $\mathcal{T}$ is a finite set of *concept inclusions (CIs)*, which are formulae of the form $C \sqsubseteq D$, where C and D are $\mathcal{EL}$ concepts. We abbreviate $C \sqsubseteq D$ and $D \sqsubseteq C$ by $C \equiv D$. An interpretation $\mathcal{I}$ satisfies a CI $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$, and $\mathcal{I}$ is called *model* of a TBox $\mathcal{T}$ if $\mathcal{I}$ satisfies all the CIs in $\mathcal{T}$. We write $(\mathcal{I}, a) \models H$ to express that $a \in \mathcal{D}^\mathcal{I}$ satisfies H, and $(\mathcal{I}, a) \models \Gamma$ if $(\mathcal{I}, a) \models H$ for each H in a set of concepts Γ. For a TBox, an interpretation or a concept X, we denote its *signature* by $\mathrm{sig}(X)$. To denote the *concept signature* of X, we write $\mathrm{sig_C}(X) = \mathrm{sig}(X) \cap \mathsf{N_C}$, and $\mathrm{sig_R}(X) = \mathrm{sig}(X) \cap \mathsf{N_R}$ for the *role signature*. A prominent reasoning problem for DLs is to decide subsumption. Given two concepts C and D and a TBox $\mathcal{T}$, *subsumption* (denoted by $\mathcal{T} \models C \sqsubseteq D$) decides whether for each model $\mathcal{I}$ of $\mathcal{T}$, it is the case that $C^\mathcal{I} \subseteq D^\mathcal{I}$. The main method for deciding subsumption is to compute the canonical model of an $\mathcal{EL}$ TBox [9]. To compute the canonical model, the respective TBox needs to be normalized first. A TBox is in *normal form* if and only if it only contains CIs of the forms:

$$A \sqsubseteq B, \qquad A_1 \sqcap A_2 \sqsubseteq B, \qquad \exists r.A \sqsubseteq B, \qquad or \qquad A \sqsubseteq \exists r.B,$$

where A, A_1, A_2, and B are concept names or $\top$, and r is a role name. Every TBox $\mathcal{T}$ can be transformed into a TBox $\mathcal{T}'$ in normal form, s.t. the size of $\mathcal{T}'$ is linear in the size of $\mathcal{T}$ and every model of $\mathcal{T}'$ is a model of $\mathcal{T}$ [4]. Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ TBox. In [4], the *canonical model* $\mathcal{I}_\mathcal{T}$ of $\mathcal{T}$ is defined as follows:

- $\mathcal{D}^{\mathcal{I}_\mathcal{T}} := \{A \mid A \in \mathrm{sig_C}(\mathcal{T})\} \cup \{\top\}$,
- $A^{\mathcal{I}_\mathcal{T}} := \{B \in \mathcal{D}^{\mathcal{I}_\mathcal{T}} \mid \mathcal{T} \models B \sqsubseteq A\}$ f.a. (for all) $A \in \Sigma_\mathsf{C}$,
- $r^{\mathcal{I}_\mathcal{T}} := \{(A, B) \in \mathcal{D}^{\mathcal{I}_\mathcal{T}} \times \mathcal{D}^{\mathcal{I}_\mathcal{T}} \mid \mathcal{T} \models A \sqsubseteq \exists r.B\}$ f.a. $r \in \Sigma_\mathsf{R}$,

We use of the following convenient property of canonical models.

Theorem 2.1 ([4]). *For any normalized $\mathcal{EL}$ TBox $\mathcal{T}$ and two concept names A and B, we have $\mathcal{T} \models A \sqsubseteq B$ if and only if $\mathcal{I}_\mathcal{T} \models A \sqsubseteq B$.*

Subsumption of arbitrary concepts C and D can be tested, if the CIs $A \equiv C$ and $B \equiv D$ are added to $\mathcal{T}$. We will also use simulation relations, which can e.g. be found in [24]. Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be two interpretations and let Σ be a signature. A relation $\sim_\Sigma \subseteq \mathcal{D}^{\mathcal{I}_1} \times \mathcal{D}^{\mathcal{I}_2}$ is called Σ-simulation from $\mathcal{I}_1$ to $\mathcal{I}_2$ if

- $d_1 \sim_\Sigma d_2$ and $d_1 \in N^{\mathcal{I}_1}$ implies that $d_2 \in N^{\mathcal{I}_2}$, f.a. $N \in \Sigma_\mathsf{C}$,
- $d_1 \sim_\Sigma d_2$ and $(d_1, e_1) \in r^{\mathcal{I}_1}$ implies $(d_2, e_2) \in r^{\mathcal{I}_2}$ and $e_1 \sim_\Sigma e_2$ for some e_2, f.a. $r \in \Sigma_\mathsf{R}$.

To express that there is a Σ-simulation between $\mathcal{I}_1$ and $\mathcal{I}_2$ with $(d_1, d_2) \in \sim_\Sigma$, we write $(\mathcal{I}_1, d_1) \sim_\Sigma (\mathcal{I}_2, d_2)$. Simulation relations can be used to characterize elements of two interpretations w.r.t. their satisfaction of $\mathcal{EL}$ concepts. We omit writing Σ as an index to $\sim$ as it is clear from the context.

Theorem 2.2 ([12]). *Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be two finite interpretations over signature Σ with $d_1 \in \mathcal{D}^{\mathcal{I}_1}$ and $d_2 \in \mathcal{D}^{\mathcal{I}_2}$. Then, $(\mathcal{I}_1, d_1) \sim (\mathcal{I}_2, d_2)$ if and only if $(\mathcal{I}_1, d_1) \models H$ implies $(\mathcal{I}_2, d_2) \models H$ for all $H \in \mathcal{EL}(\Sigma)$.*

Model transformations are binary (usually functional) relations on the class of finite interpretations. We use second-order (SO) transductions as formalism to describe model transformations. Graph transductions are defined in [13] and tailored to description logic interpretations in [16]. Intuitively, a transduction specifies a model transformation by a tuple of SO formulae, called *definition scheme*, that describes how to construct an output interpretation in terms of the input interpretation. By $\mathrm{SO}(\Sigma, \mathcal{W})$ we denote the set of SO formulae with free first-order variables in $\mathcal{W}$. These variables are called *parameters*. Let Σ be a binary signature, and let $\mathcal{W}$ be a finite set of *parameters*. An *SO definition scheme* is a tuple

$$\mathsf{D} = \langle \chi, \delta, (\theta_N)_{N \in \Sigma_\mathsf{C}}, (\eta_r)_{r \in \Sigma_\mathsf{R}} \rangle, \ where$$

- $\chi \in \mathrm{SO}(\Sigma, \mathcal{W})$ (*precondition*),
- $\delta \in \mathrm{SO}(\Sigma, \mathcal{W} \cup \{x\})$ (*domain formula*),
- $\theta_N \in \mathrm{SO}(\Sigma, \mathcal{W} \cup \{x\})$, f.a. $N \in \Sigma_\mathsf{C}$, (*concept formulae*),
- $\eta_r \in \mathrm{SO}(\Sigma, \mathcal{W} \cup \{x, y\})$, f.a. $r \in \Sigma_\mathsf{R}$, (*role formulae*).

