Hitting the Sweetspot: Economic Rewriting of Knowledge Bases

Nadeschda Nikitina[1] and Birte Glimm[2]

[1] Karlsruhe Institute of Technology, Institute AIFB, DE
[2] University of Ulm, Institute of Artificial Intelligence, DE

Abstract. Three conflicting requirements arise in the context of knowledge base (KB) extraction: the size of the extracted KB, the size of the corresponding signature and the syntactic similarity of the extracted KB with the original one. Minimal module extraction and uniform interpolation assign an absolute priority to one of these requirements, thereby limiting the possibilities to influence the other two. We propose a novel technique for $\mathcal{EL}$ that does not require such an extreme prioritization. We propose a tractable rewriting approach and empirically compare the technique with existing approaches with encouraging results.

1 Introduction

In view of the practical deployment of the W3C-specified OWL Web Ontology Language [9] and its specific tractable sublanguages (the so-called *profiles* [5]), non-standard reasoning services supporting different ontology engineering tasks for lightweight logics have gained in importance. Amongst others, the task of semantics-preserving knowledge base extraction for a particular subset of terms has been investigated by the research community: given a knowledge base using a certain vocabulary (called a signature), and a subset of "relevant terms" of that vocabulary, find a knowledge base that contains as little irrelevant information as possible, and, at the same time, contains all information about the relevant terms.

Among the applications of knowledge base extraction is ontology reuse, which helps reducing the expenses of knowledge intensive applications by exploiting the variety of the existing large ontologies. Since the size of a knowledge base has a crucial impact on the maintenance costs and often on the performance of reasoning, it is important to keep the corresponding knowledge base as compact as possible. Knowledge base extraction ideally reduces the amount of irrelevant information imported from external sources, and, at the same time, preserves all relevant consequences. Another application is supporting knowledge engineers in modeling a particular domain or in understanding existing models by revealing dependencies between particular concepts and roles, as, for instance, in the case of interactive ontology revision [8]. Due to its usefulness in various contexts, the task of knowledge base extraction has been investigated by different authors. The currently existing semantics-preserving approaches can be divided into those that compute a subset of the original ontology entailing all relevant consequences (module extraction), e.g., [3, 1], and those rewriting the original ontology to contain only relevant terms while preserving all relevant consequences (uniform interpolation), e.g., [2, 4, 7]. The complexity results for approaches computing a minimal solution are not very promising: even for the lightweight logic $\mathcal{EL}$, the task of minimal module extraction is EXPTIME-hard and the task of uniform interpolation is even 3-EXPTIME-hard with a tight triple-exponential bound on the size of uniform interpolants in case a finite result exists [7]. Given that most applications of knowledge base extraction are of particular interest for large ontologies and that there are scenarios, in which long computation times are not feasible due to user interaction, tractable approaches computing a small but not necessarily minimal solution would often be a reasonable alternative. Moreover, both types of approaches are based on a specific prioritization of objectives that might be necessary in particular scenarios, but is disadvantageous in many others due to its negative impact on the size of the extracted knowledge bases.

In this paper, we consider three conflicting objectives for knowledge base extraction: reducing the size of the extracted knowledge base, reducing the size of its signature and preserving the syntactic similarity of the extracted knowledge base with the originally given one. We demonstrate that, both, minimal module extraction and uniform interpolation, assign an absolute priority to one of these objectives, thereby limiting the possibilities to achieve an improvement w.r.t. the other two. While minimal module extraction only considers subsets of the original knowledge base, thereby requiring a very strong notion of syntactic similarity, uniform interpolation fixes the signature of the extracted knowledge base, possibly yielding triple-exponentially many double-exponentially large axioms. To address scenarios, where the above uncompromising prioritization is not required, we investigate alternative prioritization, allowing for a more balanced relationship between the extents to which the objectives are achieved.

We consider the task of knowledge base extraction for the lightweight logic $\mathcal{EL}$ based on two alternative, less restrictive notions of structural similarity, further assigning the second-highest priority to the knowledge base size. First, we discuss the extraction of knowledge bases consisting only of sub-expressions occurring in the original knowledge base. We give a polynomially-bounded rewriting making particular simple consequences within the knowledge base explicit, such that minimal modules meeting this similarity requirement can be obtained in EXPTIME by applying minimal module extraction to the extended knowledge base.

Second, we consider the extraction of knowledge bases that consist of concepts structurally equivalent to sub-expressions occurring in the original knowledge base, i.e., concepts with the same structure but possibly a different set of atomic concepts. While the extraction of such minimal knowledge bases by first extending the knowledge base and then applying minimal module extraction requires in the worst-case double-exponential time, we propose a tractable rewriting approach that aims at obtaining small but not necessarily minimal knowledge bases. The approach is based on the same elementary rewriting operation as uniform interpolation in [7], namely replacing atomic concepts within expressions by their subsumees and subsumers. However, in order to obtain polynomial bounds and preserve the required structural similarity, we impose additional restrictions on the rewriting, excluding elementary rewriting operations with a negative effect on the module size or structure.

As we show in our evaluation using the Gene Ontology, knowledge bases obtained by our approach on average contain half as many axioms as their minimal justifications within the original knowledge base. A comparison with the existing implementations also yields promising results. In case of the minimal module extractor for DL-Lite$_{\texttt{bool}}$, the extracted modules are 2 to 2.2 times larger than the knowledge bases obtained by our approach. The locality-based module extractor, which is a tractable approach for extracting small but not necessarily minimal subsets of an ontology, extracts modules that are on average 12 times larger than the knowledge bases obtained by our approach.

The paper is organized as follows: In Section 2, we recall the necessary preliminaries on $\mathcal{EL}$. Section 3 formally introduces the task of knowledge base extraction and discusses the conflicting objectives for this task. In Section 4, we show how minimal modules meeting the corresponding requirements of syntactic similarity can be obtained using minimal module extraction. In Sections 5 and 6, we propose a tractable alternative for minimal module extraction based on rewriting. After introducing rewriting in Section 5, in Section 6 we discuss the necessary restrictions on rewriting operations in order to obtain polynomial bounds and preserve the required structural similarity. Finally, we present the evaluation results in Section 7 before we conclude in Section 8. Further details and proofs can be found in the extended version of this paper [6].

