Computing Minimal Projection Modules for $\mathcal{ELH}^r$-Terminologies

Jieying Chen[1](✉), Michel Ludwig[2], Yue Ma[1], and Dirk Walther[3]

[1] Laboratoire de Recherche en Informatique, Université Paris-Saclay, Paris, France
{jieying.chen,yue.ma}@lri.fr
[2] Beaufort, Luxembourg
michel.ludwig@gmail.com
[3] Fraunhofer IVI, Dresden, Germany
dirk.walther@ivi.fraunhofer.de

Abstract. For the development of large-scale representations of knowledge, the application of methodologies and design principles becomes relevant. The knowledge may be organized in ontologies in a modular and hierarchical fashion. An upper-level (reference) ontology typically provides specifications of requirements, functions, design or standards that are to be complied with by domain ontologies for a specific task on a lower level (task ontology) in the hierarchy. Verifying whether and how specifications have been implemented by a task ontology becomes a challenge when relevant axioms of the domain ontology need to be inspected. We consider specifications to be defined using entailments of certain queries over a given vocabulary. For selecting the relevant axioms from task ontologies, we propose a novel module notion called *projection module* that entails the queries that follow from a reference ontology. We develop algorithms for computing minimal projection modules of Description Logic terminologies for subsumption, instance and conjunctive queries.

1 Introduction

A common practice in the area of the Semantic Web is to reuse and extend existing ontologies for a specific task. Therefore, an approach to comparing multiple ontologies is often desired. In this paper, we propose the notion of *projection module* which characterizes the relative knowledge of an ontology, say, an ontology developed for a specific task (called *task ontology*), by taking another ontology as a reference (called *reference ontology*), e.g. an upper-level ontology. This can thus lead to (1) a method for comparing the entailment capacities of any two ontologies about a given vocabulary of interest, and (2) a fine-grained ontology comparison measurement between two ontologies.

As illustrated in Fig. 1, for a user interest expressed as a set Σ of concept and role names, the reference ontology $\mathcal{T}_1$ (resp. task ontology $\mathcal{T}_2$) contains

This work is partially funded by the ANR project GoAsQ (ANR-15-CE23-0022).

F. Calimeri et al. (Eds.): JELIA 2019, LNAI 11468, pp. 355–370, 2019.
https://doi.org/10.1007/978-3-030-19570-0_23

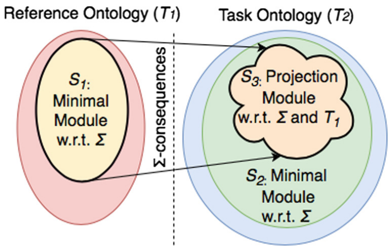
Reference Ontology ($\mathcal{T}_1$)
Task Ontology ($\mathcal{T}_2$)
S_1: Minimal Module w.r.t. Σ
Σ-consequences
S_3: Projection Module w.r.t. Σ and $\mathcal{T}_1$
S_2: Minimal Module w.r.t. Σ

Fig. 1. Projection module

$\mathcal{T}_1 = \{\alpha_1\}$, $\mathcal{T}_2 = \{\beta_1, \beta_2, \beta_3, \beta_4\}$

$$\begin{array}{lrl}
\alpha_1 : & \mathsf{Professor} & \sqsubseteq \mathsf{Faculty} \sqcap \exists \mathsf{has.PhD} \sqcap \exists \mathsf{teach.Course} \\
\beta_1 : & \mathsf{Professor} & \sqsubseteq \mathsf{Faculty} \sqcap \exists \mathsf{has.PhD} \sqcap \\
 & & \quad \exists \mathsf{teach.Master\_Course} \sqcap \\
 & & \quad \exists \mathsf{teach.Bachelor\_Course} \\
\beta_2 : & \mathsf{Faculty} & \sqsubseteq \exists \mathsf{has.PhD} \\
\beta_3 : & \mathsf{Bachelor\_Course} & \sqsubseteq \mathsf{Course} \\
\beta_4 : & \mathsf{Master\_Course} & \sqsubseteq \mathsf{Course}
\end{array}$$

Fig. 2. Example: minimal projection modules

S_1 (resp. S_2) as a sub-ontology, e.g., a minimal module [9,10], that provides a minimal number of axioms entailing all Σ-consequences. A projection module, on the other hand, additionally takes a reference ontology into account as to preserve the relevant Σ-knowledge only, which can yield an even smaller module S_3 of $\mathcal{T}_2$. Figure 2 shows a concrete example with $\mathcal{T}_1$ being the upper-level (reference) ontology modelling aspects of a university domain, and a task ontology $\mathcal{T}_2$ as an extended and modified version of $\mathcal{T}_1$. Consider the signature $\Sigma = \{\mathsf{Professor}, \mathsf{Faculty}, \mathsf{has}, \mathsf{PhD}, \mathsf{teach}, \mathsf{Course}\}$ (consisting of the symbols marked in blue in Fig. 2). Then $\mathcal{T}_1$ has merely one module $S_1 = \{\alpha_1\}$ that preserves $\mathcal{T}_1$'s knowledge about Σ. When extracting modules of $\mathcal{T}_2$ for Σ, we obtain $\mathcal{T}_2$ itself using existent module notions such as modules based on locality [14] or the module extracted by MEX [20]. Here we have two candidates for a minimal module S_2 of $\mathcal{T}_2$ that each preserve all inclusions over Σ: $\{\beta_1, \beta_2, \beta_3\}$ and $\{\beta_1, \beta_2, \beta_4\}$ [9]. The projection modules of $\mathcal{T}_2$, however, preserving the Σ-inclusions entailed by $\mathcal{T}_1$ are even smaller with $S_3 = \{\beta_1, \beta_3\}$ or $S_3 = \{\beta_1, \beta_4\}$. Every projection module S_3 is a strict subset of a minimal module of $\mathcal{T}_2$, which is in line with the fact that the task ontology $\mathcal{T}_2$ has extended the reference ontology $\mathcal{T}_1$ with new Σ-consequences, e.g., $\mathsf{Faculty} \sqsubseteq \exists \mathsf{has.PhD}$.

Various approaches to comparing ontologies have been suggested, including ontology matching [13] and logical difference [22–24,26]. Ontology matching is the process of determining correspondences, e.g., the subsumption relations between two concept or role names from different ontologies, for which a good concept similarity [1,25] is often helpful. In contrast, logical difference focuses on the comparison of entailed logical consequences from each ontology and returns difference witnesses if differences are present. When an ontology has no logical difference compared to another one, our approach further extracts sub-ontologies of the first that contain the knowledge as represented by the second ontology.

Ontology modularity [9,18,22,24,27,28] is about the extraction of sub-ontologies that preserve all logical consequences over a signature. The proposed projection module is different from modules of a single ontology, as illustrated by the example in Fig. 2. To compute projection modules, in this paper, we generalize the notion of justification to the notion of *subsumption justification* as a minimal set of axioms that maintains the entailment of a consequence. Our algorithm employs the classical notion of justification to compute subsumption justifications. Currently, the approaches for computing all justifications of an ontology for a consequence can be classified into two categories: "glass-box" [2,5,16,17] and "black-box" [11,16,29].

We proceed as follows. After reviewing some preliminaries in Sect. 2, the notion of a minimal project module for subsumption, instance and conjunctive queries is introduced in Sect. 3. In Sect. 4, we introduce the algorithm for computing minimal projection modules. In Sect. 5, two applications of minimal projection modules are presented. Finally, we close the paper with a conclusion in Sect. 6.

2 Preliminaries

We start by reviewing the description logic $\mathcal{EL}$ and several of its extensions.

Let $\mathsf{N_C}$, $\mathsf{N_R}$ and $\mathsf{N_I}$ be mutually disjoint and countably infinite sets of concept names, role names and instance names. The signature $\mathrm{sig}(\xi)$ is the set of concept and role names occurring in ξ, where ξ ranges over any syntactic object. The sets of $\mathcal{EL}$ *-concepts* C, $\mathcal{EL}^{\mathsf{ran}}$ *-concepts* D, $\mathcal{EL}^{\sqcap}$ *-concepts* E, and $\mathcal{EL}^{\sqcap,u}$ *-concepts* F, and the sets of $\mathcal{ELH}^r$*-inclusions* α, $\mathcal{EL}^{\mathsf{ran}}$*-inclusions* β and $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$*-inclusions* γ are built according to the grammar rules:

$$\begin{array}{rcl}
C & ::= & A \mid C \sqcap C \mid \exists r.C \mid \mathsf{dom}(r) \\
D & ::= & A \mid D \sqcap D \mid \exists r.D \mid \mathsf{dom}(r) \mid \mathsf{ran}(r) \\
E & ::= & A \mid E \sqcap E \mid \exists R.E \\
F & ::= & A \mid F \sqcap F \mid \exists R.F \mid \exists u.F \\
\alpha & ::= & C \sqsubseteq C \mid \mathsf{ran}(r) \sqsubseteq C \mid \mathsf{ran}(r) \sqcap C \sqsubseteq C \mid C \equiv C \mid r \sqsubseteq s \\
\beta & ::= & D \sqsubseteq C \mid r \sqsubseteq s \\
\gamma & ::= & D \sqsubseteq F \mid r \sqsubseteq s
\end{array}$$

where $A \in \mathsf{N_C}$, $r, s \in \mathsf{N_R}$, u is a fresh logical symbol (the *universal role*) and $R = r_1 \sqcap \ldots \sqcap r_n$ with $r_1, ..., r_n \in \mathsf{N_R}$, for $n \geq 1$. We refer to inclusions also as *axioms*. A Γ*-TBox* is a finite set of Γ-inclusions, where Γ ranges over the sets of $\mathcal{ELH}^r$- and $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$-inclusions. We use $\mathrm{lhs}(\alpha)$ (resp. $\mathrm{rhs}(\alpha)$) to represent the left-hand side (resp. right-hand side) of an inclusion α.