First, the *precondition* χ needs to be satisfied by the input interpretation for the transduction to be defined. Then, the *domain formula* δ defines the interpretation domain of the output interpretation.[2] For this domain, the *concept formulae* θ and *role formulae* η define the interpretation function. Let $\mathcal{I}$ be an interpretation over a signature Σ, let $\mathcal{W}$ be a set of parameters, and let λ be a $\mathcal{W}$-assignment in $\mathcal{I}$, i.e., $\lambda : \mathcal{W} \to \mathcal{D}^\mathcal{I}$. A definition scheme D defines $\mathcal{I}'$ from $(\mathcal{I}, \lambda)$ if

- $(\mathcal{I}, \lambda) \models \chi(\mathcal{W})$,
- $\mathcal{D}^{\mathcal{I}'} := \{a \in \mathcal{D}^\mathcal{I} \mid (\mathcal{I}, \lambda) \models \delta(\mathcal{W}, a)\}$,
- $N^{\mathcal{I}'} := \{a \in \mathcal{D}^{\mathcal{I}'} \mid (\mathcal{I}, \lambda) \models \theta_N(\mathcal{W}, a)\}$ f.a. $N \in \Sigma_\mathsf{C}$,
- $r^{\mathcal{I}'} := \{(a, b) \in (\mathcal{D}^{\mathcal{I}'})^2 \mid (\mathcal{I}, \lambda) \models \eta_r(\mathcal{W}, a, b)\}$ f.a. $r \in \Sigma_\mathsf{R}$,

with $(\mathcal{I}, \lambda) \models \delta(\mathcal{W}, a)$ meaning $(\mathcal{I}, \lambda') \models \delta(\mathcal{W}, x)$, where λ' is the assignment extending λ such that $\lambda' : x \mapsto a$ (and accordingly for θ and η). We denote $\widehat{\mathsf{D}}(\mathcal{I}, \lambda) = \mathcal{I}'$. The *transduction* τ induced by D is defined as

$$\tau := \{(\mathcal{I}, \widehat{\mathsf{D}}(\mathcal{I}, \lambda)) \mid \lambda \text{ is a } \mathcal{W}\text{-assignment in } \mathcal{I} \text{ with } (\mathcal{I}, \lambda) \models \chi\},$$

and $\tau(\mathcal{I})$ denotes $\{\widehat{\mathsf{D}}(\mathcal{I}, \lambda) \mid (\mathcal{I}, \lambda) \models \chi \text{ for some } \lambda\}$. For functional transductions, we write $\tau(\mathcal{I}) = \mathcal{I}'$.

[2] We indicate parameter variables from the set $\mathcal{W} = \{z_1, \ldots, z_n\}$ in a formula φ by writing $\varphi(\mathcal{W}, x)$ instead of $\varphi(z_1, \ldots, z_n, x)$.

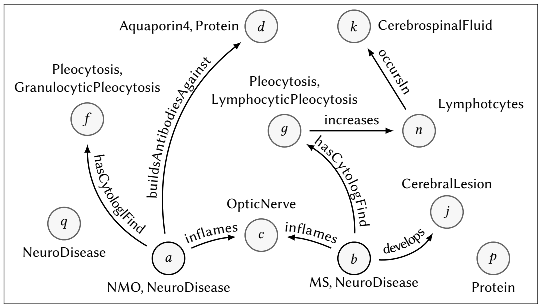

Figure 1: Canonical Model $\mathcal{I}_{\mathcal{T}_{ex}}$

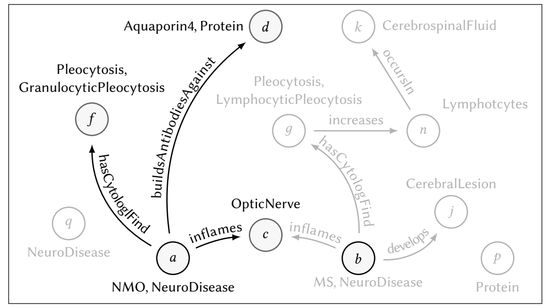

Figure 2: α-relevant Part of $\mathcal{I}_{\mathcal{T}_{ex}}$

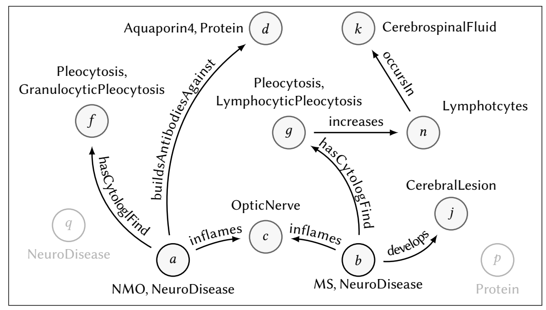

Figure 3: β-relevant Part of $\mathcal{I}_{\mathcal{T}_{ex}}$

3 COUNTERMODELS AS EXPLANATIONS

In this section, we summarize the definitions of relevant parts of countermodels that serve as explanation for non-entailments of CIs with respect to $\mathcal{EL}$ TBoxes [3]. We explain and exemplify each definition by a running example. Asking for the validity of a concept inclusion $C \sqsubseteq D$ w.r.t. $\mathcal{T}$ is called *subsumption query*, denoted by $C \sqsubseteq_{\mathcal{T}} D$. Without loss of generality, we assume that the subsumption queries only use concept names A and B. We call an interpretation $\mathcal{I}$ *countermodel* to $A \sqsubseteq_{\mathcal{T}} B$ if $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \not\models A \sqsubseteq B$. In principle, every countermodel is an explanation for the non-entailment of the subsumption query. However, these models might be unnecessarily large and contain irrelevant parts to the subsumption query. For instance, a medical ontology likely contains knowledge about various diseases that are not related to the diseases from the subsumption query. Our goal is to reduce the amount of domain elements, concept labels, and role labels and thereby provide concise explanations.

Example 3.1. Neuromyelitis optica (NMO) was long considered a subtype of multiple sclerosis (MS), which both are autoimmune, demyelinating diseases of the central nervous system. However, in 2004, it was discovered that NMO is actually a distinct disease [23]. Due to a number of overlapping clinical features, medical differentiation can be challenging, which potentially leads to misdiagnoses [19]. We express some of these characteristics in the TBox

$$
\begin{aligned}
\mathcal{T}_{ex} := \big\{\, & \mathsf{NMO} \sqsubseteq \mathsf{NeuroDisease} \sqcap \exists \mathsf{inflames.OpticNerve} \sqcap \\
& \exists \mathsf{buildsAntibodiesAgainst.Aquaporin4} \sqcap \\
& \exists.\mathsf{hasCytologicalFinding.GranulocyticPleocytosis}\,, \\
& \mathsf{MS} \sqsubseteq \mathsf{NeuroDisease} \sqcap \exists \mathsf{develops.CerebralLesion} \sqcap \\
& \exists \mathsf{hasCytologicalFinding.LymphocyticPleocytosis} \sqcap \\
& \exists \mathsf{inflames.OpticNerve}\,,\ \mathsf{Aquaporin4} \sqsubseteq \mathsf{Protein}\,, \\
& \mathsf{LymphocyticPleocytosis} \sqsubseteq \exists \mathsf{increases.Lymphocytes}\,, \\
& \mathsf{Lymphocytes} \sqsubseteq \exists \mathsf{occursIn.CerebrospinalFluid}\,, \\
& \mathsf{GranulocyticPleocytosis} \sqsubseteq \mathsf{Pleocytosis}\,, \\
& \mathsf{LymphocyticPleocytosis} \sqsubseteq \mathsf{Pleocytosis} \big\}
\end{aligned}
$$

The example subsumption query is $\phi_{ex} := \mathsf{NMO} \sqsubseteq_{\mathcal{T}_{ex}} \mathsf{MS}$, which is not entailed by the TBox $\mathcal{T}_{ex}$. Figure 1 depicts the canonical model $\mathcal{I}_{\mathcal{T}_{ex}}$ of $\mathcal{T}_{ex}$.[3]

[3]We omit displaying the $\top$ element for a better overview.

We identify the relevant substructures of countermodels by demanding these substructures to be smallest models of relevant subsumers. For a concept H, we denote by $H[N/M]$ the syntactic substitution of every occurrence of concept N by concept M in H.