2 Preliminaries

Let N_C and N_R be countably infinite and mutually disjoint sets of concept symbols and role symbols. An $\mathcal{EL}$ concept C is defined as $C ::= A|\top|C \sqcap C|\exists r.C$, where A and r range over N_C and N_R, respectively. In the following, we use symbols A, B to denote atomic concepts and C, D to denote arbitrary concepts. A *terminology* or *TBox* consists of *concept inclusion* axioms $C \sqsubseteq D$ and *concept equivalence* axioms $C \equiv D$ used as a shorthand for $C \sqsubseteq D$ and $D \sqsubseteq C$. While knowledge bases in general can also include a specification of individuals with the corresponding concept and role assertions (ABox), in this paper we abstract from ABoxes and concentrate on TBoxes. The signature of an $\mathcal{EL}$ concept C or an axiom α, denoted by $\mathrm{sig}(C)$ or $\mathrm{sig}(\alpha)$, respectively, is the set of concept and role symbols occurring in it. To distinguish between the set of concept symbols and the set of role symbols, we use $\mathrm{sig}_C(C)$ and $\mathrm{sig}_R(C)$, respectively. The signature of a TBox $\mathcal{T}$, in symbols $\mathrm{sig}(\mathcal{T})$ (correspondingly, $\mathrm{sig}_C(\mathcal{T})$ and $\mathrm{sig}_R(\mathcal{T})$), is defined analogously. Next, we recall the semantics of the above introduced DL constructs, which is defined by the means of interpretations. An interpretation $\mathcal{I}$ is given by the domain $\Delta^{\mathcal{I}}$ and a function $\cdot^{\mathcal{I}}$ assigning each concept $A \in N_C$ a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and each role $r \in N_R$ a subset $r^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation of $\top$ is fixed to $\Delta^{\mathcal{I}}$. The interpretation of an arbitrary $\mathcal{EL}$ concept is defined inductively, i.e., $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$ and $(\exists r.C)^{\mathcal{I}} = \{x \mid (x, y) \in r^{\mathcal{I}}, y \in C^{\mathcal{I}}\}$. An interpretation $\mathcal{I}$ satisfies an axiom $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. $\mathcal{I}$ is a model of a TBox, if it satisfies all of its axioms. We say that a TBox $\mathcal{T}$ entails an axiom α (in symbols, $\mathcal{T} \models \alpha$), if α is satisfied by all models of $\mathcal{T}$.

3 Knowledge Base Extraction Revisited

While, in principle, there exist many approaches without a logical background, in this work we focus on logic-based approaches, i.e., approaches that guarantee a preservation of the semantics for the set of relevant entities. We say that the semantics is preserved, if all logical consequences concerning only the relevant entities are preserved. The logical foundation for such a preservation of relevant consequences is given by the established notion of *inseparability*. Two knowledge bases, $\mathcal{T}_1$ and $\mathcal{T}_2$, are inseparable w.r.t. a signature Σ if they have the same Σ-consequences, i.e., consequences whose signature is a subset of Σ. Depending on the particular application requirements, the expressivity of those Σ-consequences can vary from subsumption queries and instance queries to conjunctive queries. In the following, we consider concept inseparability of general $\mathcal{EL}$ terminologies defined analogously to previous work [3, 2, 4, 7], as follows:

Definition 1. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two general $\mathcal{EL}$ knowledge bases and Σ a signature. $\mathcal{T}_1$ and $\mathcal{T}_2$ are concept-inseparable w.r.t. Σ, in symbols $\mathcal{T}_1 \equiv^{c}_{\Sigma} \mathcal{T}_2$, if for all $\mathcal{EL}$ concepts C, D with $sig(C) \cup sig(D) \subseteq \Sigma$ holds $\mathcal{T}_1 \models C \sqsubseteq D$, iff $\mathcal{T}_2 \models C \sqsubseteq D$.*

Given a signature Σ and a knowledge base $\mathcal{T}$, the task of knowledge base extraction in general is to compute a knowledge base $\mathcal{T}'$, which is entailed by $\mathcal{T}$ and is concept-inseparable from it. We call the result $\mathcal{T}'$ a *general module* of $\mathcal{T}$.

Definition 2. *Let $\mathcal{T}$ be an $\mathcal{EL}$ knowledge base and Σ a signature. An $\mathcal{EL}$ knowledge base $\mathcal{T}'$ is a* general module *of $\mathcal{T}$ w.r.t. Σ, written $\mathcal{T}' \in \texttt{MOD}(\mathcal{T}, \Sigma)$, iff (1) $\mathcal{T} \equiv^{c}_{\Sigma} \mathcal{T}'$ and (2) $\mathcal{T} \models \mathcal{T}'$.*

The above definition is very generic. It captures the preservation of the semantics, but does not address the quality criteria important for general modules in order to be useful in practice. We consider the following requirements for the task of knowledge base extraction:

1. **Syntactic Similarity**: In scenarios, where the knowledge base is meant to be used by human experts, the syntactic structure of the module determining its comprehensiveness or cognitive complexity has to be taken into account. The extent, to which a general module has to be syntactically similar to the original knowledge base $\mathcal{T}$ depends on the particular application requirements. For instance, modules can be required to be a subset of $\mathcal{T}$, to consist only of sub-expressions occurring in $\mathcal{T}$ or to consist only of concepts structurally equivalent to sub-expressions occurring in $\mathcal{T}$, but possibly referencing different atomic concepts.
2. **Small Knowledge Base Size**: Reducing the size of the knowledge base is a core objective for the task of knowledge base extraction, since smaller knowledge bases (assuming that the particular syntactic similarity requirement is fulfilled in both cases) require less computational and manual effort in many different ontology management activities.
3. **Small Signature Size**: Decreasing the size of the signature results in a decrease of irrelevant entities occurring in the knowledge base, which is also one of the core objectives of knowledge base extraction.

While uniform interpolation clearly prioritizes small signature size making no compromises w.r.t. the other two requirements, minimal module extraction gives the highest priority to syntactic similarity, thereby not allowing for rewriting and, therefore, limiting the possibilities to reduce the size. While such uncompromising prioritization can be required in some particular scenarios, in other scenarios it leads to a disadvantage. The following example demonstrates the drawbacks of minimal module extraction and uniform interpolation in terms of knowledge base size caused by the extreme choice of priorities.

Example 1. Consider the following knowledge base $\mathcal{T}$ [3]:

$$A_{13} \sqsubseteq A_9 \quad A_{10} \sqcap A_{13} \sqsubseteq A_{16} \quad A_9 \sqsubseteq \exists r.A_9 \quad A_{i+1} \sqsubseteq A_i \qquad 1 \leq i \leq 14, i \neq 13$$

For the signature $\Sigma = \{A_1, A_8, A_{12}, A_{15}, A_{16}, r\}$, neither the uniform interpolation nor minimal module extraction are effective in terms of reducing the size. While minimal module extraction would return the whole knowledge base, uniform interpolation fails to extract a finite knowledge base due to the cyclic dependency given by $A_9 \sqsubseteq \exists r.A_9$. However, if we are not restricted to subsets of $\mathcal{T}$, but are also interested in modules consisting of sub-expressions occurring in $\mathcal{T}$, then there is a representation of the relevant information about Σ, which uses half as many axioms as the original TBox: $\{A_{12} \sqsubseteq A_{10}$, $A_{15} \sqsubseteq A_{13}$, $A_{10} \sqcap A_{13} \sqsubseteq A_{16}$, $A_{10} \sqsubseteq A_9$, $A_{13} \sqsubseteq A_9$, $A_9 \sqsubseteq \exists r.A_9$, $A_9 \sqsubseteq A_8$, $A_8 \sqsubseteq A_1\}$. If we are, additionally, allowed to exchange atomic concepts within sub-expressions while leaving the structure of expressions unchanged, then there is an even smaller representation consisting of 6 axioms: $\{A_{12} \sqcap A_{15} \sqsubseteq A_{16}$, $A_{12} \sqsubseteq A_9$, $A_{15} \sqsubseteq A_9$, $A_8 \sqsubseteq A_1$, $A_9 \sqsubseteq A_8$, $A_9 \sqsubseteq \exists r.A_9\}$.