The semantics is defined as usual in terms of interpretations interpreting concept/role names as unary/binary relations and are then inductively extended to complex concepts. The notions of satisfaction of a concept, axiom and TBox as well as the notions of a model and the logical consequence relation are defined as usual [4].

An $\mathcal{ELH}^r$ *-terminology* $\mathcal{T}$ is an $\mathcal{ELH}^r$-TBox consisting of axioms α of the form $A \sqsubseteq C$, $A \equiv C$, $r \sqsubseteq s$, $\mathsf{ran}(r) \sqsubseteq C$ or $\mathsf{dom}(r) \sqsubseteq C$, where A is a concept name, C an $\mathcal{EL}$-concept and no concept name occurs more than once on the left-hand side of an axiom. To simplify the presentation we assume that terminologies do not contain axioms of the form $A \equiv B$ or $A \equiv \top$ (after having removed multiple $\top$-conjuncts) for concept names A and B. For a terminology $\mathcal{T}$, let $\prec_{\mathcal{T}}$ be a binary relation over $\mathsf{N_C}$ satisfying that $A \prec_{\mathcal{T}} B$ iff there is an axiom of the form

$A \sqsubseteq C$ or $A \equiv C$ in $\mathcal{T}$ such that $B \in \mathrm{sig}(C)$. A terminology $\mathcal{T}$ is *acyclic* if the transitive closure $\prec_{\mathcal{T}}^{+}$ of $\prec_{\mathcal{T}}$ is irreflexive; otherwise $\mathcal{T}$ is *cyclic*. We say that a concept name A is *conjunctive in* $\mathcal{T}$ iff there exist concept names $B_1, \ldots, B_n$, $n > 0$, such that $A \equiv B_1 \sqcap \ldots \sqcap B_n \in \mathcal{T}$; otherwise A is said to be *non-conjunctive in* $\mathcal{T}$. An $\mathcal{ELH}^r$-terminology $\mathcal{T}$ is *normalised* iff it only contains axioms of the forms

- $r \sqsubseteq s$, $\varphi \sqsubseteq B_1 \sqcap \ldots \sqcap B_n$, $A \sqsubseteq \exists r.B$, $A \sqsubseteq \mathsf{dom}(r)$, and
- $A \equiv B_1 \sqcap \ldots \sqcap B_m$, $A \equiv \exists r.B$,

where $\varphi \in \{A, \mathsf{dom}(s), \mathsf{ran}(s)\}$, $n \geq 1$, $m \geq 2$, $A, B, B_i \in \mathsf{N_C}$, $r, s \in \mathsf{N_R}$, and each conjunct B_i is non-conjunctive in $\mathcal{T}$. Every $\mathcal{ELH}^r$-terminology $\mathcal{T}$ can be normalised in polynomial time such that the resulting terminology is a conservative extension of $\mathcal{T}$ [19]. A subset $M \subseteq \mathcal{T}$ is called a *justification for an $\mathcal{ELH}$ -concept inclusion α from $\mathcal{T}$* iff $M \models \alpha$ and $M' \not\models \alpha$ for every $M' \subsetneq M$.

We denote the set of all justifications for an $\mathcal{ELH}$-concept inclusion α from an $\mathcal{ELH}$-terminology $\mathcal{T}$ with $\mathrm{Just}_{\mathcal{T}}(\alpha)$. The latter may contain exponentially many justifications in the number of axioms in $\mathcal{T}$. An ABox contains *assertions* of the form $\top(a)$, $A(a)$ and $r(a, b)$, where $a, b \in \mathsf{N_I}$ and $r \in \mathsf{N_R}$. An *ABox* consists of finitely many ABox assertions.

Let $\mathsf{N_I}$ and $\mathsf{N_V}$ be disjoint sets of individual and variable names. A conjunctive query is a first-order formula built according to the following format: $\exists y_1 \ldots \exists y_n . \bigwedge_{i \in I_1} A_i(s_i) \wedge \bigwedge_{j \in I_2} r_j(t_j, t'_j)$, where $y_1, \ldots, y_n \in \mathsf{N_V}$ for $n \geq 1$ are variable names, I_1, I_2 are finite sets of indices, and for $i \in I_1$ and $j \in I_2$, A_i ranges over concept names in $\mathsf{N_C}$, r_j ranges over role names in $\mathsf{N_R}$, and s_i, t_j, t'_j range over individual and variable names in $\mathsf{N_I} \cup \mathsf{N_V}$.

A signature Σ is a finite set of symbols from $\mathsf{N_C}$ and $\mathsf{N_R}$. The symbol Σ is used as a subscript to sets of concepts or inclusions to denote that the elements only use symbols from Σ, e.g., $\mathcal{ELH}^r_\Sigma$ and $\mathcal{EL}^{\mathsf{ran}, \sqcap, u}_\Sigma$. For a signature Σ, let $\Sigma^{\mathsf{dom}} = \{\, \mathsf{dom}(r) \mid r \in \mathsf{N_R} \cap \Sigma \,\}$ and $\Sigma^{\mathsf{ran}} = \{\, \mathsf{ran}(r) \mid r \in \mathsf{N_R} \cap \Sigma \,\}$ be the sets consisting of concepts of the form $\mathsf{dom}(r)$ and $\mathsf{ran}(r)$ for every role name r in Σ, respectively. We recall the notion of logical difference for concept subsumption queries, instance queries and conjunctive queries from [19,23]. For a more detailed introduction to description logics, we refer to [3,4]. For latest results on logical inseparability see [7,8,15], and for a survey on query inseparability, see [6].

Definition 1 (Logical Difference). *The $\mathcal{L}$-subsumption query difference, for some logic $\mathcal{L}$, the instance and conjunctive query difference between $\mathcal{T}_1$ and $\mathcal{T}_2$ w.r.t. Σ are the sets $\mathsf{cDiff}^{\mathcal{L}}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, $\mathsf{iDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, and $\mathsf{qDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, respectively, where*

- *$\varphi \in \mathsf{cDiff}^{\mathcal{L}}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ iff φ is an $\mathcal{L}$-inclusion, $\mathcal{T}_1 \models \varphi$ and $\mathcal{T}_2 \not\models \varphi$;*
- *$(\mathcal{A}, \lambda) \in \mathsf{iDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ iff $\mathcal{A}$ is a Σ-ABox and λ a Σ-instance assertion such that $(\mathcal{T}_1, \mathcal{A}) \models \lambda$ and $(\mathcal{T}_2, \mathcal{A}) \not\models \lambda$;*
- *$(\mathcal{A}, q(\boldsymbol{a})) \in \mathsf{qDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ iff $\mathcal{A}$ is a Σ-ABox and $q(\boldsymbol{a})$ a Σ-conjunctive query such that $(\mathcal{T}_1, \mathcal{A}) \models q(\boldsymbol{a})$ and $(\mathcal{T}_2, \mathcal{A}) \not\models q(\boldsymbol{a})$.*

According to [19], $\mathcal{L}$-subsumption queries for $\mathcal{L} = \mathcal{EL}^{\mathsf{ran}}$ and $\mathcal{L} = \mathcal{EL}^{\mathsf{ran},\sqcap,u}$ are sufficient to detect the absence of any instance query and conjunctive query differences, respectively. Therefore, we only consider how to detect $\mathcal{L}$-subsumption queries for $\mathcal{L} = \{\mathcal{ELH}^r, \mathcal{EL}^{\mathsf{ran}}, \mathcal{EL}^{\mathsf{ran},\sqcap,u}\}$. Let $\alpha^{\mathcal{L}}_{\Sigma}$ be an $\mathcal{L}$-inclusion that only uses symbols in Σ. We organise the Σ-symbols (and the domain and range concepts over role names from Σ) that occur as "witnesses" of a $\mathcal{L}$-subsumption query difference between $\mathcal{T}_1$ and $\mathcal{T}_2$ as follows:

$$\mathsf{Wtn}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) := (\mathsf{roleWtn}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2), \mathsf{lhsWtn}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2), \mathsf{rhsWtn}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)),$$

where $\mathsf{roleWtn}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) = \{\, r \in \Sigma \cap \mathsf{N_R} \mid r \sqsubseteq s \text{ or } s \sqsubseteq r \in \mathsf{cDiff}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) \,\}$, $\mathsf{lhsWtn}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) = \{\varphi \in (\Sigma \cap \mathsf{N_C}) \cup \Sigma^{\mathsf{dom}} \cup \Sigma^{\mathsf{ran}} \mid \varphi \sqsubseteq \mathrm{rhs}(\alpha^{\mathcal{L}}_{\Sigma})$ and α is a $\mathcal{L}_{\Sigma}$ − inclusion$\}$ and $\mathsf{rhsWtn}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) = \{\, A \in \Sigma \cap \mathsf{N_C} \mid \mathrm{lhs}(\alpha^{\mathcal{L}}_{\Sigma}) \sqsubseteq A \in \mathsf{cDiff}^{Q}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) \,\}$. The set $\mathsf{Wtn}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$ can be seen as a finite representation of the set $\mathsf{cDiff}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$, which is typically infinite when it is not empty. It follows from the "primitive witnesses" theorems in [19] that $\mathsf{cDiff}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$ iff $\mathsf{Wtn}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) = (\emptyset, \emptyset, \emptyset)$. Thus, deciding the existence of logical differences is equivalent to decide non-emptiness of the three witness sets.

3 Projection Modules

To understand the relations among different ontologies, we introduce the notion of projection module, as a way to explain how the knowledge that is encoded in a reference ontology is implemented in a task ontology. We are interested in computing *all* projection modules, since it provides a complete list of all implementations of an ontology regarding a reference, each of which may be necessary to be checked. To enable a manual validation by domain experts, we need to present only necessary information, so we focus on computing *minimal* projection modules.