Definition 3.2 (Relevant Subsumer Sets). Let $\mathcal{T}$ be an $\mathcal{EL}$ TBox and $A, B \in \mathrm{sig}(\mathcal{T})$ be concept names. The *relevant subsumer sets of* $\mathcal{T}$ w.r.t. A and B are:

$$\mathrm{S}_{\mathcal{T}}(A) := \{H \mid \mathcal{T} \models A \sqsubseteq H\}, \tag{1}$$

$$\mathrm{C}_{\mathcal{T}}(A,B) := \{H \mid H \in \mathrm{S}_{\mathcal{T}}(A),\ H \in \mathrm{S}_{\mathcal{T}}(B)\}, \tag{2}$$

$$\bar{\mathrm{S}}_{\mathcal{T}}(A,B) := \{H \mid H \in \mathrm{S}_{\mathcal{T}}(B),\ H[N/\top][\exists r.\top/\top] \in \mathrm{S}_{\mathcal{T}}(A) \text{ for all } N \in \mathrm{sig}_{\mathrm{C}}(H) \text{ and all } r \in \mathrm{sig}_{\mathrm{R}}(H)\}. \tag{3}$$

Definition 3.3 (Relevant Parts of Countermodels). Let $\mathcal{T}$ be a TBox with signature Σ, let $\phi := A \sqsubseteq_{\mathcal{T}} B$ be a subsumption query, let $\mathcal{I}$ be a countermodel to ϕ w.r.t. $\mathcal{T}$. An interpretation $\mathcal{I}'$ is called $\{\alpha, \beta, \Delta, \bar{\Delta}\}$*-relevant part of* $\mathcal{I}$ *w.r.t.* ϕ *and* $\mathcal{T}$ if $\mathcal{I}'$ is a smallest substructure of $\mathcal{I}$, s.t. $\mathcal{I}' \not\models \phi$, and there is an

- $a \in A^{\mathcal{I}'}$ and $(\mathcal{I}', a) \models \mathrm{S}_{\mathcal{T}}(A)$. ($\alpha$)
- $a \in A^{\mathcal{I}'}, b \in B^{\mathcal{I}'}$, $(\mathcal{I}', a) \models \mathrm{S}_{\mathcal{T}}(A)$, $(\mathcal{I}', b) \models \mathrm{S}_{\mathcal{T}}(B)$. ($\beta$)
- $a \in A^{\mathcal{I}'}, b \in B^{\mathcal{I}'}$, $(\mathcal{I}', a) \models \mathrm{C}_{\mathcal{T}}(A,B)$, $(\mathcal{I}', b) \models \mathrm{S}_{\mathcal{T}}(B)$. ($\Delta$)
- $a \in A^{\mathcal{I}'}, b \in B^{\mathcal{I}'}$, $(\mathcal{I}', a) \models \mathrm{C}_{\mathcal{T}}(A,B)$, $(\mathcal{I}', b) \models \bar{\mathrm{S}}_{\mathcal{T}}(A,B)$. ($\bar{\Delta}$)

By smallest substructure we mean that any proper substructure of $\mathcal{I}'$ is not a model of the respective relevant subsumer sets. For canonical models, these parts always exist but are not unique up to isomorphism. For $\otimes \in \{\alpha, \beta, \Delta, \bar{\Delta}\}$, we sometimes write "$\otimes$-relevant countermodel", and we write $\otimes$-relevant subsumer set(s) to refer to the respective relevant subsumer sets used in Definition 3.3.

We explain the intuition behind the four relevance notions in reference to the relevant parts of $\mathcal{I}_{\mathcal{T}}$ w.r.t. ϕ_{ex} depicted in Figures 2 to 5. The α-relevant part highlights the information on A w.r.t. $\mathcal{T}$ and

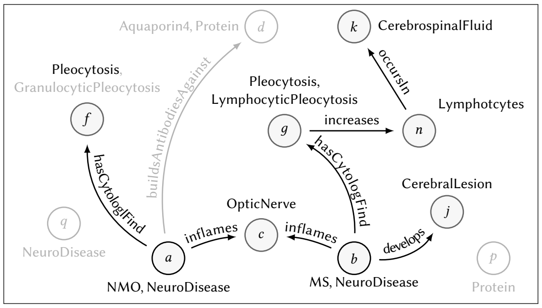

Figure 4: Δ-relevant Part of $\mathcal{I}_{\mathcal{T}_{ex}}$

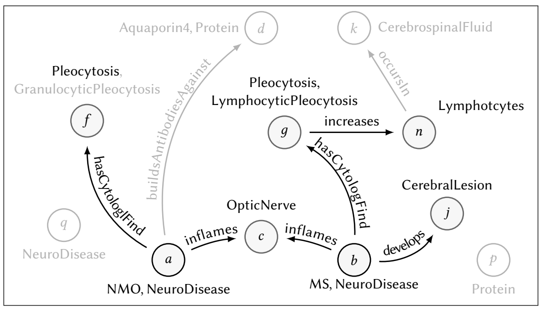

Figure 5: $\bar{\Delta}$-relevant Part of $\mathcal{I}_{\mathcal{T}_{ex}}$

gives an instance of an element that is a basic reason for why the subsumption query does not hold w.r.t. $\mathcal{T}$ – an element that is in the extension of A but not of B. The β-relevant part follows the same idea but for both concepts A and B. This part contains the witness for the non-entailment, element a, and also a representative for B to show the contrast between A and B and thereby explain the non-entailment.

The Δ-relevant part is a refinement of the β-relevant part in the sense that the aforementioned contrast between A and B is reduced to a logically relevant extent. The underlying consideration is the following. If $\mathcal{T} \models A \sqsubseteq B$ was the case, the set of subsumers of A w.r.t. $\mathcal{T}$ would be a superset of the set of subsumers imposed on B by $\mathcal{T}$. Loosely speaking, to explain why $\mathcal{T} \not\models A \sqsubseteq B$, we display only subsumers of B that are not subsumers of A to illustrate what A elements are missing in order to be B elements. In terms of the running example, to explain why NMO is not MS, the fact that NMO comes with antibodies against aquaporin 4 does not make a difference. But MS developing cerebral lesions, which NMO does not, is in fact a reason for the non-entailment and is hence part of the Δ-relevant countermodel.

The $\bar{\Delta}$-relevant part shortens the Δ-relevant part by illustrating a flattened from of difference. This is best explained with the running example. In the Δ-relevant part, MS comes with the cytological finding lymphocytic pleocytosis, which increases the lymphotcytes that occur in a cerebrospinal fluid. Now, the $\bar{\Delta}$-relevant part cuts the information that lymphotcytes that occur in a cerebrospinal fluid because the increase of lymphotcytes alone shows already a sufficient difference to understand the non-entailment. More generally, the difference is flat in the sense that the role depth is restricted to only going one step further in the B-reachable part compared to A-reachable part.

4 COMPUTING MODEL TRANSFORMATIONS

The four kinds of relevant parts of canonical models can be obtained by model transformation. Canonical models of normalized $\mathcal{EL}$ TBoxes are finite and computable in polynomial time [4]. Note that, by Theorem 2.1, canonical models satisfy $A \sqsubseteq B$ if and only if the subsumption query $A \sqsubseteq_{\mathcal{T}} B$ is entailed by $\mathcal{T}$. Thus, canonical models are countermodel to negative subsumption queries by default. For each relevance type, we introduce a separate transduction. In the first step of the extraction process, suitable representatives of the concepts of the subsumption query are selected. An element $a \in A^{\mathcal{I}_{\mathcal{T}}}$ is called *representative of* A in $\mathcal{I}_{\mathcal{T}}$ if and only if for all other $x \in A^{\mathcal{I}_{\mathcal{T}}}$ there is no concept name A', s.t. $x \in A'^{\mathcal{I}_{\mathcal{T}}}$ and $a \notin A'^{\mathcal{I}_{\mathcal{T}}}$ – and likewise for $b \in B^{\mathcal{I}_{\mathcal{T}}}$.