In the following, we aim at establishing a balance between these requirements in order to account for scenarios not requiring the above mentioned extreme prioritization. The following example completes the picture roughly sketched above and demonstrates mutual influences of the three requirements upon each other.

Example 2. The following TBox $\mathcal{T}$ models a "counter" with numbers $X_0, \ldots, X_{10}$, where the lowest number X_0 has two subsumees:

$$A_1 \sqsubseteq X_0 \quad A_2 \sqsubseteq X_0 \quad \exists r.X_i \sqcap \exists s.X_i \sqsubseteq X_{i+1} \qquad 0 \leq i \leq 9$$

Given this TBox, we could extract a knowledge base not referencing a particular atomic concept by replacing its occurrence by its direct subsumees. For instance, if we want to represent the information without using X_1, we can omit $\exists r.X_0 \sqcap \exists s.X_0 \sqsubseteq X_1$ and replace X_1 on the left-hand side of the remaining axioms by its direct subsumee $\exists r.X_0 \sqcap \exists s.X_0$, leading to $\exists r.(\exists r.X_0 \sqcap \exists s.X_0) \sqcap \exists s.(\exists r.X_0 \sqcap \exists s.X_0) \sqsubseteq X_2$. Concerning the extraction of knowledge bases from $\mathcal{T}$, we can more generally observe the following:

- Assume that we are interested in the dependencies between X_0, and X_{10} including those using roles r, s. By replacing any of the concepts $X_1, ..., X_9$ by their direct subsumees, we reduce both, the number of axioms and the number of referenced concept names, but we increase the nesting depth of the resulting TBox. A complete replacement of $X_1, ..., X_9$ would yield a subsumee of X_{10} with a nesting depth of 10 and exponentially many occurrences of X_0. Even though the TBox contains only three axioms and no irrelevant concept names, it is less comprehensive than the original knowledge base.

[3] The TBox is structurally similar to minimal modules obtained within our evaluation and can be extended to a more typical TBox by adding more axioms to the subsumption hierarchy without any effect on the obtained general modules.

- Assume that we are interested in A_1, A_2 instead of X_0. Eliminating X_0 from $\mathcal{T}$ would yield four different subsumees of X_1, namely $\exists r.A_1 \sqcap \exists s.A_1$, $\exists r.A_1 \sqcap \exists s.A_2$, $\exists r.A_2 \sqcap \exists s.A_1$ and $\exists r.A_2 \sqcap \exists s.A_2$. Each of these subsumees is required in order to preserve the relevant consequences, since none of the four concepts subsumes another. Replacing X_0 in the extracted knowledge base using only A_1, A_2, X_0, X_{10} and r, s by its two subsumees, A_1 and A_2, would result in double exponentially many ($2^{2^{10}}$) different subsumees of X_{10}. Therefore, the elimination of a single concept name is, in most cases, not justified from the practical point of view.

While Example 1 focuses on the disadvantage in terms of knowledge base size caused by an unnecessarily strong notion of syntactic similarity, the latter example demonstrates more clearly the effect of unrestricted rewriting aiming at signature reduction on the knowledge base size. In the following, we consider two particular, more balanced requirement prioritizations. Analogously to minimal module extraction, we aim at preserving syntactic similarity of the extracted knowledge base, however based on the following, less restrictive similarity notions:

1. **Identical Sub-Expressions**:[4] Modules fulfill this notion of similarity, if they consist only of sub-expressions occurring in the original knowledge base.
2. **Structurally Equivalent Sub-Expressions**: Modules fulfill this notion of similarity, if for all of their sub-expressions there is a structurally equivalent sub-expression occurring in the original knowledge base, i.e. an expression with the same syntactic structure, but possibly different atomic concepts. For instance, $A \sqcap \exists r.A$ is structurally equivalent to $B_1 \sqcap \exists r.B_2$.

In the next sections, we investigate the task of knowledge base extraction based on these two notions of syntactic similarity with the second-highest priority given to the knowledge base size. We first show how we can extend the original knowledge base to contain all minimal general modules such that minimal module extraction can identify one of them. Subsequently, we add the computational complexity as a forth dimension. We investigate how we can obtain a tractable alternative to minimal module extraction by sacrificing the minimality guarantee, while fulfilling the requirement of syntactic similarity and reaching a decent effectiveness in terms of module size. In our evaluation, we show that, on average, the approach outperforms minimal module extraction applied directly to the original knowledge base.

4 Computing Modules using Minimal Module Extraction

In this section, we show how, by normalizing the original knowledge base $\mathcal{T}$ and extending it with a subset of its deductive closure, we can obtain a union of all general modules fulfilling the syntactic similarity notions of identical sub-terms and structurally equivalent sub-terms. Applying minimal module extraction to this extended knowledge base would yield all minimal general modules, which can subsequently be ordered according to the signature size.

[4] Conjunctions containing a subset of conjuncts are not considered as sub-expressions

In order to obtain a union of all minimal general modules under the restriction to identical sub-terms, we need to identify all subsumptions between sub-terms occurring in $\mathcal{T}$. We can structurally transform the knowledge base as follows: we assign a temporary concept name to each non-atomic sub-term occurring in $\mathcal{T}$, such that the knowledge base can be represented without nested expressions, i.e., using only axioms of the form $A \sqsubseteq B$, $A \equiv B_1 \sqcap \ldots \sqcap B_n$, and $A \equiv \exists r.B$, where A and $B_{(i)}$ are atomic concepts or $\top$. This can be realized in time linear in the size of $\mathcal{T}$ by recursively replacing complex concepts $C_{(i)}$ in expressions $C_1 \sqcap ... \sqcap C_n$ and $\exists r.C$ by fresh concept symbols with the corresponding equivalence axioms. Note that the original form of the knowledge base can easily be obtained by replacing the temporary concept names by their definitions.

By classifying the obtained knowledge base and extending it with the classification results, all subsumptions between sub-terms occurring in $\mathcal{T}$ are explicitly present in the resulting knowledge base $\mathcal{T}'$, which we call normalized $\mathcal{T}$. Thus, we obtain a polynomially-bounded, complete union of all possible general modules consisting of sub-terms occurring in $\mathcal{T}$ by replacing the temporary concept names by their definitions. Applying minimal module extraction to this knowledge base yields minimal general modules fulfilling the corresponding syntactic similarity requirement. In this way, we obtain a linear bound on the size of the general modules and, as demonstrated by Example 1, in most cases outperform minimal module extraction applied to the original knowledge base w.r.t. both, size of the signature and size of the general module.

If, in addition to identical sub-terms from $\mathcal{T}$, we can also use structurally equivalent sub-terms, we can introduce temporary concept names for all structurally equivalent sub-terms and then apply classification to obtain the corresponding dependencies. However, extending the knowledge base with all dependencies of the corresponding form and then applying minimal module extraction would lead to an increase of the overall complexity from exponential to double-exponential.

5 Computing Modules using Rewriting

In the following, we propose a tractable approach to extracting general modules consisting of concepts structurally equivalent to sub-expressions of the original knowledge base. The approach is based on rewriting as it is used for uniform interpolation [7] and the aim of this section is to show how general modules are obtained using this rewriting.