A terminology $\mathcal{T}_1$ together with a signature Σ and a type of query Q determine a set Φ of queries from Q formulated using only symbols from Σ that follow from $\mathcal{T}_1$. A projection module of another terminology $\mathcal{T}_2$ is a subset of $\mathcal{T}_2$ that entails the queries in Φ. For convenience, we bundle the parameters together in a tuple $\rho = \langle \mathcal{T}_1, \Sigma, \mathcal{T}_2 \rangle$, which we call a *projection setting.*

Definition 2 (Projection Module). *Let $\rho = \langle \mathcal{T}_1, \Sigma, \mathcal{T}_2 \rangle$ be a projection setting, $\mathcal{A}$ be a Σ-ABox. A subset $\mathcal{M} \subseteq \mathcal{T}_2$ is a* subsumption (resp. instance, conjunctive) query projection module *under projection setting ρ, denoted as $\mathcal{M}^c_\rho$ (resp. $\mathcal{M}^i_\rho$, $\mathcal{M}^q_\rho$) iff:*

- *$\mathcal{M}^c_\rho$: for each $\mathcal{ELH}^r_\Sigma$-inclusion α, if $\mathcal{T}_1 \models \alpha$, then $\mathcal{M} \models \alpha$;*
- *$\mathcal{M}^i_\rho$: for each Σ-instance assertion λ, if $(\mathcal{T}_1, \mathcal{A}) \models \lambda$, then $(\mathcal{M}, \mathcal{A}) \models \lambda$;*
- *$\mathcal{M}^q_\rho$: for each $q(\boldsymbol{a})$, if $(\mathcal{T}_1, \mathcal{A}) \models q(\boldsymbol{a})$, then $(\mathcal{M}, \mathcal{A}) \models q(\boldsymbol{a})$, where $\boldsymbol{a}$ is a tuple of individual names from $\mathcal{A}$ and $q(\boldsymbol{a})$ is a Σ-conjunctive query.*

A minimal subsumption (resp. instance, conjunctive) query projection module is a projection module $\mathcal{M}^c_\rho$ (resp. $\mathcal{M}^i_\rho, \mathcal{M}^q_\rho$) minimal w.r.t. $\subsetneq$.

Note that there may exist several, even exponentially many minimal projection modules. It can readily be checked that $\mathsf{cDiff}^{\mathcal{L}}_\Sigma(\mathcal{T}_1, \mathcal{M}^c_\rho) = \emptyset$, for $\mathcal{L} = \mathcal{ELH}^r$, $\mathsf{iDiff}_\Sigma(\mathcal{T}_1, \mathcal{M}^i_\rho) = \emptyset$ and $\mathsf{qDiff}_\Sigma(\mathcal{T}_1, \mathcal{M}^q_\rho) = \emptyset$ (cf. Definition 1).

Example 1. Suppose $\mathcal{T}_1 = \{A_1 \sqsubseteq A_2, A_2 \sqsubseteq A_3\}$, $\mathcal{T}_2 = \{A_1 \sqsubseteq A_3 \sqcap B_1, B_1 \sqsubseteq \exists r.A_3\}$, and the interested vocabulary $\Sigma = \{A_1, A_3, r\}$. $\mathcal{T}_2$ has no logical difference from $\mathcal{T}_1$. However, the concept project module of $\mathcal{T}_2$ with respect to $\mathcal{T}_1$ and Σ is $\{A_1 \sqsubseteq A_3 \sqcap B_1\} \subsetneq \mathcal{T}_2$. This means that a strict sub-ontology of $\mathcal{T}_2$ is sufficient to capture all the information of $\mathcal{T}_1$ about Σ. Moreover, $\mathcal{T}_2$ also entails a consequence $A_1 \sqsubseteq \exists r.A_3$, which is not the case for $\mathcal{T}_1$.

The following example shows that the three notions of projection modules based on different query languages are distinct.

Example 2. Let $\mathcal{T} = \{X \sqsubseteq Y, Y \sqsubseteq \exists t.Z, \mathsf{ran}(r) \sqsubseteq A_1, \mathsf{ran}(s) \sqsubseteq A_2, B \equiv A_1 \sqcap A_2\}$, $\Sigma = \{X, Y, Z, B, r, s\}$, and $\rho = \langle \mathcal{T}, \sigma, \mathcal{T}\rangle$. We have that $\mathcal{M}^c_\rho = \{X \sqsubseteq Y\}$, $\mathcal{M}^i_\rho = \mathcal{M}^c_\rho \cup \{\mathsf{ran}(r) \sqsubseteq A_1, \mathsf{ran}(s) \sqsubseteq A_2, B \equiv A_1 \sqcap A_2\}$ and $\mathcal{M}^q_\rho = \mathcal{T}$.

Definition 3. *Let $Q \in \{c, i, q\}$. The relationship between $\mathcal{T}_1$ and $\mathcal{T}_2$ is a $\langle \Sigma, Q\rangle$-*implementation, *denoted $\mathcal{T}_1 \rightsquigarrow^Q_\Sigma \mathcal{T}_2$, iff there exists a projection module $\mathcal{M}^Q_\rho$ under the setting $\rho = \langle \mathcal{T}_1, \Sigma, \mathcal{T}_2\rangle$.*

If $\mathcal{T}_1 \rightsquigarrow^Q_\Sigma \mathcal{T}_2$, we also say that $\mathcal{T}_2$ $\langle \Sigma, Q\rangle$-implements $\mathcal{T}_1$. In case $\mathcal{T}_1$ and $\mathcal{T}_2$ $\langle \Sigma, Q\rangle$-implement each other, they cannot be separated using the query language Q.

Proposition 1. *Let $\mathcal{T}_1 \rightsquigarrow^Q_\Sigma \mathcal{T}_2$ and $\mathcal{T}_2 \rightsquigarrow^Q_\Sigma \mathcal{T}_1$. Then:*

- *$Q = c$: $\mathsf{cDiff}^{\mathcal{L}}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \mathsf{cDiff}^{\mathcal{L}}_\Sigma(\mathcal{T}_2, \mathcal{T}_1) = \emptyset$, for $\mathcal{L} = \mathcal{ELH}^r$;*
- *$Q = i$: $\mathsf{iDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \mathsf{iDiff}_\Sigma(\mathcal{T}_2, \mathcal{T}_1) = \emptyset$; and*
- *$Q = q$: $\mathsf{qDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \mathsf{qDiff}_\Sigma(\mathcal{T}_2, \mathcal{T}_1) = \emptyset$.*

We obtain the following monotonicity properties of the $\langle \Sigma, Q\rangle$-implementation relation.

Proposition 2 (Implementation Monotonicity).

(i) If $\mathcal{T}_2 \subseteq \mathcal{T}_3$ and $\mathcal{T}_1 \rightsquigarrow^Q_\Sigma \mathcal{T}_2$, then $\mathcal{T}_1 \rightsquigarrow^Q_\Sigma \mathcal{T}_3$;
(ii) If $\mathcal{T}_1 \subseteq \mathcal{T}_2$ and $\mathcal{T}_2 \rightsquigarrow^Q_\Sigma \mathcal{T}_3$, then $\mathcal{T}_1 \rightsquigarrow^Q_\Sigma \mathcal{T}_3$.

Property (i) states that if a terminology $\mathcal{T}_3$ is obtained from $\mathcal{T}_2$ by adding further axioms to it, then it $\langle \Sigma, Q\rangle$-implements all the terminologies $\mathcal{T}_1$ that $\mathcal{T}_2$ $\langle \Sigma, Q\rangle$-implements. Property (ii) states $\mathcal{T}_3$ $\langle \Sigma, Q\rangle$-implements all subsets $\mathcal{T}_1$ of $\mathcal{T}_2$ provided that $\mathcal{T}_3$ $\langle \Sigma, Q\rangle$-implements $\mathcal{T}_2$. We leave investigating certain robustness properties of $\rightsquigarrow^Q_\Sigma$ regarding signature extensions and varying query languages for future work; see, e.g., [21].

4 Computing Minimal Projection Modules

It is shown in [19] that detecting concept inclusion differences formulated in $\mathcal{EL}^{\mathsf{ran}}$ and $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$ is equivalent to detecting a difference with instance and conjunctive queries, respectively. We therefore consider subsumption queries from $\mathcal{EL}^{\mathsf{ran}}$ and $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$ to compute minimal projection justifications for instance and conjunctive queries, respectively.

4.1 Definition of Subsumption Projection Justifications

For computing minimal subsumption projection modules for subsumption queries, we introduce the notion of a subsumption projection justification between two terminologies. As the notion depends on several parameters, we organise them for better readability in a tuple χ of the form $\langle \mathcal{T}_1, X_1, \Sigma, \mathcal{T}_2, X_2, \mathcal{L} \rangle$, where $\mathcal{T}_1$ and $\mathcal{T}_2$ are normalised $\mathcal{ELH}^r$-terminologies, Σ is a signature, $X_1, X_2 \in \mathsf{N_C} \cup \{\, \mathsf{dom}(r), \mathsf{ran}(r) \mid r \in \mathsf{N_R} \,\}$, and $\mathcal{L} \in \{\mathcal{ELH}^r, \mathcal{EL}^{\mathsf{ran}}, \mathcal{EL}^{\mathsf{ran},\sqcap,u}\}$.