In order to devise the transductions, we define auxiliary predicates that are used in the definition schemes. The predicate for the representative of a concept name A in a canonical model is defined by $rep_A(x) :=$

$$A(x) \land (\forall y : y \neq x \land A(y) \to (\bigwedge_{N \in \Sigma_C} N(y) \to N(x))) \qquad (4)$$

We denote the respective A-representative and B-representative form the domain of a canonical model by a and by b. Note that $rep_A(x) \to A(x) \land \neg B(x)$ for canonical models that do not satisfy the subsumption query. Furthermore, we define $succ(x, y) := (\bigvee_{r \in \Sigma_R} r(x, y)) \lor x = y$. We then define reachability of two elements in an interpretation, denoted by $reach(x, y)$, as the reflexive and transitive closure of the $succ$ relation. An according MSO formula for the predicate $reach$ is given in [13].

4.1 Coarse Relevant Parts of Canonical Models

We next introduce the definition schemes D_α to $\mathsf{D}_{\bar{\Delta}}$ that induce the transductions τ_α to $\tau_{\bar{\Delta}}$ [3]. These transductions cut down the canonical model to its *coarse relevant part*, which only contains information about the concepts in question. These parts, however, need not be minimal as required in Definition 3.3. For this reason, we introduce another transduction $\tau_{\min}$ in Section 4.2 that is composed with τ_α to $\tau_{\bar{\Delta}}$ for minimizing the coarse relevant part.

Let $\mathcal{W} := \{u\}$. The definition scheme D_α for τ_α consists of:

$$\chi(\mathcal{W}) := rep_A(u), \qquad (5) \qquad\qquad \theta_N(\mathcal{W}, x) := N(x), \qquad (7)$$

$$\delta(\mathcal{W}, x) := reach(u, x), \qquad (6) \qquad\qquad \eta_r(\mathcal{W}, x, y) := r(x, y). \qquad (8)$$

The transduction selects the representative of A as the witness for the non-subsumption $A \not\sqsubseteq B$ from $\mathcal{I}_{\mathcal{T}}$ by (5) and collects all the reachable successors of it by (6) to induce a substructure of $\mathcal{I}_{\mathcal{T}}$ w.r.t. $\mathrm{sig}(\mathcal{I}_{\mathcal{T}})$ by (7) and (8).

To construct D_β, we modify D_α. Let now $\mathcal{W} = \{u, v\}$. We use

$$\chi(\mathcal{W}) := rep_A(u) \land rep_B(v), \qquad (9)$$

$$\delta(\mathcal{W}, x) := reach(u, x) \lor reach(v, x), \qquad (10)$$

and (7) and (8) as D_β. Hence, τ_α and τ_β require only reachability checks. As mentioned earlier, the normalization and computation of the canonical model of an $\mathcal{EL}$ TBox can be done in polynomial time [5, 9], and these models always exist for $\mathcal{EL}$ TBoxes. The single relevant parts always exist for canonical models.

Lemma 4.1 (Soundness for τ_α and τ_β [3]). *Let $\phi := A \sqsubseteq_{\mathcal{T}} B$, s.t. $\mathcal{T} \not\models \phi$, let $\mathcal{I}_{\mathcal{T}}$ be the canonical model of $\mathcal{T}$, and let $\otimes \in \{\alpha, \beta\}$. We have that $\tau_\otimes(\mathcal{I}_{\mathcal{T}})$ is a model of the $\otimes$-relevant subsumer set(s).*

Proof sketch. We have $(\tau_\alpha(\mathcal{I}_{\mathcal{T}}), a) \models \mathsf{S}_{\mathcal{T}}(A)$ because $\mathcal{I}_{\mathcal{T}} \models A \sqsubseteq H$ for all $H \in \mathsf{S}_{\mathcal{T}}(A)$ by Theorem 2.1, in particular for the A representative a. Since the representative(s) of A (and B) always exists in $\mathcal{I}_{\mathcal{T}}$ if $\mathcal{I}_{\mathcal{T}} \not\models A \sqsubseteq_{\mathcal{T}} B$, there is an assignment λ with $(\mathcal{I}_{\mathcal{T}}, \lambda) \models \chi(\mathcal{W})$. Since τ_α induces the reachable substructure of a from $\mathcal{I}_{\mathcal{T}}$, every concept is satisfied at a as in $\mathcal{I}_{\mathcal{T}}$. The same argumentation holds for τ_β w.r.t. the B representative b. □

The complexity of a transduction is determined by the highest complexity of evaluating the definitions scheme formulae over the given input. All complexity results given are expressed in the size of the input $\mathcal{I}_{\mathcal{T}}$.

Theorem 4.2. *Computing τ_α and τ_β is NL-complete.*

The most expensive task for computing τ_α and τ_β in their definition schemes is to evaluate the predicate *reach* over the input structure, which is NL-complete for directed graphs [18].

Extracting the Δ-relevant part of a canonical model is different from the previous two cases. It is required that the representative of concept A satisfies the subsumers of A that are subsumers of B as well; whereas the representative of concept B must satisfy all subsumers of B w.r.t. $\mathcal{T}$. We make use of additional auxiliary predicates. The predicate σ is true for two sets of elements if they constitute paths made of the same roles in $\mathcal{I}$ once starting from the A-representative a and once starting from the B-representative b. We denote the occurrence of free second-order variables X in a formula φ by box brackets in $\varphi[X]$ and we denote second-order binary relation variables by small bold letters.

$$\sigma[X, Y, a, b] := \exists \mathbf{h} : \varphi[\mathbf{h}, X, Y, a, b] \wedge \psi[\mathbf{h}], \text{ where} \quad (11)$$

$$\psi[\mathbf{h}] := \forall w, x, y, z : \mathbf{h}(x, w) \wedge \mathbf{h}(y, z) \to \bigvee_{r \in \Sigma_{\mathsf{R}}} r(x, y) \wedge r(w, z) \quad (12)$$

and the formula $\varphi[\mathbf{h}, X, Y, a, b]$ defines $\mathbf{h}$ as a surjective map from X to Y with $\mathbf{h}(a, b)$.[4] To express that the elements in a set $X \subseteq \mathcal{D}^{\mathcal{I}}$ in an interpretation $\mathcal{I}$ constitute a path from one element to another using the *succ* relation, we use a monadic second-order (MSO) formula $\pi[X, a, b]$ defined in [13, Proposition 5.11]. We combine these formulae in the predicate

$$sim(a, b, x, y) := \exists X, Y : \sigma[Y, X, a, b] \wedge \pi[X, a, x] \wedge \pi[Y, b, y]. \quad (13)$$

Hence, two elements x and y satisfy $sim(a, b, x, y)$ if and only if x is reachable from a over a path, s.t. there is a same path from b to y. Note that σ contains a quantification over a binary relation $\mathbf{h}$ and the definition schemes using this formula do not induce MSO transductions but rather second-order transductions. However, since the definition scheme formulae are evaluated over finite interpretations, σ is still decidable and thus the transduction computable. The definition scheme D_Δ inducing the transduction τ_Δ consists of the formulae:

$$\chi(\mathcal{W}) := rep_A(u) \wedge rep_B(v), \quad (14)$$

$$\delta(\mathcal{W}, x) := \exists y : sim(u, v, x, y) \vee reach(v, x) \quad (15)$$

[4] We reuse the names of the elements a and b as variables.