In order to simplify the tracking of subsumption dependencies during the rewriting, we use the normalization introduced in the last section. Given a normalized $\mathcal{EL}$ knowledge base, the elimination of roles can be done by omitting all axioms with subsumees and subsumers containing irrelevant roles without loosing any relevant consequences. In the following, we focus, therefore, on the elimination of irrelevant concept names and assume w.l.o.g. that the sets of subsumees and subsumers do not contain any roles not from Σ.

During the rewriting, we keep two relations that map each atomic concept in a TBox to a set of concepts. These relations initially contain, for each atomic concept, the subsumees and subsumers as given by the normalized TBox. Each rewriting step then refines these relations in such a way that the union of all corresponding subsumption axioms is still a general module.

Definition 3. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ knowledge base and $R^{\mathcal{T}}_{\sqsupseteq}, R^{\mathcal{T}}_{\sqsubseteq}$ relations that map each atomic concept $B \in sig_C(\mathcal{T})$ to a set of subsumees and a set of subsumers of B entailed by $\mathcal{T}$. Any pair $\langle R^{\mathcal{T}}_{\sqsupseteq}, R^{\mathcal{T}}_{\sqsubseteq}\rangle$ is called a* subsumee/subsumer relation pair for *$\mathcal{T}$ and it is called the* initial subsumee/subsumer relation pair for $\mathcal{T}$ *if $R^{\mathcal{T}}_{\sqsupseteq}$ and $R^{\mathcal{T}}_{\sqsubseteq}$ are as follows:*

1. $R^{\mathcal{T}}_{\sqsupseteq}(B) = \{C \mid C \sqsubseteq B \in \mathcal{T}$ *or* $C \equiv B \in \mathcal{T}\}$,
2. $R^{\mathcal{T}}_{\sqsubseteq}(B) = \{C \mid B \sqsubseteq C \in \mathcal{T}$ *or* $B \equiv C \in \mathcal{T}\}$.

If $\mathcal{T}$ is clear from the context, we simply write $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$. Starting with the initial subsumee/subsumer relation, our rewriting aims at obtaining another pair of relations that allows for constructing a uniform interpolant as follows:

Definition 4. *Let $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$ be a subsumee/subsumer relation pair and Σ a signature. We denote by $\Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$ the extension of Σ with atomic concepts occurring in the range of $R_{\sqsupseteq}$ and $R_{\sqsubseteq}$. We construct a knowledge base $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma)$ from $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$ and Σ as:*

$$\begin{aligned}\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma) = &\{C \sqsubseteq A \mid A \in \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq}), C \in R_{\sqsupseteq}(A)\} \cup \\ &\{A \sqsubseteq D \mid A \in \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq}), D \in R_{\sqsubseteq}(A)\} \cup \\ &\{C \sqsubseteq D \mid \textit{there is } A \notin \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq}), C \in R_{\sqsupseteq}(A), D \in R_{\sqsubseteq}(A)\},\end{aligned}$$

If $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma) \in \mathtt{MOD}(\mathcal{T}, \Sigma)$, we say that $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$ is complete *w.r.t. Σ.*

The above definition avoids an unnecessary extension of the knowledge base signature with atomic concepts in case $A \notin \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})$. Note that even in the initial subsumee/subsumer relation pair this case can occur, namely when concepts not from Σ do not have atomic subsumers or subsumees. We can show that the initial subsumee/subsumer relation pair meets the completeness criterion:

Theorem 1. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ knowledge base, $\Sigma \subseteq sig(\mathcal{T})$ a signature, and $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$ the initial subsumee/subsumer relation pair for $\mathcal{T}$, then $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$ is complete w.r.t. Σ.*

Before defining the rewriting step that refines the initial subsumee/subsumer relation pair into another subsumee/subsumer relation pair preserving completeness, we show the initial subsumee/subsumer relation pair $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$ and the according general module $\mathcal{T}_{\mathtt{M}} = \mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq}))$ for the knowledge base $\mathcal{T}$ from Example 1. We first normalize $\mathcal{T}$ by introducing two temporary concepts, $B_1 \equiv A_{10} \sqcap A_{13}$ and $B_2 \equiv \exists r.A_9$, then we classify the normalized knowledge base and obtain the initial subsumee/subsumer relation pair shown in Fig. 1.

The knowledge base $\mathcal{T}_{\mathtt{M}}$ contains, for each A_i with $i \in \{1, \ldots, 16\}$, all axioms of the form $C \sqsubseteq A_i$ with $C \in R_{\sqsupseteq}(A_i)$ and $A_i \sqsubseteq C$ with $C \in R_{\sqsubseteq}(A_i)$. This also holds for B_1 and B_2. It is not difficult to check that, after replacing B_1 by $A_{10} \sqcap A_{13}$ and B_2 by $\exists r.A_9$ in $\mathcal{T}_{\mathtt{M}}$, each axiom of $\{A_{12} \sqsubseteq A_{10}$, $A_{15} \sqsubseteq A_{13}$, $A_{10} \sqcap A_{13} \sqsubseteq A_{16}$, $A_{10} \sqsubseteq A_9$, $A_{13} \sqsubseteq A_9$, $A_9 \sqsubseteq \exists r.A_9$, $A_9 \sqsubseteq A_8$, $A_8 \sqsubseteq A_1\}$ is contained in it. Thus, the result contains, among other axioms, the general module given in Section 3 consisting of sub-expressions of $\mathcal{T}$, which shows completeness of $\langle R_{\sqsupseteq}, R_{\sqsubseteq}\rangle$.

	$R_{\sqsupseteq}$	$R_{\sqsubseteq}$
A_{16}	B_1	$\emptyset$
A_{15}	$\emptyset$	$A_1, \ldots, A_9, B_2, A_{13}, A_{14}$
A_{14}	A_{15}	$A_1, \ldots, A_9, B_2, A_{13}$
A_{13}	B_1, A_{14}, A_{15}	$A_1, \ldots, A_9, B_2$
A_{12}	$\emptyset$	$A_1, \ldots, A_{11}, B_2,$
A_{11}	A_{12}	$A_1, \ldots, A_{10}, B_2$
A_{10}	$B_1, A_{11}, A_{12}, A_{15}$	$A_1, \ldots, A_9, B_2$
A_9	$B_1, A_{10}, \ldots, A_{15}$	$A_1, \ldots, A_8, B_2$
A_8	$B_1, A_9, \ldots, A_{15}$	$A_1, \ldots, A_7$
A_7	$B_1, A_8, \ldots, A_{15}$	$A_1, \ldots, A_6$
A_6	$B_1, A_7, \ldots, A_{15}$	$A_1, \ldots, A_5$
A_5	$B_1, A_6, \ldots, A_{15}$	$A_1, \ldots, A_4$
A_4	$B_1, A_5, \ldots, A_{15}$	$A_1, \ldots, A_3$
A_3	$B_1, A_4, \ldots, A_{15}$	A_1, A_2
A_2	$B_1, A_3, \ldots, A_{15}$	A_1
A_1	$B_1, A_2, \ldots, A_{15}$	$\emptyset$
B_2	$\exists r.A_9, A_9, \ldots, A_{15}, B_1$	$\exists r.A_9$
B_1	$A_{10} \sqcap A_{13}$	$A_{10} \sqcap A_{13}, A_1, \ldots, A_{10}, A_{13}, B_2, A_{16}$

Fig. 1. The initial subsumee/subsumer relation pair $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$ for Example 1

Since rewritings aiming at eliminating all irrelevant concept names yield smaller modules for sparse relation pairs, we will only use a subset of the subsumee/subsumer relations used as input for minimal module extraction. We compute a reduced subsumee/subsumer relation pair that only uses the transitive reduction of the classification results, i.e., we consider $B_1 \sqsubseteq B_2$ only if there is no B_3 such that $B_1 \sqsubseteq B_3$ and $B_3 \sqsubseteq B_2$. Furthermore, we compute, in polynomial time, a reduced graph by recursively eliminating subsumers and subsumees not from Σ that do not have any outgoing edges. It is easy to check that the completeness of the initial subsumee/subsumer relation pair stated in Theorem 1 still holds. In the next section, we assume this reduced form of initial subsumee/subsumer relation pair $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$.