To obtain subsumption modules, we use an operator '$\otimes$' to combine sets of role, subsumee and subsumer projection justifications. Given a set $\mathcal{S}$ and sets of sets $\mathbb{S}_1, \mathbb{S}_2 \subseteq 2^{\mathcal{S}}$, we define $\mathbb{S}_1 \otimes \mathbb{S}_2 := \{\, \mathcal{S}_1 \cup \mathcal{S}_2 \mid \mathcal{S}_1 \in \mathbb{S}_1, \mathcal{S}_2 \in \mathbb{S}_2 \,\}$. For instance, if $\mathbb{S}_1 = \{\{\alpha_1, \alpha_2\}, \{\alpha_3\}\}$ and $\mathbb{S}_2 = \{\{\alpha_1, \alpha_3\}, \{\alpha_4, \alpha_5\}\}$, then $\mathbb{S}_1 \otimes \mathbb{S}_2 = \{\{\alpha_1, \alpha_2, \alpha_3\}, \{\alpha_1, \alpha_2, \alpha_4, \alpha_5\}, \{\alpha_3, \alpha_4, \alpha_5\}, \{\alpha_1, \alpha_3\}\}$. For a set $\mathbb{M}$ of sets, we define a function $\mathrm{Minimise}_{\subseteq}(\mathbb{M})$ as follows: $\mathcal{M} \in \mathrm{Minimise}_{\subseteq}(\mathbb{M})$ iff $\mathcal{M} \in \mathbb{M}$ and there does not exist a set $\mathcal{M}' \in \mathbb{M}$ such that $\mathcal{M}' \subsetneq \mathcal{M}$. Continuing with the example, $\mathrm{Minimise}_{\subseteq}(\mathbb{S}_1 \otimes \mathbb{S}_2) = \{\{\alpha_1, \alpha_3\}, \{\alpha_1, \alpha_2, \alpha_4, \alpha_5\}, \{\alpha_3, \alpha_4, \alpha_5\}\}$.

Definition 4 (Subsumption Projection Justification). *Let* $\chi = \langle \mathcal{T}_1, X_1, \Sigma, \mathcal{T}_2, X_2, \mathcal{L} \rangle$. *A set* $\mathcal{M}$ *is a* subsumee module under χ *iff* $\mathcal{M} \subseteq \mathcal{T}_2$ *and for every* $\mathcal{L}_\Sigma$*-inclusion* α*:* $\mathcal{T}_1 \models \mathit{lhs}(\alpha) \sqsubseteq X_1$ *implies* $\mathcal{M} \models \mathit{lhs}(\alpha) \sqsubseteq X_2$*; and* $\mathcal{M}$ *is a* subsumer module under χ *iff* $\mathcal{M} \subseteq \mathcal{T}_2$ *and for every* $\mathcal{L}_\Sigma$*-inclusion* α*:* $\mathcal{T}_1 \models X_1 \sqsubseteq \mathit{rhs}(\alpha)$ *implies* $\mathcal{M} \models X_2 \sqsubseteq \mathit{rhs}(\alpha)$.

$\mathcal{M}$ *is called a* subsumption projection module under χ *iff* $\mathcal{M}$ *is a subsumee and a subsumer projection module under* χ. *A* subsumee (subsumer, subsumption) projection justification under χ *is a subsumee (resp. subsumer, subsumption) projection module under* χ *that is minimal w.r.t.* $\subsetneq$.

We denote the set of all subsumee (resp. subsumer, subsumption) justifications under χ as $\mathbb{J}^{\leftarrow}_{\chi}$ (resp. $\mathbb{J}^{\rightarrow}_{\chi}$, $\mathbb{J}_{\chi}$), where $\chi = \langle \mathcal{T}_1, \varphi_1, \Sigma, \mathcal{T}_2, \varphi_2, \mathcal{L} \rangle$, and $\varphi_1, \varphi_2 \in (\mathsf{N_C} \cap \Sigma) \cup \Sigma^{\mathsf{dom}} \cup \Sigma^{\mathsf{ran}}$.

Definition 5 (Role Subsumption Projection Justification). *Let* $\rho = \langle \mathcal{T}_1, \Sigma, \mathcal{T}_2 \rangle$ *be a projection setting. A set* $\mathcal{M}$ *is called a* role subsumption module under ρ *iff* $\mathcal{M} \subseteq \mathcal{T}_2$ *and for every* $r, s \in \mathsf{N_R} \cap \Sigma$*,* $\mathcal{T}_1 \models r \sqsubseteq s$ *implies* $\mathcal{M} \models r \sqsubseteq s$*. A minimal* role subsumption projection justification under ρ *is the role subsumption module under* ρ *that is minimal w.r.t.* $\subsetneq$.

We denote the set of all role subsumption projection justifications under ρ as $\mathbb{J}^R_\rho$. The following lemma states how role subsumption projection justifications can be computed. The lemma can be shown using Definition 5 and the notion of justification.

Lemma 1. *Let $\rho = \langle \mathcal{T}_1, \Sigma, \mathcal{T}_2 \rangle$ be a projection setting.*

$$\mathbb{J}^R_\rho = Minimize_{\subseteq}\big(\bigotimes_{r,s \in \Sigma \cap \mathsf{N_R}, \mathcal{T}_1 \models r \sqsubseteq s} \mathrm{Just}_{\mathcal{T}_2}(r \sqsubseteq s)\big)$$

Using Definitions 1 and 4, we obtain the following lemma stating the absence of certain concept names, and domain and range concepts over role names as left-hand and right-hand difference witnesses between two terminologies $\mathcal{T}_1$ and $\mathcal{T}_2$.

Lemma 2. *Let $\varphi \in (\mathsf{N_C} \cap \Sigma) \cup \Sigma^{\mathsf{dom}} \cup \Sigma^{\mathsf{ran}}$ and let $A \in \Sigma \cap \mathsf{N_C}$. Additionally, let $\chi = \langle \mathcal{T}_1, \varphi, \Sigma, \mathcal{T}_2, \varphi, \mathcal{L} \rangle$, $\chi' = \langle \mathcal{T}_1, A, \Sigma, \mathcal{T}_2, A, \mathcal{L} \rangle$ and $\mathcal{L} \in \{\mathcal{ELH}^r, \mathcal{EL}^{\mathsf{ran}}, \mathcal{EL}^{\mathsf{ran},\sqcap,u}\}$. Then:*

- $\varphi \notin \mathsf{lhsWtn}^{\mathcal{L}}_\Sigma(\mathcal{T}_1, J_\varphi)$ *for every* $J_\varphi \in \mathbb{J}^{\rightarrow}_\chi$*;*
- $A \notin \mathsf{rhsWtn}^{\mathcal{L}}_\Sigma(\mathcal{T}_1, J_A)$ *for every* $J_A \in \mathbb{J}^{\leftarrow}_{\chi'}$*;*
- $\mathsf{roleWtn}^{\mathcal{L}}_\Sigma(\mathcal{T}_1, J) = \emptyset$ *for every* $J \in \mathbb{J}^R_\rho$*.*

We need at least one subsumption justification, for every potential difference witness, to be contained in a projection module in order to prevent the witness; cf. Lemma 2. This is made precise in the following theorem.

Theorem 1. *Let $\rho = \langle \mathcal{T}_1, \Sigma, \mathcal{T}_2 \rangle$ be a projection setting and let $Q \in \{c, i, q\}$. Additionally, let $\mathbb{M}^Q_\rho$ be the set of all minimal projection modules under ρ for query type Q. Finally, let*

$$\mathbb{S}^{\mathcal{L}}_\rho = Minimize_{\subseteq}\big(\mathbb{J}^R_\rho \otimes \bigotimes_{\varphi \in (\Sigma \cap \mathsf{N_C}) \cup \Sigma^{\mathsf{dom}} \cup \Sigma^{\mathsf{ran}}} \mathbb{J}^{\rightarrow}_{\chi(\varphi,\mathcal{L})} \otimes \bigotimes_{A \in \Sigma \cap \mathsf{N_C}} \mathbb{J}^{\leftarrow}_{\chi(A,\mathcal{L})}\big)$$

where $\chi(\psi, \mathcal{L}) = \langle \mathcal{T}_1, \psi, \Sigma, \mathcal{T}_2, \psi, \mathcal{L} \rangle$, and $\mathcal{L} = \mathcal{ELH}^r$ if $Q = c$, $\mathcal{L} = \mathcal{EL}^{\mathsf{ran}}$ if $Q = i$, and $\mathcal{L} = \mathcal{EL}^{\mathsf{ran},\sqcap,u}$ if $Q = q$. Then it holds that $\mathbb{M}^Q_\rho = \mathbb{S}^{\mathcal{L}}_\rho$.

In this paper, we present an algorithm for computing subsumee projection justifications for subsumption queries of $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$. Recall that $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$-inclusions are sufficient to detect any difference that is detectable with conjunctive queries. The algorithm for computing subsumer projection justifications, and the algorithms for the other query languages are similar.

4.2 Computing Subsumee Projection Justifications

We now present the algorithm for computing subsumee projection justifications (Fig. 3). The basic idea of the algorithm is to collect as few axioms from a terminology $\mathcal{T}_2$ as possible while ensuring the existence of a so-called subsumee simulation between another terminology $\mathcal{T}_1$ and $\mathcal{T}_2$ [12,26].