as well as the concept name formulae $\theta_N(\mathcal{W}, x) :=$

$$(\exists y : sim(u, v, x, y) \wedge N(x) \wedge N(y)) \vee (reach(v, x) \wedge N(x)) \quad (16)$$

with N as a concept name variable in θ_N that ranges over $\Sigma_{\mathsf{C}} \setminus \{A\}$. In addition, θ_A is defined as θ_N with the further disjunct $x = u$. The role formulae are defined as $\eta_r(\mathcal{W}, x, y) :=$

$$((\exists w, z : sim(u, v, x, w) \wedge r(w, z)) \vee reach(v, x)) \wedge r(x, y) \quad (17)$$

where the role variable r ranges over Σ_{R}. The definition scheme $\mathsf{D}_{\bar{\Delta}}$ is D_Δ with a modified domain formula $\delta(\mathcal{W}, x) :=$

$$(\exists y : sim(u, v, x, y)) \vee (\exists y, z : sim(u, v, y, z) \wedge succ(z, x)). \quad (18)$$

Lemma 4.3 (Soundness for τ_Δ and $\tau_{\bar{\Delta}}$ [3]). *Let $\phi := A \sqsubseteq_{\mathcal{T}} B$, s.t. $\mathcal{T} \not\models \phi$, let $\mathcal{I}_{\mathcal{T}}$ be the canonical model of $\mathcal{T}$, and let $\otimes \in \{\Delta, \bar{\Delta}\}$. We have that $\tau_\otimes(\mathcal{I}_{\mathcal{T}})$ is a model of the $\otimes$-relevant subsumer sets.*

Proof sketch. For $(\tau_\Delta(\mathcal{I}_{\mathcal{T}}), b) \models \mathsf{S}_{\mathcal{T}}(B)$, one can simply adopt the proof from Lemma 4.1. We show $(\tau_\Delta(\mathcal{I}_{\mathcal{T}}), a) \models \mathsf{C}_{\mathcal{T}}(A, B)$ by contradiction. Assume there is an $H \in \mathsf{C}_{\mathcal{T}}(A, B)$, s.t. $(\tau_\Delta(\mathcal{I}_{\mathcal{T}}), a) \not\models H$. Hence, $(\mathcal{I}_{\mathcal{T}}, a) \models H$, $(\mathcal{I}_{\mathcal{T}}, b) \models H$ and $(\tau_\Delta(\mathcal{I}_{\mathcal{T}}), a) \not\models H$. This leads to a contradiction since the predicate *sim* is used to copy relevant parts in $\mathcal{I}_{\mathcal{T}}$ reachable from a in the definition scheme of τ_Δ. For $\tau_{\bar{\Delta}}$, recall that $\tau_{\bar{\Delta}}(\mathcal{I}_{\mathcal{T}}), b) \models \bar{\mathsf{S}}_{\mathcal{T}}(A, B)$ is required by Definition 3.3. This is achieved by using formula (18) in $\mathsf{D}_{\bar{\Delta}}$ with the additional conjunct $succ(z, x)$ to satisfy $(\tau_{\bar{\Delta}}(\mathcal{I}_{\mathcal{T}}), b) \models \{H \mid H \in \mathsf{S}_{\mathcal{T}}(B),\ H[N/\top][\exists r.\top/\top] \in \mathsf{S}_{\mathcal{T}}(A)\}$. Intuitively, this means that b in $\tau_{\bar{\Delta}}(\mathcal{I}_{\mathcal{T}})$ satisfies subsumers of B in $\mathcal{I}_{\mathcal{T}}$ resticted by the nesting depth of common subsumers with A plus 1 as the subsitution in $H[N/\top][\exists r.\top/\top]$ is applied only once. □

Theorem 4.4. *Computing τ_Δ and $\tau_{\bar{\Delta}}$ requires linear time.*

Proof. In the predicate $sim(a, b, x, y)$, is true for an element $\lambda(x)$ reachable from a if and only if there is an element $\lambda(y)$ reachable from b, s.t. the edge transitions from a to $\lambda(x)$ can also be made from b to $\lambda(y)$. The transitions can be seen as letters of words of accepted by a finite state automaton (FSA) that consists of a as start state, and the induced substructure of the elements reachable from a w.r.t. Σ_{R}, where every reachable element is a finial state. The same is done for the b induced substructure. The acceptance condition is that the word is readable in the a automaton as well as in the b automaton. Reading a word from an FSA and checking if the same word is accepted by another FSA requires linear time. □

4.2 Minimizing Transduction

Now, the resulting structure of the previous transductions is not necessarily minimal yet as required by Definition 3.3. In order to achieve minimality, we compose τ_α to $\tau_{\bar{\Delta}}$ with another transformation τ_{min} that yields a required minimal substructure of the image of τ_α to $\tau_{\bar{\Delta}}$ respectively. The transformation τ_{min} is a composition of intermediate transductions, which we introduce next.

For the images of τ_α and τ_β, the transduction τ_{min} is defined as composition of two further transduction τ_μ and τ_{reach}. The minimization process is simpler for τ_α and τ_β compared to τ_Δ and $\tau_{\bar{\Delta}}$, where τ_{min} requires an additional pre-processing transformation τ_{sim}. For the latter two cases, we define $\tau_{\mathsf{min}} := \tau_{\mathsf{reach}} \circ \tau_\mu \circ \tau_{\mathsf{sim}}$, and for the first two cases without τ_{sim}.

The intuition behind τ_{sim} is that, in some cases, the Δ and $\bar{\Delta}$ coarse relevant part still entails more than necessary because of loops. An additional check for roles is needed in order to reduce the unrequired entailments caused by loops reachable from a to a possible minimum. We define $\mathsf{D}_{\mathsf{sim}}$ by (9) as precondition, $\delta(\mathcal{W}, x) := \top(x)$ as domain formula, and as role name formulae

$$\eta_r(\mathcal{W}, x, y) := r(x, y) \wedge (reach(v, x) \vee (\exists x', y' : sim(u, v, x, x') \wedge r(x', y') \wedge ((reach(y, x) \wedge (\forall z : reach(y, z) \rightarrow \exists X : \pi[X, y, z] \wedge a \notin X)) \rightarrow (sim(x, x', y, y') \vee \exists y'' : r(x, y''))) \quad (19)$$

Concept names are assigned with an appropriate modification of (16) using in the sense of the third conjunct of (19). Intuitively, the roles that causes loops in the a reachable part are kept only if there is a similar role in the b reachable part or if the role is necessary to model common subsumers of A and B.

In the definition scheme of τ_{sim}, we use the simulation relation from the images of the transductions for the coarse relevant parts to themselves. It follows from [15] that, if there is a simulation relation, then there is a unique maximal simulation over them. The idea for τ_μ is to erase simulation equivalent elements but one w.r.t. a given simulation relation and to furthermore make non-maximal elements unreachable from a and b. To cut off the unreachable states in the resulting interpretation, τ_{reach} is applied afterwards, which is defined for images τ_β to $\tau_{\bar{\Delta}}$ as τ_β, and for the image of τ_α as τ_α.[5] As parameters for the definition scheme of τ_μ, we use a selection of representatives for the equivalence classes defined by

$$\mathbb{S} := \{[e] \mid [e] = \{d \in \mathcal{D}^{\mathcal{I}} \setminus \{a, b\} \mid (e \sim d) \wedge (d \sim e)\}\}.$$

The so defined relation is an equivalence relation. We enumerate the equivalence classes by $[e]_i$ (e.g. by using lexicographical order over the names of the domain elements). The parameters are defined as $\mathcal{W} := \{x_i \mid i \in [1 : |\mathbb{S}|]\} \cup \{u, v\}$, using the simulation relation from $\tau_\otimes(\mathcal{I}_\mathcal{T})$ to itself. We furthermore use the predicate $\mu(u, x)$ in D_μ for simulation maximal elements defined by

$$\mu(y, x) := reach(y, x) \wedge \forall z : reach(y, z) \wedge z \sim x \rightarrow x = z. \quad (20)$$