As demonstrated in Example 2, within the task of uniform interpolation, a single rewriting step replaces occurrences of an atomic concept in all subsumees and subsumers within a relation pair by its subsumees and subsumers, respectively. Since, in general, an atomic concept can have infinitely many subsumees and subsumers, using the whole set of subsumees and subsumers for rewriting is not feasible in practice. Interestingly, if the initial relation pair is complete, then a small subset of all subsumees and subsumers of the replaced concept is sufficient to preserve the completeness of the relation pair (in general, however, the sets of direct subsumees and subsumers are not sufficient). Among other things, the relevant subset does not need to include subsumees that can be obtained from other subsumees by adding arbitrary conjuncts to arbitrary sub-expressions. For instance, if B is a subsumee of A, then we do not need $B \sqcap B'$ for the replacement of A. Similarly, the minimal subset of subsumers required for replacement does not include concepts that can be obtained from other subsumers by omitting arbitrary conjuncts from arbitrary sub-expressions. While, in case of subsumees, a conjunction is not required if at least one of the conjuncts is a subsumee, in case of subsumers, we need to introduce a conjunction in particular when replacing an atomic concept within the scope of an existential restriction. Using the standard substitution notation $C[A/B]$ for denoting the concept obtained by replacing all occurrences of B within C by A, we give the following definition of an elementary rewriting.

Definition 5. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ knowledge base, $\Sigma \subseteq sig(\mathcal{T})$ a signature, and $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$ a subsumee/subsumer relation pair for $\mathcal{T}$. For atomic concepts $A, B \in sig_C(\mathcal{T})$ and $\bowtie \in \{\sqsupseteq, \sqsubseteq\}$, an* elementary rewriting $\mathtt{Rew}_{R_{\bowtie}}(B, C, A)$ *of a subsumee/subsumer $C \in R_{\bowtie}(B)$ w.r.t. A is given by*

1. $\mathtt{Rew}_{R_{\sqsupseteq}}(B, C, A) = \{(B, C') \mid A' \in R_{\sqsupseteq}(A), C' = C[A'/A]\}$.
2. $\mathtt{Rew}_{R_{\sqsubseteq}}(B, C, A) = \begin{cases} \{(B, C') \mid D' = \bigsqcap_{D \in R_{\sqsubseteq}(A)} D, C' = C[D'/A]\}, & (a) \\ \{(B, C') \mid A' \in R_{\sqsubseteq}(A), C' = C[A'/A]\}, & (b) \end{cases}$

where (a) is used when A is within the scope of an existential restriction and (b) is used otherwise. Let $S_A = \{(B, C) \mid C \in R_{\bowtie}(B)$ and A occurs in $C\}$. A rewriting w.r.t. A is given by $\mathtt{Rew}_{R_{\bowtie}}(A) = \bigcup_{(B,C) \in S_A} \mathtt{Rew}_{R_{\bowtie}}(B, C, A) \cup R_{\bowtie} \setminus S_A$.

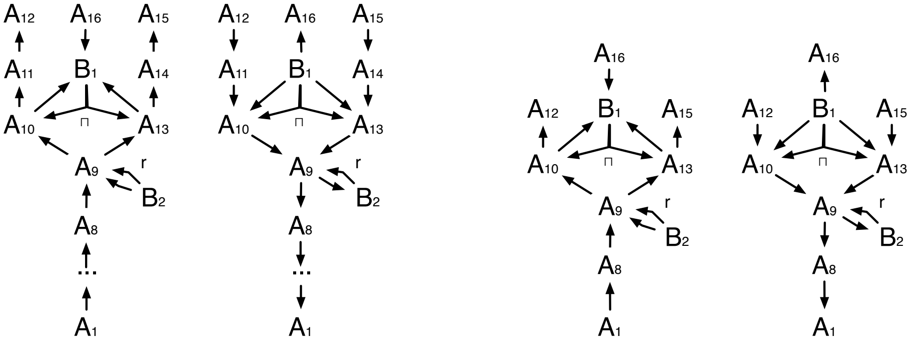
In order to keep the relations as small as possible, we further remove trivial subsumees and subsumers obtained during the rewriting, namely atomic concepts themselves and, in case of subsumee relations, conjunctions with the atomic concept itself as one of the conjuncts. This check is inexpensive from the computational point of view, since such trivial subsumees and subsumers can be identified independently from other subsumees and subsumers. In what follows, we assume that such trivial subsumees and subsumers are removed after each rewriting. We obtain the following result concerning the completeness w.r.t. Σ:

Theorem 2. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$ knowledge base, $\Sigma \subseteq sig(\mathcal{T})$ a signature, $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$ a subsumee/subsumer relation pair for $\mathcal{T}$ that is complete w.r.t. Σ. Then, for any $B' \notin \Sigma$ holds $\langle \mathtt{Rew}_{R_{\sqsupseteq}}(B'), R_{\sqsubseteq} \rangle$ and $\langle R_{\sqsupseteq}, \mathtt{Rew}_{R_{\sqsubseteq}}(B') \rangle$ are subsumee/subsumer relation pairs for $\mathcal{T}$, which are complete w.r.t. Σ.*

Thus, starting with the initial subsumee/subsumer relation pair $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$, after each rewriting step we obtain a subsumee/subsumer relation pair over $\mathcal{T}$ that is complete w.r.t. Σ. However, without further restrictions, the above rewritings would potentially introduce many large nested concept expressions or might not even terminate. In the next section, we show how these problems can be avoided by stating the corresponding validity criteria for rewritings on subsumee/subsumer relation pairs.

6 Restricting Rewriting

In this section, we address the problems caused by unrestricted application of rewriting pointed out in Example 2. On the one hand, the example shows that rewriting can significantly change the syntactic structure of a knowledge base. On the other hand, it demonstrates that, while in some cases an elimination of a particular concept name can lead to a smaller knowledge base, it can cause the knowledge base to grow by several factors or even get infinite in other cases.