Algorithm 1: Computing all Subsumee Projection Justifications

```
function $\textsc{Cover}_{\leftarrow}$ $(\mathcal{T}_1, X_1, \Sigma, \mathcal{T}_2, X_2, \zeta)$
    if $X_1$ is not $\Sigma$-entailed w.r.t. $\mathcal{T}_1$ then
        return $\{\emptyset\}$
    $\mathbb{M}^{\leftarrow}_{(X_1,X_2)} := \textsc{Cover}^{\mathsf{N_C}}_{\leftarrow}(\mathcal{T}_1, X_1, \Sigma, \mathcal{T}_2, X_2, \zeta)$
    if $X_1$ is not complex $\Sigma$-entailed in $\mathcal{T}_1$ then
        return $\mathbb{M}^{\leftarrow}_{(X_1,X_2)}$
    if $X_1 \equiv \exists r.Y \in \mathcal{T}_1$, and $r, Y$ are $\Sigma$-entailed
      w.r.t. $\mathcal{T}_1$ then
        $\mathbb{M}^{\leftarrow}_{(X_1,X_2)} :=$
          $\mathbb{M}^{\leftarrow}_{(X_1,X_2)} \otimes \textsc{Cover}^{\exists}_{\leftarrow}(\mathcal{T}_1, X_1, \Sigma, \mathcal{T}_2, X_2, \zeta)$
    else if $X_1 \equiv Y_1 \sqcap \ldots \sqcap Y_m \in \mathcal{T}_1$ then
        $\mathbb{M}^{\leftarrow}_{(X_1,X_2)} :=$
          $\mathbb{M}^{\leftarrow}_{(X_1,X_2)} \otimes \textsc{Cover}^{\sqcap}_{\leftarrow}(\mathcal{T}_1, X_1, \Sigma, \mathcal{T}_2, X_2, \zeta)$
    return $\text{Minimise}_{\subseteq}(\mathbb{M}^{\leftarrow}_{(X_1,X_2)})$
```

Algorithm 2: Computing all Subsumee Projection Justifications ($S^{\leftarrow}_{\exists}$)

```
function $\textsc{Cover}^{\exists}_{\leftarrow}$ $(\mathcal{T}_1, X_1, \Sigma, \mathcal{T}_2, X_2, \zeta)$
    let $\alpha_{X_1} := X_1 \equiv \exists r.Y_1 \in \mathcal{T}_1$
    $\mathbb{M}^{\leftarrow}_{(X_1,X_2)} := \{\text{max-tree}^{\sqcap}_{\mathcal{T}_2}(X_2)\}$
    for every $s \in \mathsf{N_R} \cap \Sigma$ such that $\mathcal{T}_1 \models s \sqsubseteq r$
      do
        for every $X_2' \in \text{non-conj}_{\mathcal{T}_2}(X_2)$ such that
          $\zeta \neq \varepsilon$ implies $\mathcal{T}_2 \not\models \mathsf{ran}(\zeta) \sqsubseteq X_2'$ and
          $\mathcal{T}_2 \not\models \mathsf{dom}(s) \sqsubseteq X_2'$ do
            let $\alpha_{X_2'} := X_2' \equiv \exists r'.Y_2' \in \mathcal{T}_2$
              $\mathbb{M}^{\leftarrow}_{Y_2'} := \textsc{Cover}_{\leftarrow}(\mathcal{T}_1, Y_1, \Sigma, \mathcal{T}_2, Y_2', s)$
              $\mathbb{M}^{\leftarrow}_{(X_1,X_2)} := \mathbb{M}^{\leftarrow}_{(X_1,X_2)}$
              $\otimes \big(\{\{\alpha_{X_2'}\}\} \otimes \text{Just}_{\mathcal{T}_2}(s \sqsubseteq r) \otimes \mathbb{M}^{\leftarrow}_{Y_2'}\big)$
    return $\mathbb{M}^{\leftarrow}_{(X_1,X_2)}$
```

Algorithm 3: Computing all Subsumee Projection Justifications ($S^{\leftarrow}_{\mathsf{N_C}}$)

```
function $\textsc{Cover}^{\mathsf{N_C}}_{\leftarrow}$ $(\mathcal{T}_1, X_1, \Sigma, \mathcal{T}_2, X_2, \zeta)$
    $\mathbb{M}^{\leftarrow}_{(X_1,X_2)} := \{\emptyset\}$
    for every $B \in \Sigma^{\zeta}$ such that $\mathcal{T}_1 \models B \sqsubseteq X_1$ do
        for every $X_2 \in$ non-conj$_{\mathcal{T}_2}(X_1)$ such that
          $\zeta = \varepsilon$ or $\mathcal{T}_2 \models \mathsf{ran}(\zeta) \sqsubseteq X_2$ do
            $\mathbb{M}^{\leftarrow}_{(X_1,X_2)} := \mathbb{M}^{\leftarrow}_{(X_1,X_2)} \otimes \text{Just}_{\mathcal{T}_2}(B \sqsubseteq X_2)$
    return $\mathbb{M}^{\leftarrow}_{(X_1,X_2)}$
```

Algorithm 4: Computing all Subsumee Projection Justifications ($S^{\leftarrow}_{\sqcap}$)

```
function $\textsc{Cover}^{\sqcap}_{\leftarrow}$ $(\mathcal{T}_1, X_1, \Sigma, \mathcal{T}_2, X_2, \zeta)$
    let $\alpha_{X_1} := X_1 \equiv Y_1 \sqcap \ldots \sqcap Y_m \in \mathcal{T}_1$
    $\mathbb{M}^{\leftarrow}_{(X_1,X_2)} := \emptyset$
    for every $\Gamma \in \text{DefForest}^{\sqcap}_{\mathcal{T}_2}(X_2)$ do
        let $\delta_{\Gamma} := \{\, \text{def}^{\sqcap}_{\mathcal{T}_2}(X') \mid X' \in$
          $\text{leaves}(\Gamma) \cap \text{def}^{\sqcap}_{\mathcal{T}_2} \,\}$
        $\mathbb{M}^{\leftarrow}_{\Gamma} := \{\Gamma\}$
        for every $X_2' \in \text{leaves}(\Gamma)$ such that $\zeta = \varepsilon$
          or $\mathcal{T}_2 \not\models \mathsf{ran}(\zeta) \sqsubseteq X_2'$ do
            $\mathbb{M}^{\leftarrow}_{X_2'} := \emptyset$
            for every $X_1' \in \text{non-conj}_{\mathcal{T}_1}(X_1)$ such
              that $\zeta = \varepsilon$ or $\mathcal{T}_2 \not\models \mathsf{ran}(\zeta) \sqsubseteq X_1'$ do
                if $\langle \mathcal{T}_1, X_1' \rangle \sim^{\leftarrow}_{\Sigma,\zeta} \langle \mathcal{T}_2 \setminus \delta_{\Gamma}, X_2' \rangle$ then
                    $\mathbb{M}^{\leftarrow}_{X_2'} := \mathbb{M}^{\leftarrow}_{X_2'} \cup$
                      $\textsc{Cover}_{\leftarrow}(\mathcal{T}_1, X_1', \Sigma, \mathcal{T}_2 \setminus \delta_{\Gamma}, X_2', \epsilon)$
            $\mathbb{M}^{\leftarrow}_{\Gamma} := \mathbb{M}^{\leftarrow}_{\Gamma} \otimes \mathbb{M}^{\leftarrow}_{X_2'}$
        $\mathbb{M}^{\leftarrow}_{(X_1,X_2)} := \mathbb{M}^{\leftarrow}_{(X_1,X_2)} \cup \mathbb{M}^{\leftarrow}_{\Gamma}$
    return $\mathbb{M}^{\leftarrow}_{(X_1,X_2)}$
```

Fig. 3. Algorithms for computing all subsumee justifications

Inclusions of the form $\mathsf{ran}(r) \sqsubseteq X$ might cause non-trivial entailments. For example, let $\mathcal{T}_1 = \{X \equiv \exists r.Y, Y \equiv A_1 \sqcap A_2\}$ and $\Sigma = \{X, A_1, A_2, r\}$. Then $\mathcal{T}_1$ entails that X is subsumed by the Σ-concepts $\exists r.\top$ and $\exists r.(A_1 \sqcap A_2)$ (modulo equivalence). For $\mathcal{T}_2 = \mathcal{T}_1 \cup \{\mathsf{ran}(r) \sqsubseteq A_1\}$, however, we additionally obtain $\mathcal{T}_2 \models \exists r.A_2 \sqsubseteq X$. Hence, when formulating the algorithms for computing subsumee simulations, an additional parameter $\zeta \in \{\epsilon\} \cup (\mathsf{N_R} \cap \Sigma)$ is needed which is used in range concepts of the form $\mathsf{ran}(\zeta)$. We call this parameter *context of a role.* We treat ϵ as a special role name and set $\mathsf{ran}(\epsilon) = \top$. The set of all role contexts, in symbols $\mathcal{C}^{\Sigma}$, is defined as $\mathcal{C}^{\Sigma} = \{\epsilon\} \cup (\mathsf{N_R} \cap \Sigma)$.

To identify concept and role names that are relevant for a subsumee simulation that we propose later, we first use the following notion of Σ-entailment:

- $A \in \mathsf{N_C}$ is *Σ-entailed in $\mathcal{T}$* iff there is an $\mathcal{EL}^{\mathsf{ran}}_{\Sigma}$-concept C such that $\mathcal{T} \models C \sqsubseteq A$;
- $s \in \mathsf{N_R}$ is *Σ-entailed in $\mathcal{T}$* iff there exists $s' \in \mathsf{N_R} \cap \Sigma$ such that $\mathcal{T} \models s' \sqsubseteq s$;
- $A \in \mathsf{N_C}$ is *(Σ, s)-entailed in $\mathcal{T}$* iff there is an $\mathcal{EL}^{\mathsf{ran}}_{\Sigma}$-concept C such that $\mathcal{T} \models C \sqcap \mathsf{ran}(s) \sqsubseteq A$.

Moreover, we say that $X \in \mathsf{N_C}$ is *complex Σ-entailed* w.r.t. $\mathcal{T}$ iff for every $Y \in \text{non-conj}_{\mathcal{T}}(X)$ one of the following conditions holds:

– there exists $B \in \Sigma$ such that $\mathcal{T} \models B \sqsubseteq Y$ and $\mathcal{T} \not\models B \sqsubseteq X$;
– there exists $Y \equiv \exists r.Z \in \mathcal{T}$ and r, Z are Σ-entailed in $\mathcal{T}$.