The precondition of definition scheme D_μ maps the representatives of the equivalence classes to the parameters while preferring common successor elements by

$$\chi(\mathcal{W}) := \bigwedge_{i=1}^{n} x_i \in [e]_i \wedge rep_A(u) \wedge rep_B(v) \wedge (\exists y : y \in [e]_i \wedge reach(u, y) \wedge reach(v, y) \rightarrow reach(u, x) \wedge reach(v, x)). \quad (21)$$

Domain elements are selected by a suitable choice of parameters.

$$\delta(\mathcal{W}, x) := \bigvee_{i=1}^{n} x = x_i, \quad \text{where } i \in [1 : |\mathcal{W}|] \quad (22)$$

The concept names $N \in \Sigma$ are assigned by $\theta_N(\mathcal{W}, x) := N(x)$. Lastly, the role name formulae of D_μ make non-maximal states unreachable by

$$\eta_r(\mathcal{W}, x, y) := r(x, y) \wedge \mu(u, x) \vee \mu(v, x). \quad (23)$$

Using the so defined transduction τ_{min}, we obtain the following theorem for the complete transformation process.

Theorem 4.5. *Let $\phi := A \sqsubseteq_\mathcal{T} B$ be a subsumption query, s.t. $\mathcal{T} \not\models \phi$, let $\mathcal{I}_\mathcal{T}$ be the canonical model of $\mathcal{T}$, and let $\otimes \in \{\alpha, \beta, \Delta, \bar{\Delta}\}$. Then, $\tau_{min}(\tau_\otimes(\mathcal{I}_\mathcal{T}))$ is an $\otimes$-relevant part of $\mathcal{I}_\mathcal{T}$ w.r.t. ϕ and $\mathcal{T}$.*

[5]We omit indexing τ_{min} as well as τ_{reach} as it is clear from the context.

Proof sketch. The single transformation steps of τ_{min} are model preserving w.r.t. to the $\otimes$-relevant subsumer sets and every substructure of the image of τ_{min} is not a model of the respective relevance sets. While τ_{sim} reduces the a reachable part by unnecessary loops by using the *sim* predicate and the simulation relation, τ_μ is model preserving as well. First, one suitable representative of the simulation equivalence classes is picked, and then, roles are added by to simulation maximal elements of the before selected domain elements only. This implies that the image of $\tau_{\mathsf{reach}} \circ \tau_\mu$ is simulation equivalent to the preimage. Furthermore, since maximal elements and one representative of the equivalence classes remain, by Theorems 2.2 and 2.1, reducing the image of τ_{min} would violate the satisfaction of the relevant subsumer sets. □

Computing the largest simulation between two finite interpretations needs polynomial time [12]. Hence, we can make the following statement regarding the computational effort for τ_{min}.

Theorem 4.6. *Computing τ_{min} requires polynomial time.*

4.3 Post-processing Transductions

As final remarks in to this section, we suggest two more simple transformations that can be composed with the previously introduced transformations. First, we have assumed throughout the paper that the TBox is given in normal form. In order to clean the countermodel from the freshly introduced concept names of the normalization process, we devise a transduction defined by

$$\langle \delta(x) := \top(x), \theta_N(x) := N(x), \eta_r(x, y) := r(x, y) \rangle$$

for $\mathrm{sig}(\mathcal{T})$, where $\mathcal{T}$ is the not yet normalized TBox. Second, in order to avoid shared elements that are reachable from the A and B representatives, one can use transduction that copies elements. We have not formally introduced copying transductions, but the idea is that each element reachable from the A representative gets an index value assigned, the same is done for all elements from the B representative with a second index value, and the domain of the transductions image is the disjoint union of both sets of indexed elements.

5 IMPLEMENTATION AND EVALUATION

In this section we describe our implementation for extracting relevant parts of countermodels and the experiment we ran to evaluate it. We first implement the procedure of computing canonical models of $\mathcal{EL}$ ontologies [4] using the reasoner Elk [20]. Since the computation of a canonical model is not a part of our approach for generating relevant parts of countermodels (we assume it is a part of the input), we do not include statistical data about the efficiency of computing these models. However, we show how we adapt our approach to be more practical and circumvent the computation of the entire canonical model. We also implement a procedure for generating and minimizing the four different types of relevant parts as defined in Definition 3.3, which follows the logic of the transductions introduced in Sections 4. The tool and the experiment are implemented using Java. The evaluation was conducted on a Linux Debian machine (Intel Xeon CPU E5-2640 0, 2.50 GHz, 20 GB Java heap space). The Experiment resources are available here [2].

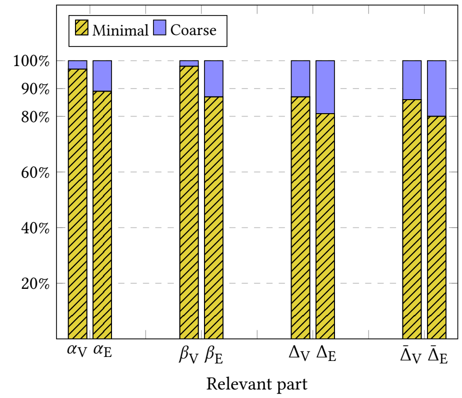

Figure 6: Average reduction percentage in *elements count* (left column) and in *edges count* (right column) for each of the relevant parts for all subsumption queries

Relevant Axioms. Extracting a relevant part from a canonical model of the full TBox is guided by the input subsumption query, and more precisely, by the signature of this query. Therefore, parts of the canonical model that are not related to this signature are discarded. This means that the TBox axioms that are modeled by these parts can also be discarded. Then instead of computing the entire canonical model of the TBox, we can compute the canonical model of a subset of the TBox that contains the relevant axioms. These subsets are called *modules* [17]. There are different types of modules, and in our setting, we use what is called $\bot$*-modules*. Intuitively, extracting a $\bot$-module for the signature of a subsumption query, results in a module containing all subsumers of concepts of this signature. Hence a module allows us to focus on a relatively smaller set of axioms which leads to a relatively smaller canonical model. In order to compute $\bot$-modules, we use the module extraction functionality provided by the OWL API [6].

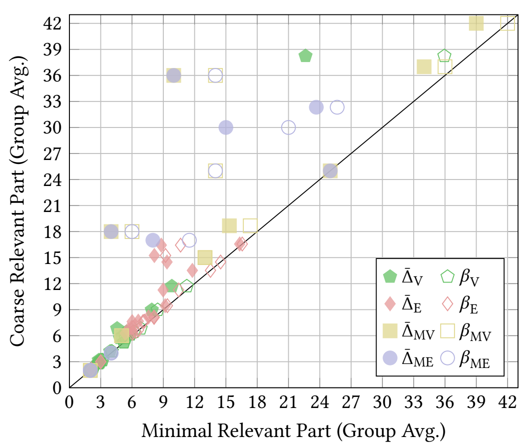

Figure 7: Groups average reduction for $\bar{\Delta}$ and β relevant parts

Extension and refinement. After computing the canonical model of a module, we extract the four different coarse relevant parts. The implementation of these extraction functionalities is relatively straightforward, and based on ideas introduced in Section 4. Since a coarse relevant part is extracted from a canonical model, each path in its graph can be mapped to an $\mathcal{EL}$ concept in the signature of the TBox. If there are two paths p_1 and p_2 starting at some element e, where some concepts C_1 and C_2 are mapped to p_1 and p_2 respectively s.t. $C_1 \sqsubseteq C_2$. Then p_2 is redundant w.r.t. C_2 because p_1 models both C_1 and C_2. So in order to obtain a minimal relevant part, we first *extend* each element e in a coarse part with a set of concepts representing every path starting at e. We then *refine* each element by removing redundant paths. A path p starting at an element e is redundant iff for every concept C that is mapped to p, there exists some other path p' starting at e s.t. C' is mapped to p' and $\models C' \sqsubseteq C$.