(a) Initial $R_{\sqsupseteq}$ (left) and $R_{\sqsubseteq}$ (right)

(b) Rewriting w.r.t. $A_2 - A_7, A_{11}, A_{14}$

Fig. 2. Hypergraphs for the knowledge base in Example 1

In order to avoid the above negative effects of rewriting, after each rewriting step we identify and exclude *invalid* rewritings, i.e., rewritings having a negative impact on the structure of the resulting module or the size of the relation pair. In particular, we exclude rewritings replacing atomic concepts by the conjunction of their direct subsumers corresponding to case (a) in Definition 5, since such a replacement possibly introduces concept expressions with a new structure not occurring in the original knowledge base. Thus, the set of valid rewritings is restricted to replacements of atomic concepts by their direct subsumees and subsumers. For the same reason, we additionally exclude rewritings yielding nested concept expressions, i.e., replacements of an atomic concept within a conjunction or existential restriction by one of its non-atomic subsumees or subsumers. Since the initial subsumee/subsumer relation pair contains only concepts of the form $B, \exists r.B$ and $B_1 \sqcap ... \sqcap B_n$, after each valid rewriting step, all subsumees and subsumers have also this simple form. In this way, subsumee/subsumer relations can be represented as hypergraphs with atomic concepts as nodes and three types of edges, namely $A \rightarrow B$ representing atomic subsumees/subsumers, $A \xrightarrow{r} B$ representing existential restrictions, and multi-edges $A \xrightarrow{\sqcap} B_1, ..., B_n$ representing conjunctions. The corresponding hypergraphs for the initial subsumee/subsumer relation pair $\langle R_{\sqsupseteq}, R_{\sqsubseteq} \rangle$ for the knowledge base in Example 1 are shown in Fig. 2(a).

In the following, we give a set of excluding conditions for rewritings according to the requirement of syntactic similarity and an inequation excluding rewritings negatively affecting knowledge base size. Within the excluding conditions, we distinguish three types of successors and predecessors according to the types of edges. For an atomic concept A and a relation $R_{\bowtie}$ with $\bowtie \in \{\sqsupseteq, \sqsubseteq\}$, we use

$$\begin{aligned}
\mathtt{IN}_A(A) &:= \{B \in \mathrm{sig}_C(\mathcal{T}) \mid A \in R_{\bowtie}(B)\} \\
\mathtt{OUT}_A(A) &:= R_{\bowtie}(A) \cap \mathrm{sig}_C(\mathcal{T}) \\
\mathtt{IN}_{\mathtt{Roles}}(A) &:= \{B \mid \exists r.A \in R_{\bowtie}(B)\} \\
\mathtt{OUT}_{\mathtt{Roles}}(A) &:= \{B \mid \exists r.B \in R_{\bowtie}(A)\} \\
\mathtt{IN}_{\mathtt{Con}}(A) &:= \{B \mid B'_1 \sqcap ... \sqcap B'_n \in R_{\bowtie}(B) \text{ with } A = B'_i \text{ for some } i \in \{1, ..., n\}\} \\
\mathtt{OUT}_{\mathtt{Con}}(A) &:= \{B'_1 \sqcap ... \sqcap B'_n \mid B'_1 \sqcap \ldots \sqcap B'_n \in R_{\bowtie}(A)\}
\end{aligned}$$

Further, let $\mathtt{IN}(A) = \mathtt{IN}_A(A) \cup \mathtt{IN}_{\mathtt{Roles}}(A) \cup \mathtt{IN}_{\mathtt{Con}}(A)$ and $\mathtt{OUT}(A) = \mathtt{OUT}_A(A) \cup \mathtt{OUT}_{\mathtt{Roles}}(A) \cup \mathtt{OUT}_{\mathtt{Con}}(A)$.

In order to avoid an introduction of structurally new concept expressions during the rewriting and ensure termination, we exclude a rewriting w.r.t. an atomic concept A if one of the following conditions is true:

$$(\mathtt{IN}_{\mathtt{Roles}}(A) \cup \mathtt{IN}_{\mathtt{Con}}(A) \neq \emptyset) \text{ and } \mathtt{OUT}_A(A) \text{ contains temporary concepts;} \tag{1}$$

$$(\mathtt{IN}_{\mathtt{Roles}}(A) \cup \mathtt{IN}_{\mathtt{Con}}(A) \neq \emptyset) \text{ and } (\mathtt{OUT}_{\mathtt{Roles}}(A) \cup \mathtt{OUT}_{\mathtt{Con}}(A) \neq \emptyset); \tag{2}$$

$$R_{\bowtie} \text{ is a subsumer relation and } |\mathtt{IN}_{\mathtt{Roles}}(A)| \geq 1 \text{ and } |\mathtt{OUT}(A)| \geq 2; \tag{3}$$

$$\text{Some } C \in R_{\bowtie}(A) \text{ contains } A; \tag{4}$$

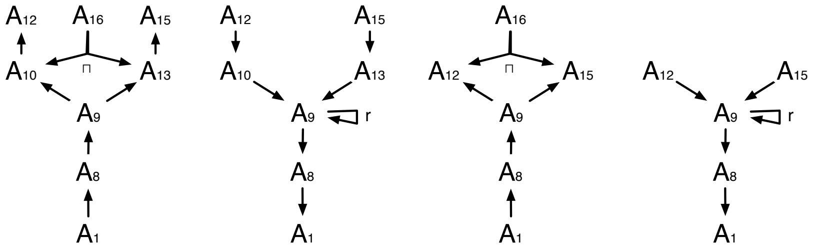
For instance, in Example 1 the rewriting w.r.t. A_9 in $R_{\sqsubseteq}$ is invalid due to Condition (3) and rewriting w.r.t. A_{10}, A_{13} in $R_{\sqsupseteq}$ are invalid due to Condition (2).

In order to identify rewritings that would increase the size of a relation, we compare the number of edges before and after the rewriting. While the number of edges potentially affected by a rewriting w.r.t. a concept A can be given by $|\mathtt{IN}(A)| + |\mathtt{OUT}(A)|$, the corresponding number of affected edges after the rewriting is in general bounded by $|\mathtt{OUT}(A)| + |\mathtt{IN}(A)| \cdot |\mathtt{OUT}(A)|$. Interestingly, if a concept B is unreferenced, it is usually possible to remove some elements from the corresponding sets $R_{\sqsupseteq}(B)$ and $R_{\sqsubseteq}(B)$ without losing any Σ consequences, or even without losing any axioms in $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq}))$. We can remove subsumees and subsumers of unreferenced concepts, if none of the corresponding axioms in $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq}))$ that contain these subsumees and subsumers, add any new Σ consequences to $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq}))$. Thus, in order to determine if a subsumee $C \in R_{\sqsupseteq}(B)$ of $B \notin \Sigma$ is unnecessary, we check for each element $D \in R_{\sqsubseteq}(B)$, if $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})) \setminus \{C \sqsubseteq D\} \models C \sqsubseteq D$. Unnecessary subsumers can be determined in the same manner. For instance, in case of A_2 in Example 1, after the corresponding rewriting of both relations we can remove its subsumee A_3 and subsumer A_1, if $\mathtt{M}(R_{\sqsupseteq}, R_{\sqsubseteq}, \Sigma^{\mathtt{ext}}(R_{\sqsupseteq}, R_{\sqsubseteq})) \setminus \{A_1 \sqsubseteq A_3\} \models A_1 \sqsubseteq A_3$. It is easy to check given the corresponding hypergraphs that this is indeed the case. In fact, the corresponding sets of necessary subsumees and subsumers after the rewriting are empty for $A_2, \ldots, A_7$, $A_{10}, A_{11}, A_{13}, A_{14}$ and B_1, B_2.