X is said to be *simply Σ-entailed* if X is Σ-entailed but not complex Σ-entailed. For example, let $\mathcal{T} = \{X \equiv X_1 \sqcap X_2,\ B_1 \sqsubseteq X_1,\ X_2 \equiv \exists r.Z,\ B_2 \sqsubseteq Z, s \sqsubseteq r\}$. We have that $\text{non-conj}_{\mathcal{T}}(X) = \{X_1, X_2\}$, then r is Σ-entailed w.r.t. $\mathcal{T}$; X is complex Σ-entailed w.r.t. $\mathcal{T}$ for $\Sigma = \{B_1, B_2, s\}$; but X is not complex Σ'-entailed w.r.t. $\mathcal{T}$, where Σ' ranges over $\{B_1, B_2\}$, $\{B_1, s\}$, $\{B_2, s\}$. Additionally, X is not complex Σ-entailed w.r.t. $\mathcal{T} \cup \{B_1 \sqsubseteq X\}$.

We now define the notion of a *subsumee simulation* from $\mathcal{T}_1$ to $\mathcal{T}_2$ as a subset of $(\mathsf{N_C} \cap \text{sig}(\mathcal{T}_1)) \times (\mathsf{N_C} \cap \text{sig}(\mathcal{T}_2)) \times \mathcal{C}^{\Sigma}_{\mathcal{T}_1}$, where $\mathcal{C}^{\Sigma}_{\mathcal{T}_1} := \{\epsilon\} \cup (\mathsf{N_R} \cap (\Sigma \cup \text{sig}(\mathcal{T}_1)))$ is the range of role contexts.

Definition 6 (Subsumee Simulation). *A relation $S \subseteq sig^{\mathsf{N_C}}(\mathcal{T}_1) \times sig^{\mathsf{N_C}}(\mathcal{T}_2) \times \mathcal{C}^{\Sigma}_{\mathcal{T}_1}$ is a Σ-*subsumee simulation from $\mathcal{T}_1$ to $\mathcal{T}_2$ *iff the following conditions hold:*

($S^{\leftarrow}_{\mathsf{N_C}}$) *if $(X_1, X_2, \zeta) \in S$, then for every $\varphi \in \Sigma^{\zeta}$ and for every $X_2' \in$ non-conj$_{\mathcal{T}_2}(X_2)$ with $\mathcal{T}_2 \not\models \mathsf{ran}(\zeta) \sqsubseteq X_2'$, $\mathcal{T}_1 \models \varphi \sqsubseteq X_1$ implies $\mathcal{T}_2 \models \varphi \sqsubseteq X_2'$;*

($S^{\leftarrow}_{\exists}$) *if $(X_1, X_2, \zeta) \in S$ and $X_1 \equiv \exists r.Y_1 \in \mathcal{T}_1$ such that $\mathcal{T}_1 \models s \sqsubseteq r$ for $s \in \Sigma$ and Y_1 is (Σ, s)-entailed in $\mathcal{T}_1$, then for every $X_2' \in$ non-conj$_{\mathcal{T}_2}(X_2)$ not entailed by $\mathsf{dom}(s)$ or $\mathsf{ran}(\zeta)$ w.r.t. $\mathcal{T}_2$, there exists $X_2' \equiv \exists r'.Y_2 \in \mathcal{T}_2$ such that $\mathcal{T}_2 \models s \sqsubseteq r'$ and $(Y_1, Y_2, s) \in S$;*

($S^{\leftarrow}_{\sqcap}$) *if $(X_1, X_2, \zeta) \in S$ and $X_1 \equiv Y_1 \sqcap Y_2 \sqcap \ldots \sqcap Y_n \in \mathcal{T}_1$, then for every $Y_2 \in$ non-conj$_{\mathcal{T}_2}(X_2)$ not entailed by $\mathsf{ran}(\zeta)$ w.r.t. $\mathcal{T}_2$, there exists $Y_1 \in$ non-conj$_{\mathcal{T}_1}(X_1)$ not entailed by $\mathsf{ran}(\zeta)$ w.r.t. $\mathcal{T}_2$ with $(Y_1, Y_2, \epsilon) \in S$.*

We write $\mathcal{T}_1 \sim^{\leftarrow}_{\Sigma} \mathcal{T}_2$ iff there is a Σ-subsumee simulation S from $\mathcal{T}_1$ to $\mathcal{T}_2$ such that for every $A \in \mathsf{N_C} \cap \Sigma$: $(A, A, \epsilon) \in S$.

For $\zeta \in \Sigma \cap \mathsf{N_R}$, we write $\langle \mathcal{T}_1, X_1 \rangle \sim^{\leftarrow}_{\Sigma,\zeta} \langle \mathcal{T}_2, X_2 \rangle$ iff there is a Σ-subsumee simulation S from $\mathcal{T}_1$ to $\mathcal{T}_2$ with $(X_1, X_2, \zeta) \in S$ for which $\mathcal{T}_1 \sim^{\leftarrow}_{\Sigma} \mathcal{T}_2$.

A subsumee simulation captures the set of subsumees in the sense that $\mathcal{T}_1 \sim^{\leftarrow}_{\Sigma} \mathcal{T}_2$ iff $\mathsf{rhsWtn}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$. Moreover, if a concept name X_2 in $\mathcal{T}_2$ Σ-subsumee simulates a concept name X_1 in $\mathcal{T}_1$, then X_2 subsumes all Σ-concepts w.r.t. $\mathcal{T}_2$ that are subsumed by X_1 w.r.t. $\mathcal{T}_1$. Formally: $\langle \mathcal{T}_1, X_1 \rangle \sim^{\leftarrow}_{\Sigma,\zeta} \langle \mathcal{T}_2, X_2 \rangle$ iff for every $C \in \mathcal{EL}^{\mathsf{ran}}_{\Sigma}$: $\mathcal{T}_1 \models C \sqsubseteq X_1$ implies $\mathcal{T}_2 \models C \sqsubseteq X_2$ [26].

Algorithm 1 provides the function $\textsc{Cover}_{\leftarrow}$ for computing the set of all subsumee justifications. The algorithm recursively computes sets of axioms sufficient to construct a subsumee simulation. For better readability, the algorithm is structured into several parts, one for each condition of a subsumee simulation, cf. Definition 6. Algorithm 3 handles Case ($S^{\leftarrow}_{\mathsf{N_C}}$), Algorithm 2 takes care of Case ($S^{\leftarrow}_{\exists}$) and Algorithm 4 is responsible for Case ($S^{\leftarrow}_{\sqcap}$). Note that each of these algorithms requires a role context ζ as an input parameter. The notion of complex Σ-entailment is employed in Algorithm 1. If X is not complex Σ-entailment, then neither the existential nor the conjunctive case need to be considered, and Algorithm 1 terminates in Line 6.

Compared with computing subsumer projection justifications, the challenge of computing subsumee projection justifications is to handle conjunctions on the left-hand side of subsumptions. Concept names defined as conjunctions in $\mathcal{T}_2$ use conjuncts which in turn may also be defined as conjunctions. Such axioms form tree structures. When selecting axioms, all minimal subsets of $\mathcal{T}_2$, i.e., all sub-trees, that maintain a subsumee simulation need to be considered. To this end, we define for each concept name X a so-called *definitorial forest* consisting of sets of axioms of the form $Y \equiv Y_1 \sqcap \ldots \sqcap Y_n$ which can be thought of as forming *trees*. Any $\langle X, \Sigma\rangle$-subsumee projection justification contains the axioms of a selection of these trees, i.e., one tree for every conjunction formulated over Σ that entails X w.r.t. $\mathcal{T}$. Formally, we define a set of a $\text{DefForest}^{\sqcap}_{\mathcal{T}}(X) \subseteq 2^{\mathcal{T}}$ to be the smallest set closed under the following conditions:

- $\emptyset \in \text{DefForest}^{\sqcap}_{\mathcal{T}}(X)$;
- $\{\alpha\} \in \text{DefForest}^{\sqcap}_{\mathcal{T}}(X)$ for $\alpha := X \equiv X_1 \sqcap \ldots \sqcap X_n \in \mathcal{T}$; and
- $\Gamma \cup \{\alpha\} \in \text{DefForest}^{\sqcap}_{\mathcal{T}}(X)$ for $\Gamma \in \text{DefForest}^{\sqcap}_{\mathcal{T}}(X)$ with $Z \equiv Z_1 \sqcap \ldots \sqcap Z_k \in \Gamma$ and $\alpha := Z_i \equiv Z_i^1 \sqcap \ldots \sqcap Z_i^n \in \mathcal{T}$.

Given a tree $\Gamma \in \text{DefForest}^{\sqcap}_{\mathcal{T}}(X)$ rooted at X, we use leaves(Γ) to denote the set $\text{sig}(\Gamma) \setminus \{\, X \in \text{sig}(C) \mid X \equiv C \in \Gamma \,\}$ of leaves if $\Gamma \neq \emptyset$; and $\{X\}$ otherwise. We denote with $\text{max-tree}^{\sqcap}_{\mathcal{T}}(X)$ the set in $\text{DefForest}^{\sqcap}_{\mathcal{T}}(X)$ that is maximal w.r.t. $\subseteq$. Finally, we set $\text{non-conj}_{\mathcal{T}}(X) := \text{leaves}(\text{max-tree}^{\sqcap}_{\mathcal{T}}(X))$ to be the set of leaves of the maximal tree. For example, for $\mathcal{T} = \{\alpha_1, \alpha_2, \alpha_3\}$ with $\alpha_1 := X \equiv Y \sqcap Z$, $\alpha_2 := Y \equiv Y_1 \sqcap Y_2$, and $\alpha_3 := Z \equiv Z_1 \sqcap Z_2$, we obtain $\text{DefForest}^{\sqcap}_{\mathcal{T}}(X) = \{\emptyset, \{\alpha_1\}, \{\alpha_1, \alpha_2\}, \{\alpha_1, \alpha_3\}, \{\alpha_1, \alpha_2, \alpha_3\}\}$. Moreover, we have that $\text{leaves}(\{\alpha_1, \alpha_3\}) = \{Y, Z_1, Z_2\}$, $\text{max-tree}^{\sqcap}_{\mathcal{T}}(X) = \{\alpha_1, \alpha_2, \alpha_3\}$, and $\text{non-conj}_{\mathcal{T}}(X) = \{Y_1, Y_2, Z_1, Z_2\}$.