Experiment description. We ran our experiment using $\mathcal{EL}$ ontologies extracted from ORE 2015 Reasoner Competition Corpus [25]. We considered ontologies with a size $\leq$ 10 MB, with a total of 171 ontologies out of 235. Creating a relatively large number of plausible subsumption queries that lead to interesting relevant parts while avoiding selection bias is not an easy task. In total, we generate 11,600 subsumption queries. We limit the total of queries per ontology to 1% of the amount of concept names appearing in it, with a minimum of 50 queries. The idea here is that the more concepts an ontology has, the more queries one can raise. The queries are generated randomly, but we also tried to create queries that are "reasonable" in the context of their TBox. We start by randomly selecting a concept C from a TBox $\mathcal{T}$, and all concepts C' appearing in $\mathcal{T}$ s.t. $\mathcal{T} \models C' \sqsubseteq C$. A subsumption query is then an axiom $\eta := C'_1 \sqsubseteq C'_2$ where $\mathcal{T} \not\models \eta$. For each subsumption query, the four types of coarse relevant parts and their minimization are computed. Each of these computations is set to time out after 10 minutes. For 11,544 subsumption queries, all types of relevant parts are computed successfully. Whereas 56 queries timed out at the minimization stage. In total, four ontologies timed out, hence 167 ontologies were successfully processed.

Experiment results. We aggregate subsumption queries, and by extension the different relevant parts in four group categories. Two of which are based on the canonical model of the TBox from which the subsumption query is generated. These group categories are labeled with V and E. The former is based on the total number of elements in the canonical model of the TBox, whereas the latter is based on the total number of edges. The other two group categories, labeled MV and ME, are based on the total number of elements and edges respectively, but of canonical models of modules instead of full TBoxes. There are 18 groups in each group category, and the size of each group is depicted in Figure 12. In Figure 6 we can see that on average, the minimization of coarse parts leads to 2% – 20% reduction in these relevant parts. Because of the nature of canonical models, and how ontologies are usually designed, we can see that redundancy occurs much more in edges in comparison to domain elements. However, when we consider Δ and $\bar{\Delta}$ parts, we get more redundancy in domain elements in comparison to α and β parts. For β_V and β_E, the max reduction percentage per ontology is 39% and 89% respectively. Analogously, the max reduction w.r.t. $\bar{\Delta}_V$ and $\bar{\Delta}_E$ is 72% and 88%. In Figure 7 we can see the average reductions of β and $\bar{\Delta}$ w.r.t. the four grouping categories. Figures 8-11 show the average computation time of the coarse (lower opacity

[6] http://owlcs.github.io/owlapi/

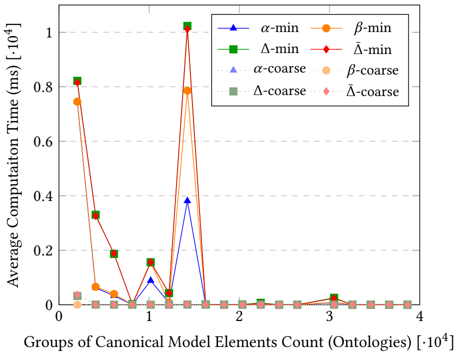

Figure 8: V**-groups average comp. time of relevant parts**

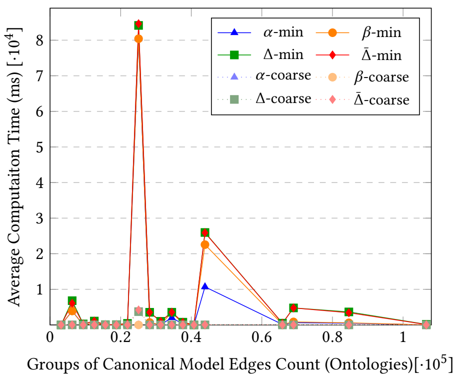

Figure 9: E**-groups average comp. time of relevant parts**

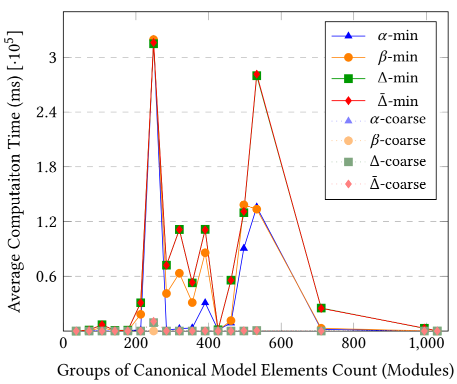

Figure 10: MV**-groups average comp. time of relevant parts**

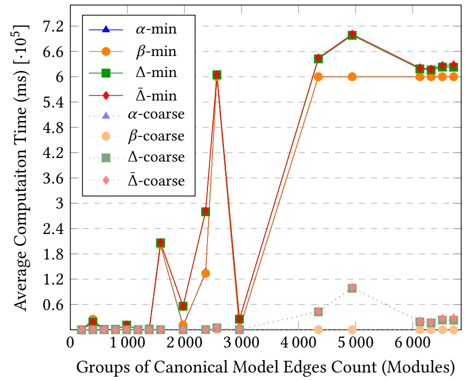

Figure 11: ME**-groups average comp. time of relevant parts**

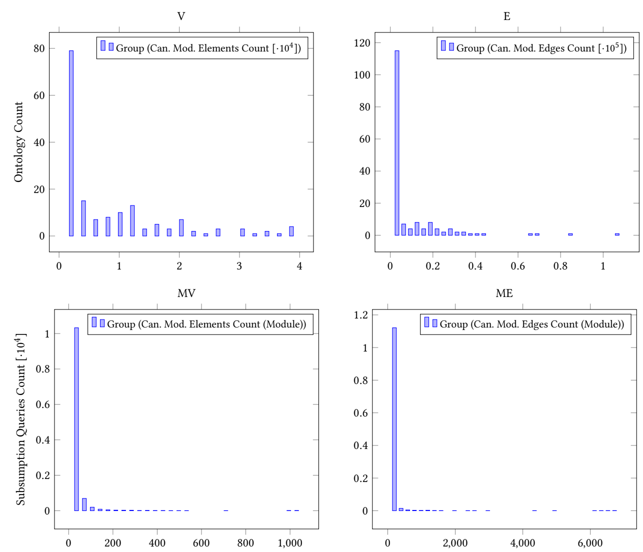

Figure 12: Groups Categories

markers) and minimized relevant parts. In Figures 8 and 9 we get an average time that ranges over seconds but with no clear trend for the computation behavior. On the other hand, in Figures 10 and 11 we get an average time that ranges over minutes but with more informative chart. It is clear that larger values on x-axes do not imply longer computation times. However, the more edges in a coarse part, the higher the chance that the refinement becomes harder. The main disadvantage with our approach is not the large number of edges in a canonical model, but rather the tighter concept dependencies which translate in more connectivity in the graph.

6 CONCLUSIONS AND FUTURE WORK

We have introduced an approach for explaining non-entailment of $\mathcal{EL}$ ontologies based on various notions of relevance and discussed the computational properties of the problem of extracting relevant parts from canonical models. The extraction is described by means of model transformations using graph transductions. We furthermore present an implementation of our approach and discuss the results of our evaluation.

