Computing Minimal Equivalent Acyclic $\mathcal{EL}$ Ontologies

Patrick Koopmann and Nadeschda Nikitina

Department of Computer Science, University of Oxford, UK

Abstract. Computing equivalent $\mathcal{EL}$-ontologies of minimal size is useful for various reasoning tasks such as uniform interpolation, ontology learning, rewriting ontologies into simpler DLs, abduction and knowledge revision. The corresponding tool support for minimising concepts and ontologies can also provide great help to ontology developers and end-users. We present a method for computing equivalent acyclic $\mathcal{EL}$-ontologies of minimal size, where size is measured by number of occurrences of predicates. We applied our minimisation method to both known and generated ontologies with promising results.

1 Introduction

Logical languages allow to represent the same facts in different equivalent ways, whose complexity can vary significantly. Description logics are used to model ontologies describing 100,000s of terms. Understanding ontologies of this size is further hindered by unnecessary verbosity. For example, the following are two simplified versions of axioms found in the Galen ontology:

$$\begin{aligned}\texttt{TrueCavity} \equiv\ & \texttt{BodyCavity}\\ & \sqcap \exists\texttt{isDefinedBy}.\exists\texttt{hasTopology}.\exists\texttt{hasState}.\texttt{trulyHollow} \qquad (1)\end{aligned}$$

$$\texttt{TruelyHollowBodyStructure} \equiv \exists\texttt{hasTopology}.\exists\texttt{hasState}.\texttt{trulyHollow} \qquad (2)$$

Axiom 1 seems unnecessarily verbose, especially since Galen already contains a definition for the concept `TruelyHollowBodyStructure`. Using this concept, we can reformulate Axiom 1, obtaining a much more concise and accessible definition for `TrueCavity`, while preserving all logical consequences of the ontology:

$$\texttt{TrueCavity} \equiv \texttt{BodyCavity} \sqcap \exists\texttt{isDefinedBy}.\texttt{TruelyHollowBodyStructure} \qquad (3)$$

Arguably, Axiom 3 is more accessible than Axiom 1, as the user has to parse less information when reading it. Furthermore, Axiom 3 makes more use of the structure provided by the ontology, and contains less redundant information.

In this paper, we investigate the task of computing equivalent ontologies of minimal size, where size is defined as the number of occurrences of concept and role symbols in the ontology, or as the sum of its axiom sizes. The corresponding decision problem is NP-complete. Automatic simplification of ontologies can benefit a range of areas, of which we give some examples. **Optimise existing ontologies.** Simplifying large ontologies by hand can be cumbersome, as the developer needs full knowledge of all concepts available in the ontology and has to consider numerous alternatives. A tool that automatically detects and removes redundancy and simplifies concept expressions could provide great help to ontology developers and end-users. **(Semi-)Automated Ontology Generation.** There is a variety of techniques to generate ontologies automatically or semi-automatically from different data sources such as tables or text documents [27, 5]. Ontologies generated this way usually require additional manual work, and the quality of the presentation will be worse than that of hand-crafted ontologies. Reducing redundancy and verbosity automatically has the potential to significantly improve the process of generating ontologies this way. **Non-Standard Reasoning Services.** There is an increasing number of non-standard reasoning services that generate sets of axioms. Examples include methods for *uniform interpolation* [16, 21, 18], *ontology learning* [13, 17], *rewriting ontologies into simpler DLs* [3, 19], *abduction* [6, 12] and *knowledge revision* [7, 24]. Usually, the corresponding tools use heuristics for removing redundancies. In contrast, using our method as a post-processing task in these procedures results in an optimal representation of the results.

While there exist solutions for certain sub-problems such as removing redundant axioms [8] or minimising $\mathcal{EL}$ concepts [2], the only existing algorithm for computing ontologies of minimal size only supports $\mathcal{EL}$-ontologies that do not use conjunctions [22]. We present a method that applies to a more general class of $\mathcal{EL}$-ontologies, based on a semantic acyclity condition defined in Section 4. If the ontology is acyclic, minimised ontologies can be computed in an incremental manner. For this, the ontology is divided into partitions that are minimised one after another. The minimisation of partitions involves the computation of minimal equivalent $\mathcal{EL}$-concepts—a problem which on its own is already NP-complete, and for which we provide a practical method based on regular tree grammars for generating subsumers introduced in [21]. An implementation of our method is available online.[1] Full proofs of lemmas and theorems are provided in the long version of the paper [14].