Given the relation $R^{\mathtt{red}}_{\bowtie}$ obtained by omitting such unnecessary elements from the set $R_{\bowtie}(B)$, we can use a tighter bound on the number of edges after rewriting based on $n_{R_{\bowtie}} = |R^{\mathtt{red}}_{\bowtie}(B)|$ instead of $|R_{\bowtie}(B)|$. Thus, we obtain the following inequation that holds for rewritings potentially increasing the size of relations:

$$|\mathtt{IN}(A)| + |\mathtt{OUT}(A)| < n_{R_{\bowtie}} + |\mathtt{IN}(A)| \cdot |\mathtt{OUT}(A)| \tag{5}$$

In Example 1, $|\mathtt{IN}_A(A_i)| = 1$ and $|\mathtt{OUT}_A(A_i)| = 1$ holds for all $i \in \{2, ..., 7, 11, 14\}$. Since both, $n_{R_{\sqsupseteq}}$ and $n_{R_{\sqsubseteq}}$ are 0 for all A_i, the number of edges decreases by one in case of each rewriting. After each rewriting including the subsequent omitting of unnecessary successors of the replaced concept, the number of edges as well as $n_{R_{\sqsupseteq}}$ and $n_{R_{\sqsubseteq}}$ remain the same for all remaining concepts. Thus, the conditions for the remaining concepts A_i with $i \in \{2, ..., 7, 11, 14\}$ do not change during any of the above rewritings. After performing all of the above rewritings, we obtain the subsumee/subsumer relation pair shown in Fig. 2(b).



(a) Rewriting w.r.t. B_1, B_2

(b) Rewriting w.r.t. A_{10}, A_{13}

Fig. 3. Rewriting for the knowledge base in Example 1

In case of B_1, in $R_\sqsubseteq$ we have only outgoing edges. Since both, $n_{R_\sqsupseteq}$ and $n_{R_\sqsubseteq}$ are 0, we can eliminate the concept from in $R_\sqsubseteq$ by omitting its subsumers. In $R_\sqsupseteq$, we have three incoming and one outgoing edge, i.e., Inequation (5) does not hold. The number of edges decreases also in this case, since two of the conjunction edges obtained by rewriting are trivial as specified in the last section and are removed directly after the rewriting. In case of B_2, we only need to consider $R_\sqsubseteq$, since in $R_\sqsupseteq$ the concept is already unreferenced. Since we again have one incoming and one outgoing edge and $n_{R_\sqsubseteq}$ is 0, we can also perform the corresponding rewriting and eliminate B_2, thereby obtaining the relation pair shown in Fig. 3(a).

Now, we can also perform rewriting w.r.t. A_{10}, A_{13} in $R_\sqsupseteq$, since Condition (2) does not hold any more. Checking for unnecessary subsumees and subsumers reveals that both, $n_{R_\sqsupseteq}$ and $n_{R_\sqsubseteq}$ are still 0 for both, A_{10} and A_{13}. Since Inequation (5) does not hold in any of the two graphs, we can perform the corresponding rewriting and eliminate both, A_{10} and A_{13}, thereby obtaining the relation pair shown in Fig. 3(b).

We recall that, in Example 1, $\Sigma = \{A_1, A_8, A_{12}, A_{15}, A_{16}, r\}$. Thus, the only atomic concept not from Σ still referenced within the subsumee/subsumer relations is A_9, which is not eligible for rewriting due to Condition (4). Therefore, the rewriting process is finished. After computing $\mathtt{M}(R_\sqsupseteq, R_\sqsubseteq, \Sigma^{\mathtt{ext}}(R_\sqsupseteq, R_\sqsubseteq))$, we obtain the smaller of the two general modules given in Example 1.

Algorithm 1 shows the rewriting process starting with the initial subsumee/subsumer relation pair $\langle R^0_\sqsupseteq, R^0_\sqsubseteq \rangle$. The computation terminates, when no further subsumees/subsumers could be eliminated during one iteration. We obtain a rewritten subsumee/subsumer relation pair $\langle R_\sqsupseteq, R_\sqsubseteq \rangle$ over $\mathcal{T}$ complete w.r.t. Σ, which is of a polynomial size in the size of the original (not normalized) knowledge base $\mathcal{T}_o$ and does not contain any nested concept expressions. Moreover, after replacing all temporary concept names in $\mathtt{M}(R_\sqsupseteq, R_\sqsubseteq, \Sigma^{\mathtt{ext}}(R_\sqsupseteq, R_\sqsubseteq))$ by their definitions, we obtain a general module of $\mathcal{T}_o$, which does not contain any structurally new concept expressions not occurring in $\mathcal{T}_o$. We can summarize the results as follows.

Theorem 3. *Let $\mathcal{T}$ be a normalization of an $\mathcal{EL}$ knowledge base $\mathcal{T}_o$, $\Sigma \subseteq sig(\mathcal{T}_o)$ a signature, $\langle R_\sqsupseteq, R_\sqsubseteq \rangle$ the output of Algorithm 1 for an initial subsumee/subsumer pair $\langle R^0_\sqsupseteq, R^0_\sqsubseteq \rangle$ of $\mathcal{T}$. For $\mathcal{T}_r$ the knowledge base obtained by replacing all temporary concept names in $\mathtt{M}(R_\sqsupseteq, R_\sqsubseteq, \Sigma^{\mathtt{ext}}(R_\sqsupseteq, R_\sqsubseteq))$ by their definitions:*

Algorithm 1: Rewriting of Subsumee/Subsumer Relation Pairs

```
Data: $\langle R^0_\sqsupseteq, R^0_\sqsubseteq \rangle$ initial subsumee/subsumer relation pair for a normalized knowledge
      base
Result: $\langle R_\sqsupseteq, R_\sqsubseteq \rangle$ rewritten subsumee/subsumer relation pair
$\langle R_\sqsupseteq, R_\sqsubseteq \rangle \leftarrow \langle R^0_\sqsupseteq, R^0_\sqsubseteq \rangle$;
while fixpoint is not reached do
    for $B \in sig_C(\mathcal{T}) \setminus \Sigma$ do
        if Conditions (1)–(4) are false then
            $n_{R_\sqsupseteq} \leftarrow |R^{\mathtt{red}}_\sqsupseteq(B)|$;
            $n_{R_\sqsubseteq} \leftarrow |R^{\mathtt{red}}_\sqsubseteq(B)|$;
            if Inequation (5) does not hold then
                $R_\sqsupseteq \leftarrow \mathtt{Rew}_{R_\sqsupseteq}(B) \setminus (R_\sqsupseteq(B) \setminus R^{\mathtt{red}}_\sqsupseteq(B))$;
                $R_\sqsubseteq \leftarrow \mathtt{Rew}_{R_\sqsubseteq}(B) \setminus (R_\sqsubseteq(B) \setminus R^{\mathtt{red}}_\sqsubseteq(B))$;
return $\langle R_\sqsupseteq, R_\sqsubseteq \rangle$;
```

- $\mathtt{M}(R_\sqsupseteq, R_\sqsubseteq, \Sigma^{\mathtt{ext}}(R_\sqsupseteq, R_\sqsubseteq))$ *can be computed in polynomial time and is polynomial in the size of* $\mathcal{T}_o$*;*
- *for all sub-expressions* C' *occurring in* $\mathcal{T}_r$ *there is a sub-expression* C *of* $\mathcal{T}_o$ *such that* C' *can be obtained from* C *by exchanging atomic concepts.*

7 Evaluation

For our evaluation, we use the $\mathcal{EL}$ fragment of the Gene Ontology[5] describing gene product characteristics in terms of how gene products behave in a cellular context. The OWL version of the ontology (April 2012) comprises 36,251 atomic classes, 8 object properties and 316,580 logical axioms, out of which 66,117 axioms are terminological (the $\mathcal{EL}$ fragment contains 66,101 terminological axioms).