The definitorial forest is used to enumerate and find all trees for which Case $(S^{\leftarrow}_{\sqcap})$ holds, which is done in Algorithm 4. The set $\text{non-conj}_{\mathcal{T}}(X)$, however, is also used in Algorithm 2, which we discuss next. The existence of axiom $\alpha_{X_1} := X_1 \equiv \exists r.Y_1 \in \mathcal{T}_1$ in Line 2 of Algorithm 2 is guaranteed by Line 7 of Algorithm 1. The axiom $\alpha_{X_2'} := X_2' \equiv \exists r'.Y_2' \in \mathcal{T}_2$ in Line 6 of Algorithm 2 exists as we assume that $\mathcal{T}_2, X_2$ subsumee-simulates $\mathcal{T}_1, X_1$ w.r.t. Σ. Moreover, there is at most one axiom $\alpha_{X_1} \in \mathcal{T}_1$ and at most one $\alpha_{X_2'} \in \mathcal{T}_2$ as $\mathcal{T}_1$ and $\mathcal{T}_2$ are terminologies. The concept name X_2 may be defined as a conjunction in $\mathcal{T}_2$ whose conjuncts in turn may also be defined as a conjunction in $\mathcal{T}_2$ and so forth. In Line 3 all axioms forming the maximal resulting definitorial conjunctive tree are collected.

For the next algorithm, we define $\text{def}^{\sqcap}_{\mathcal{T}} := \{\, X \in \text{sig}^{\mathsf{N_C}}(\mathcal{T}) \mid X \equiv Y_1 \sqcap \ldots \sqcap Y_n \in \mathcal{T} \,\}$ to be the set of concept names that are conjunctively defined in $\mathcal{T}$. For every $X \in \text{def}^{\sqcap}_{\mathcal{T}}$, we set $\text{def}^{\sqcap}_{\mathcal{T}}(X) := \alpha$, where $\alpha = X \equiv Y_1 \sqcap \ldots \sqcap Y_n \in \mathcal{T}$.

The axiom $\alpha_{X_1} := X_1 \equiv Y_1 \sqcap \ldots \sqcap Y_m \in \mathcal{T}_1$ in Line 2 of Algorithm 4 is guaranteed to exist by Line 9 of Algorithm 4. In case X_2 is defined as a conjunction in $\mathcal{T}_2$, the pair consisting of $\mathcal{T}_2$ containing only a partial conjunctive tree rooted at X_2 and X_2 needs to be considered to be sufficient to subsumee simulate $\mathcal{T}_1, X_1$. Therefore Algorithm 4 considers every partial conjunctive tree Γ from $\mathrm{DefForest}^{\sqcap}_{\mathcal{T}_2}(X_2)$ in Line 4 and removes the axioms in δ_Γ connecting the leaves of Γ with the remaining conjunctive tree from $\mathcal{T}_2$ in lines 10 and 11.

The following theorem states that Algorithm 1 indeed computes the set of subsumee projection justifications. The proof establishes that Algorithm 1 computes all possible subsets of $\mathcal{T}_2$ that are minimal w.r.t. $\subsetneq$ while preserving one of the considered Σ-subsumee simulations from $\mathcal{T}_1$ to $\mathcal{T}_2$.

Theorem 2. *Let* $\chi = \langle \mathcal{T}_1, \varphi_1, \Sigma, \mathcal{T}_2, \varphi_2, \mathcal{EL}^{\mathsf{ran},\sqcap,u}\rangle$, *and* $\varphi_1, \varphi_2 \in (\mathsf{N_C} \cap \Sigma) \cup \{\mathsf{dom}(r), \mathsf{ran}(r) \mid r \in \mathsf{N_R} \cap \Sigma\}$. *Additionally, let* $\mathbb{M} := \textsc{Cover}_{\leftarrow}(\mathcal{T}_1, \varphi_1, \Sigma, \mathcal{T}_2, \varphi_2, \epsilon)$ *using Algorithm 1. Then* $\mathbb{M}$ *is the set of all subsumee justifications under* χ.

Algorithm 1 runs in exponential time in the number of axioms contained in the input terminologies, in the worst case. On the one hand, the algorithm uses justifications (see Line 6 of Alg. 2 and Line 5 of Alg. 3) whose number grows exponentially for role inclusions as well as concept name inclusions. The different justifications are each incorporated using the operator $\otimes$ resulting in possibly different subsumption justifications. The majority of the running time will be spent on computing justifications. Another source of exponential blowup is contained in Line 4 of Algorithm 4. The number of elements in the set $\mathrm{DefForest}^{\sqcap}_{\mathcal{T}}(X)$ grows exponentially in $|\mathcal{T}|$. According to our experience so far, however, it seems plausible to assume that definitorial forests in practical ontologies remain rather small and, thus, they do not cause a serious slowdown of the algorithm.

5 Application of Minimal Projection Modules

In this section, we discuss two applications of minimal projection modules.

5.1 Computing Minimal Query Modules

We first define the minimal query modules for different queries.

Definition 7 (Query Module). *A set* $M \subseteq \mathcal{T}$ *is a* subsumption (resp. instance, conjunctive) *query module of* $\mathcal{T}$, *denoted as* M^c_Σ *(resp.* M^i_Σ, M^q_Σ*).*

- M^c_Σ: *for each* $\mathcal{ELH}^r_\Sigma$*-inclusion* α, *if* $\mathcal{T} \models \alpha$, *then* $M \models \alpha$;
- M^i_Σ: *for each* Σ*-instance assertion* λ, *if* $(\mathcal{T}, \mathcal{A}) \models \lambda$, *then* $(M, \mathcal{A}) \models \lambda$;
- M^q_Σ: *for each* $q(\boldsymbol{a})$, *if* $(\mathcal{T}, \mathcal{A}) \models q(\boldsymbol{a})$, *then* $(M, \mathcal{A}) \models q(\boldsymbol{a})$, *where* $\boldsymbol{a}$ *is a tuple of individual names from* $\mathcal{A}$ *and* $q(\boldsymbol{a})$ *is a* Σ*-conjunctive query.*

A subsumption (resp. instance, conjunctive) query module is called a *minimal* subsumption (resp. instance, conjunctive) query module iff it is minimal w.r.t. $\subsetneq$.

In general, the reference and the implementing/task ontologies do not coincide. Intuitively, the task ontology $\mathcal{T}_2$ might contain more knowledge about Σ than the reference ontology $\mathcal{T}_1$. The following lemma illustrates the relationship between minimal projection modules and minimal query modules.

Lemma 3. *Let $Q \in \{c, i, q\}$ and $\rho = \{\mathcal{T}_1, \Sigma, \mathcal{T}_2\}$. Then: for every minimal projection module $\mathcal{M}^Q_\rho$ for a query type Q and under a projection setting ρ, there exists a minimal Q-query module M^Q_Σ of $\mathcal{T}_2$ such that $\mathcal{M}^Q_\rho \subseteq M^Q_\Sigma$.*

Example 3 (Fig. 2 contd.). The minimal projection module of $\mathcal{T}_2$ under $\rho = \{\mathcal{T}_1, \Sigma, \mathcal{T}_2\}$ is $\{\{\beta_1, \beta_3\}, \{\beta_1, \beta_4\}\}$, for any query type $Q \in \{c, i, q\}$. The minimal Q-query module of $\mathcal{T}_2$ w.r.t. Σ is $\{\{\beta_1, \beta_2, \beta_3\}, \{\beta_1, \beta_2, \beta_4\}\}$. The minimal Q-query module of $\mathcal{T}_1$ is $\{\alpha_1\}$.

One solution for importing Σ-knowledge of a reference ontology to a task ontology is to import a minimal Q-query module of the reference ontology. However, one can see that if we include α_1 to $\mathcal{T}_2$, then α_1 repeats the Σ-knowledge that is already represented by β_1. Besides, the resulting ontology would not be a terminology anymore.

Consider a special projection setting of the form $\langle \mathcal{T}, \Sigma, \mathcal{T} \rangle$, where the reference ontology $\mathcal{T}$ is also the implementing ontology. We denote such reflexive projection settings with $\rho^{\circlearrowleft}$. A projection module $\mathcal{M}$ of $\mathcal{T}$ under $\rho^{\circlearrowleft}$ for subsumption (resp. instance, conjunctive) queries is a subset of $\mathcal{T}$ that preserves the answers to Σ-concept subsumption (resp. instance, conjunctive) queries as given by $\mathcal{T}$. It can readily be verified that a minimal projection module under a reflexive projection setting coincides with a *minimal module* for the type of queries considered.

5.2 Ontology Comparison Measure

In existent methods for measuring the entailment capacity of a terminology about a signature Σ for a query language, one can use logical difference. However, the following example shows that using logical difference can be not sufficient in some case.