As for future work, we would like first to conduct a user study to confirm whether the relevant parts are enhancing users understanding of non-entailments. Second, we will look into different reasoning tasks such as explaining negative query answering results [10]. Third, we will consider more expressive DLs. However, many of these DLs do not admit the canonical model property, which requires a substantial adaptation of our approach.

ACKNOWLEDGMENTS

Supported by the DFG in grant 389792660 as part of TRR 248–CPEC (https://perspicuous-computing.science), and QuantLA, Research Training Group 1763 (https://lat.inf.tu-dresden.de/quantla).

REFERENCES

[1] Christian Alrabbaa, Franz Baader, Stefan Borgwardt, Patrick Koopmann, and Alisa Kovtunova. 2020. Finding Small Proofs for Description Logic Entailments: Theory and Practice. In *Proceedings of the 23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR) (EPiC Series in Computing, Vol. 73)*, Elvira Albert and Laura Kovács (Eds.). EasyChair, 32–67.

[2] Christian Alrabbaa and Willi Hieke. 2022. *Explaining Non-Entailment by Model Transformation for the Description Logic $\mathcal{EL}$ - IJCKG22 - Resources.* https://doi.org/10.5281/zenodo.7064514

[3] Christian Alrabbaa, Willi Hieke, and Anni-Yasmin Turhan. 2021. Counter Model Transformation for Explaining Non-Subsumption in $\mathcal{EL}$. In *Proceedings of the 7th Workshop on Formal and Cognitive Reasoning co-located with 44th German Conference on Artificial Intelligence (KI) (CEUR Workshop Proceedings, Vol. 2961)*, Christoph Beierle, Marco Ragni, Frieder Stolzenburg, and Matthias Thimm (Eds.). CEUR-WS.org, 9–22.

[4] Franz Baader, Sebastian Brandt, and Carsten Lutz. 2005. Pushing the $\mathcal{EL}$ Envelope. In *IJCAI-05, Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, Leslie Pack Kaelbling and Alessandro Saffiotti (Eds.). Professional Book Center, 364–369.

[5] Franz Baader, Ian Horrocks, Carsten Lutz, and Uli Sattler. 2017. *An Introduction to Description Logic.* Cambridge University Press.

[6] Franz Baader, Carsten Lutz, and Boontawee Suntisrivaraporn. 2006. CEL - A Polynomial-Time Reasoner for Life Science Ontologies. In *Proceedings of the Third International Joint Conference on Automated Reasoning (IJCAR) (LNCS, Vol. 4130)*. Springer, 287–291.

[7] Franz Baader, Rafael Peñaloza, and Boontawee Suntisrivaraporn. 2007. Pinpointing in the Description Logic $\mathcal{EL}^+$. In *Proceedings of the 30th Annual German Conference on AI (KI) (Lecture Notes in Computer Science, Vol. 4667)*, Joachim Hertzberg, Michael Beetz, and Roman Englert (Eds.). Springer, 52–67.

[8] Alexander Borgida, Enrico Franconi, and Ian Horrocks. 2000. Explaining $\mathcal{ALC}$ Subsumption. In *ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence*, Werner Horn (Ed.). IOS Press, 209–213.

[9] Sebastian Brandt. 2004. Polynomial Time Reasoning in a Description Logic with Existential Restrictions, GCI Axioms, and - What Else?. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence (ECAI)*, Ramón López de Mántaras and Lorenza Saitta (Eds.). IOS Press, 298–302.

[10] Diego Calvanese, Magdalena Ortiz, Mantas Simkus, and Giorgio Stefanoni. 2013. Reasoning about Explanations for Negative Query Answers in DL-Lite. *Journal of Artificial Intelligence Research* 48 (2013), 635–669.

[11] İsmail İlkan Ceylan, Thomas Lukasiewicz, Enrico Malizia, Cristian Molinaro, and Andrius Vaicenavicius. 2020. Explanations for Negative Query Answers under Existential Rules. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, Diego Calvanese, Esra Erdem, and Michael Thielscher (Eds.). IJCAI Organization, 223–232.

[12] Edmund M. Clarke and Bernd-Holger Schlingloff. 2001. *Model Checking.* Elsevier, 1635—1790.

[13] Bruno Courcelle and Joost Engelfriet. 2012. *Graph Structure and Monadic Second-Order Logic.* Encyclopedia of mathematics and its applications, Vol. 138. CUP.

[14] Kathrin Dentler, Ronald Cornet, Annette ten Teije, and Nicolette de Keizer. 2011. Comparison of reasoners for large ontologies in the OWL 2 $\mathcal{EL}$ profile. *Semantic Web* 2, 2 (2011), 71–87.

[15] Orna Grumberg and David E Long. 1994. Model checking and modular verification. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 16, 3 (1994), 843–871.

[16] Willi Hieke and Anni-Yasmin Turhan. 2020. Towards Model Transformation in Description Logics — Investigating the Case of Transductions. In *Proceedings of the 6th Workshop on Formal and Cognitive Reasoning co-located with 43rd German Conference on Artificial Intelligence (KI) (CEUR Workshop Proceedings, Vol. 2680)*, Christoph Beierle, Marco Ragni, Frieder Stolzenburg, and Matthias Thimm (Eds.). CEUR-WS.org, 69–82.

[17] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ulrike Sattler, Thomas Schneider, and Rafael Berlanga Llavori. 2008. Safe and Economic Re-Use of Ontologies: A Logic-Based Methodology and Tool Support. In *Proceedings of the 21st International Workshop on Description Logics (DL) (CEUR Workshop Proceedings, Vol. 353)*, Franz Baader, Carsten Lutz, and Boris Motik (Eds.). CEUR-WS.org.

[18] Neil D. Jones, Y. Edmund Lien, and William T. Laaser. 1976. New Problems Complete for Nondeterministic Loc Space. *Mathematical Systems Theory* 10 (1976), 1–17.

[19] Maciej Juryńczyk, Matthew Craner, and Jacqueline Palace. 2015. Overlapping CNS inflammatory diseases: differentiating features of NMO and MS. *Journal of Neurology, Neurosurgery & Psychiatry* 86, 1 (2015), 20–25.

[20] Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancik. 2014. The Incredible ELK — From Polynomial Procedures to Efficient Reasoning with $\mathcal{EL}$ Ontologies. *Journal of Automated Reasoning* 53, 1 (2014), 1–61.

[21] Patrick Koopmann. 2021. Two ways of explaining negative entailments in description logics using abduction. In *Informal Proceedings of the 2nd Workshop on Explainable Logic-Based Knowledge Representation (XLoKR) co-located with the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR).*

[22] Michael J Lawley and Cyril Bousquet. 2010. Fast classification in Protégé: Snorocket as an OWL 2 $\mathcal{EL}$ reasoner. In *Proceedings of the 6th Australasian Ontology Workshop (IAOA). Conferences in Research and Practice in Information Technology*, Vol. 122. 45–49.

[23] Vanda A Lennon, Dean M Wingerchuk, Thomas J Kryzer, Sean J Pittock, Claudia F Lucchinetti, Kazuo Fujihara, Ichiro Nakashima, and Brian G Weinshenker. 2004. A serum autoantibody marker of neuromyelitis optica: distinction from multiple sclerosis. *The Lancet* 364, 9451 (2004), 2106–2112.

[24] Carsten Lutz and Frank Wolter. 2010. Deciding Inseparability and Conservative Extensions in the Description Logic $\mathcal{EL}$. *Journal of Symbolic Computing* 45, 2 (2010), 194–228.

[25] Nicolas Matentzoglu and Bijan Parsia. 2015. *ORE 2015 Reasoner Competition Corpus.* https://doi.org/10.5281/zenodo.18578

[26] Stefan Schlobach and Ronald Cornet. 2003. Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, Georg Gottlob and Toby Walsh (Eds.). Morgan Kaufmann, 355–362.