We implemented our approach in Java based on the OWL-API. The aim of the evaluation is to compare the results of our approach in terms of module size and computation time to minimal module extraction and *Locality-based extractor* [1] – an existing tractable approach to (not necessarily minimal) module extraction not based on rewriting. To the best of our knowledge, there are currently no existing implementations of minimal module extraction for $\mathcal{EL}$, but only for DL-Lite$_{\mathtt{bool}}$ [3]. Therefore, we compare the two implementations on the DL-Lite$_{\mathtt{bool}}$ fragment of $\mathcal{EL}$, obtained from an $\mathcal{EL}$ knowledge base by replacing qualified existential restrictions by the corresponding unqualified restrictions. In order to also estimate the difference in the module size for $\mathcal{EL}$, we implemented a module extractor based on minimal justifications, which, given a general module obtained using our approach, computes a subset of the original ontology entailing the general module.

For the evaluation, we use signatures with 10, 30 and 50 atomic concepts and 4 roles each. For each signature size, we randomly choose 10 signatures and let the dif-

[5] http://www.geneontology.org/

Table 1. Evaluation results on the DL-Lite$_{\texttt{bool}}$ fragment of $\mathcal{EL}$

Signature size	Rewriter	Minimal module extractor	Locality-based extractor
10	4.8	9.7 (2.0)	167 (34.8)
30	10.3	22.2 (2.2)	436 (41.1)
50	28.8	60.4 (2.1)	1245 (43.2)

Table 2. Evaluation results on $\mathcal{EL}$

Signature size	Rewriter	Minimal justification extractor	Locality-based extractor
10	21	43 (2.0)	259 (12.3)
30	45	104 (2.3)	659 (14.6)
50	151	306 (2.0)	1787 (11.8)

ferent extractors compute the corresponding general module. Subsequently, we compute the average module size, shown in Tables 1 and 2 (the number in brackets is the average module size measured in the corresponding average size of the modules computed by the rewriter). The first table shows the results for the DL-Lite$_{\texttt{bool}}$ fragment of $\mathcal{EL}$. Due to the lower expressivity, the obtained DL-Lite$_{\texttt{bool}}$ modules are considerably smaller than their $\mathcal{EL}$ correspondents in Table 2. We observe that the size of the minimal DL-Lite$_{\texttt{bool}}$ modules containing only axioms from the original knowledge base $\mathcal{T}$ are between 2.0 and 2.2 times larger than the corresponding general modules consisting of sub-expressions of $\mathcal{T}$ with possibly exchanged atomic concepts obtained using rewriter. The corresponding DL-Lite$_{\texttt{bool}}$ modules obtained by the locality-based extractor are even between 34.8 and 43.2 times larger. In case of $\mathcal{EL}$ modules, the minimal justifications of the general modules computed by rewriter are between 2.0 and 2.3 times larger, while the modules obtained by the locality-based extractor are between 11.8 and 14.6 times larger.

We further analyzed whether different proportions of particular axiom types influence the effectiveness of rewriter, but did not find this to be the case. Our conjecture is that this is a result of the simple structure of GO, which contains only axioms referencing exactly two atomic concepts, e.g., atomic subsumptions and existential restrictions. In case of such axioms, exactly two substitutions of atomic concepts are possible, each of which can potentially replace a justification consisting of several axioms, e.g., a chain of subsumption axioms. Since the effect of each such replacement is not dependent on the axiom type, but rather on the referenced concepts, this conjecture seems reasonable.

Concerning the computation time, we observe a significant difference between the tractable approaches (rewriter and the locality-based extractor) and the minimal module extractor. While, for the signatures with 50 atomic concepts, the first two approaches require less than one minute, minimal module extractor required between two hours and two days depending on the signature.

8 Summary

In this paper, we show that knowledge base extraction gains in effectiveness in terms of knowledge base size, when modules are not required to be subsets of the original knowledge base. We investigate the task of knowledge base extraction for $\mathcal{EL}$ based on two alternative, less restrictive notions for syntactic similarity.

First, we discuss the extraction of knowledge bases consisting only of sub-expressions occurring in the original knowledge base. We show how minimal modules fulfilling this similarity requirement can be obtained in EXPTIME by introducing temporary concept names for complex concepts, adding a subset of the deductive closure to the knowledge base and subsequently applying minimal module extraction.

Second, we consider the extraction of modules that consist of concepts structurally equivalent to sub-expressions occurring in the original knowledge base.We propose a tractable approach that, in most cases, yields small knowledge bases, but does not guarantee the minimality of the result. As we show in our evaluation, modules extracted during our evaluation using minimal module extractor for DL-Lite$_{\mathtt{bool}}$ are 2.0 to 2.2 times larger than those obtained by our approach. In case of $\mathcal{EL}$, knowledge bases obtained by our rewriter on average contain half as many axioms as their minimal justifications within the original knowledge base. In case of the locality-based module extractor, the extracted $\mathcal{EL}$ modules are on average 12 times larger than the general modules obtained by our approach.

Acknowledgements

This work is supported by the German Federal Ministry of Education and Research (BMBF) under grant 02PJ1002 (SyncTech).

References

1. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Just the right amount: extracting modules from ontologies. In: Proc. of the 16th Int. Conf. on World Wide Web (WWW-07). pp. 717–726 (2007)
2. Konev, B., Walther, D., Wolter, F.: Forgetting and uniform interpolation in large-scale description logic terminologies. In: Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI-09). pp. 830–835 (2009)
3. Kontchakov, R., Wolter, F., Zakharyaschev, M.: Logic-based ontology comparison and module extraction, with an application to DL-Lite. Artificial Intelligence 174, 1093–1141 (2010)
4. Lutz, C., Wolter, F.: Foundations for uniform interpolation and forgetting in expressive description logics. In: Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI-11) (2011)
5. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C. (eds.): OWL 2 Web Ontology Language: Profiles. W3C Recommendation (27 October 2009), available at http://www.w3.org/TR/owl2-profiles/
6. Nikitina, N., Glimm, B.: Hitting the sweetspot: Economic rewriting of knowledge bases. Techreport, AIFB, KIT, Karlsruhe (Mai 2012)
7. Nikitina, N., Rudolph, S.: ExpExpExplosion: Uniform interpolation in general EL terminologies. In: Proc. of the 20th European Conf. on Artificial Intelligence (ECAI-12) (2012)
8. Nikitina, N., Rudolph, S., Glimm, B.: Reasoning-supported interactive revision of knowledge bases. In: Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI-11). pp. 1027–1032 (2011)
9. OWL Working Group, W.: OWL 2 Web Ontology Language: Document Overview. W3C Recommendation (27 October 2009), available at http://www.w3.org/TR/owl2-overview/