Example 4. Let $\alpha_1 := A \sqsubseteq B_1 \sqcap B_2 \sqcap B_3 \sqcap B_4$, $\alpha_2 := B_1 \sqsubseteq B_2 \sqcap B_3$, $\alpha_3 := B_2 \sqsubseteq B_4$, $\alpha_4 := B_3 \sqsubseteq B_4$. Let $\mathcal{T}_1 = \{\alpha_1\}$, $\mathcal{T}_2 = \{\alpha_1, \alpha_2\}$, $\mathcal{T}_3 = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ and $\Sigma = \{A, B_1, B_2, B_3, B_4\}$. We have that $\mathsf{Wtn}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) = \mathsf{Wtn}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_3) = (\emptyset, \emptyset, \emptyset)$, for $\mathcal{L} \in \{\mathcal{ELH}^r, \mathcal{EL}^{\mathsf{ran}}, \mathcal{EL}^{\mathsf{ran},\sqcap,u}\}$.

In Example 4, the notion of logical difference cannot be used to distinguish between $\mathcal{T}_2$ and $\mathcal{T}_3$ w.r.t. Σ as $\mathcal{T}_2$ and $\mathcal{T}_3$ preserve the Σ-knowledge w.r.t. $\mathcal{T}_1$. However, intuitively, $\mathcal{T}_2$ and $\mathcal{T}_3$ each contain more information about the Σ-concept names B_1, B_2 and B_4 than $\mathcal{T}_1$. We therefore propose a new measure based on the notions of minimal projection module and query module for different query languages.

Definition 8 (Projection Rate). *Let $Q \in \{c, i, q\}$ and let $\mathcal{M}^Q_\rho$ range over minimal projection modules under $\rho = \langle \mathcal{T}_1, \Sigma, \mathcal{T}_2 \rangle$ and the query type Q. Additionally, let M^Q_Σ range over minimal modules of $\mathcal{T}_2$ for the query type Q. The projection rate P^Q of $\mathcal{T}_1$ over $\mathcal{T}_2$ is defined as:*

$$P^Q = \frac{|\bigcup \mathcal{M}^Q_\rho|}{|\bigcup M^Q_\Sigma|}$$

Note that $p \leq 1$ always holds by Lemma 3. Intuitively, the lower the projection rate, the more Σ-knowledge is contained in $\mathcal{T}_2$ compared with $\mathcal{T}_1$.

Example 5 (Example 4 contd.). Considering when $Q = c$, we have that $\mathcal{M}_{\rho}^{c}$=$\mathcal{M}_{\rho'}^{c}$=$\{\alpha_1\}$ under $\rho = \langle \mathcal{T}_1, \Sigma, \mathcal{T}_2 \rangle$ and $\rho' = \langle \mathcal{T}_1, \Sigma, \mathcal{T}_3 \rangle$. The minimal subsumption query module of $\mathcal{T}_2$ w.r.t. Σ is $\{\alpha_1, \alpha_2\}$. But there exists two minimal subsumption query module of $\mathcal{T}_3$ w.r.t. Σ, which are $\{\alpha_1, \alpha_2, \alpha_3\}$ and $\{\alpha_1, \alpha_2, \alpha_4\}$. So the union of minimal subsumption query module of $\mathcal{T}_3$ w.r.t. Σ is $\mathcal{T}_3$ that contains four axioms. Therefore, the projection rate P^c of $\mathcal{T}_1$ over $\mathcal{T}_2$ is $P^c = 1/2$ and the projection rate P^c of $\mathcal{T}_1$ over $\mathcal{T}_3$ is $P^c = 1/4$. So $\mathcal{T}_3$ contains more Σ-knowledge compared with $\mathcal{T}_2$ as $1/4 < 1/2$.

6 Conclusion

We proposed a novel module notion called projection module that entails the queries that follow from a reference ontology. We presented an algorithm for computing all minimal projection modules of acyclic $\mathcal{ELH}^r$-terminologies and two applications of minimal projection modules. We expect that the algorithms can be extended to deal with cyclic terminologies and possibly general $\mathcal{ELH}^r$-TBoxes, and to yield a ranking between different projection modules, e.g., via weighted signatures.

References

1. Alsubait, T., Parsia, B., Sattler, U.: Measuring similarity in ontologies: a new family of measures. In: Janowicz, K., Schlobach, S., Lambrix, P., Hyvönen, E. (eds.) EKAW 2014. LNCS (LNAI), vol. 8876, pp. 13–25. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13704-9_2
2. Arif, M.F., Mencía, C., Ignatiev, A., Manthey, N., Peñaloza, R., Marques-Silva, J.: BEACON: an efficient SAT-based tool for debugging $\mathcal{EL}^+$ ontologies. In: Creignou, N., Le Berre, D. (eds.) SAT 2016. LNCS, vol. 9710, pp. 521–530. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40970-2_32
3. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications, 2 edn. Cambridge University Press, June 2010
4. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logic. Cambridge University Press, Cambridge (2017)
5. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic $\mathcal{EL}$. In: Proceedings of DL 2007: The 20th International Workshop on Description Logics (2007)
6. Botoeva, E., Konev, B., Lutz, C., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: Inseparability and conservative extensions of description logic ontologies: a survey. In: Pan, J.Z., et al. (eds.) Reasoning Web 2016. LNCS, vol. 9885, pp. 27–89. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-49493-7_2
7. Botoeva, E., Kontchakov, R., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: Games for query inseparability of description logic knowledge bases. Artif. Intell. **234**, 78–119 (2016)

8. Botoeva, E., Lutz, C., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: Query-based entailment and inseparability for ALC ontologies. In: Proceedings of IJCAI 2016: The 25th International Joint Conference on Artificial Intelligence, pp. 1001–1007. AAAI Press (2016)
9. Chen, J., Ludwig, M., Ma, Y., Walther, D.: Zooming in on ontologies: minimal modules and best excerpts. In: d'Amato, C., et al. (eds.) ISWC 2017. LNCS, vol. 10587, pp. 173–189. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68288-4_11
10. Chen, J., Ludwig, M., Walther, D.: Computing minimal subsumption modules of ontologies. In: Proceedings of GCAI 2018: The 4th Global Conference on Artificial Intelligence, pp. 41–53 (2018)
11. Domingue, J., Anutariya, C. (eds.): ASWC 2008. LNCS, vol. 5367. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89704-0
12. Ecke, A., Ludwig, M., Walther, D.: The concept difference for $\mathcal{EL}$-terminologies using hypergraphs. In: Proceedings of DChanges 2013: The International workshop on (Document) Changes: Modeling, Detection, Storage and Visualization (2013)
13. Euzenat, J., Shvaiko, P.: Ontology Matching, 2nd edn. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38721-0
14. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: theory and practice. J. Artif. Intell. Res. **31**(1), 273–318 (2008)
15. Jung, J.C., Lutz, C., Martel, M., Schneider, T.: Query conservative extensions in horn description logics with inverse roles. In: Proceedings of IJCAI 2017: The 26th International Joint Conference on Artificial Intelligence, pp. 1116–1122. AAAI Press (2017)
16. Kalyanpur, A., Parsia, B., Sirin, E., Hendler, J.A.: Debugging unsatisfiable classes in OWL ontologies. J. Web Semant. **3**(4), 268–293 (2005)
17. Kazakov, Y., Skocovsky, P.: Enumerating justifications using resolution. In: Proceedings of DL 2017: The 30th International Workshop on Description Logics (2017)
18. Konev, B., Kontchakov, R., Ludwig, M., Schneider, T., Wolter, F., Zakharyaschev, M.: Conjunctive query inseparability of OWL 2 QL TBoxes. In: Proceedings of AAAI 2011: The 25th Conference on Artificial Intelligence. AAAI Press (2011)
19. Konev, B., Ludwig, M., Walther, D., Wolter, F.: The logical difference for the lightweight description logic EL. J. Artif. Intell. Res. **44**, 633–708 (2012)
20. Konev, B., Lutz, C., Walther, D., Wolter, F.: Semantic modularity and module extraction in description logics. In: Proceedings of ECAI 2008: The 18th European Conference on Artificial Intelligence, pp. 55–59. IOS Press (2008)
21. Konev, B., Lutz, C., Walther, D., Wolter, F.: Formal properties of modularisation. In: Stuckenschmidt, H., Parent, C., Spaccapietra, S. (eds.) Modular Ontologies. LNCS, vol. 5445, pp. 25–66. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01907-4_3
22. Konev, B., Lutz, C., Walther, D., Wolter, F.: Model-theoretic inseparability and modularity of description logic ontologies. Artif. Intell. **203**, 66–103 (2013)
23. Konev, B., Walther, D., Wolter, F.: The logical difference problem for description logic terminologies. In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR 2008. LNCS (LNAI), vol. 5195, pp. 259–274. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-71070-7_21
24. Kontchakov, R., Wolter, F., Zakharyaschev, M.: Logic-based ontology comparison and module extraction, with an application to DL-Lite. Artif. Intell. **174**(15), 1093–1141 (2010)

25. Lehmann, K., Turhan, A.-Y.: A framework for semantic-based similarity measures for $\mathcal{ELH}$-concepts. In: del Cerro, L.F., Herzig, A., Mengin, J. (eds.) JELIA 2012. LNCS (LNAI), vol. 7519, pp. 307–319. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33353-8_24
26. Ludwig, M., Walther, D.: The logical difference for $\mathcal{ELH}^r$-terminologies using hypergraphs. In: Proceedings of ECAI 2014: The 21st European Conference on Artificial Intelligence, pp. 555–560 (2014)
27. Romero, A.A., Kaminski, M., Grau, B.C., Horrocks, I.: Module extraction in expressive ontology languages via datalog reasoning. J. Artif. Intell. Res. **55**, 499–564 (2016)
28. Sattler, U., Schneider, T., Zakharyaschev, M.: Which kind of module should I extract? In: Proceedings of DL 2009. CEUR Workshop Proceedings, vol. 477. CEUR-WS.org (2009)
29. Zhou, Z., Qi, G., Suntisrivaraporn, B.: A new method of finding all justifications in OWL 2 EL. In: Proceedings of WI 2013: IEEE/WIC/ACM International Conferences on Web Intelligence, pp. 213–220 (2013)