2 Preliminaries

2.1 The Description Logic $\mathcal{EL}$

We recall the description logic $\mathcal{EL}$ used in this paper [1]. Let N_c and N_r be countably infinite and mutually disjoint sets of concept symbols and role symbols. $\mathcal{EL}$ *concepts* C are defined as

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

where $A \in N_c$ and $r \in N_r$. For a set $\mathcal{C} = \{C_1, \ldots, C_n\}$, we may abbreviate the corresponding conjunction $C_1 \sqcap \ldots \sqcap C_n$ with $\bigsqcap \mathcal{C}$. If a concept is of the form A, $A \in N_c$, it is *atomic*. Otherwise it is *complex*. An $\mathcal{EL}$ *ontology* or *TBox* consists of *concept inclusion axioms* $C \sqsubseteq D$ and *equivalence axioms* $C_1 \equiv \cdots \equiv C_n$ used

[1] http://users.ox.ac.uk/~coml0607/el_minimiser/

as a shorthand for $C_i \sqsubseteq C_j$, $i, j \leq n$. Equivalence axioms like this correspond to axioms of type `OWLEquivalentClasses` defined in the standard ontology language OWL [10]. For a set $\mathcal{C} = \{C_1, \ldots, C_n\}$, we may abbreviate the axiom $C_1 \equiv \ldots \equiv C_n$ with $\equiv(\mathcal{C})$.

We denote by $sub(C)$ and $sub(\mathcal{T})$ the concepts occurring syntactically in C and $\mathcal{T}$. The semantics used is standard (see for example [1]). In particular, for an ontology $\mathcal{T}$ and an axiom α, we use the notation $\mathcal{T} \models \alpha$ to express that α is *entailed by* $\mathcal{T}$, that is α is true in every model of $\mathcal{T}$, and $\mathcal{T}_1 \equiv \mathcal{T}_2$ to express that $\mathcal{T}_1$ and $\mathcal{T}_2$ are *equivalent*, that is they have the same models.

2.2 Regular Tree Grammars

A *regular tree grammar* is a tuple $G = \langle \mathfrak{n}_s, \mathcal{N}, \mathcal{F}, \mathcal{R} \rangle$, composed of a *start symbol* $\mathfrak{n}_s$, a set $\mathcal{N}$ of *non-terminal symbols* such that $\mathfrak{n}_s \in \mathcal{N}$, a ranked alphabet $\mathcal{F}$ of *terminal symbols* such that $\mathcal{N} \cap \mathcal{F} = \emptyset$, and a set $\mathcal{R}$ of *derivation rules* of the form $\mathfrak{n} \to_R \beta$, where $\mathfrak{n} \in \mathcal{N}$ and β is a term over $\mathcal{N} \cup \mathcal{F}$. A context $C[X]$ is a term in which one subterm is replaced by a variable X. Given a regular tree grammar $G = \langle \mathfrak{n}_s, \mathcal{N}, \mathcal{F}, \mathcal{R} \rangle$, the derivation relation $\to_G$ is a relation on terms over $\mathcal{N} \cup \mathcal{F}$ such that $t_1 \to_G t_2$ iff there is a derivation rule $\mathfrak{n} \to_R \beta \in R$ and a context $C[X]$ such that $t_1 = C[X \mapsto \mathfrak{n}]$ and $t_2 = C[X \mapsto \beta]$. We denote by $\to_G^+$ the transitive closure of $\to_G$. The *language generated by* G, denoted by $L(G)$, is the set of ground terms t over $\mathcal{F}$ such that $\mathfrak{n}_s \to_G^+ t$.

3 Minimising $\mathcal{EL}$ Concepts

Before we study the problem of minimising $\mathcal{EL}$-ontologies, we focus on minimising $\mathcal{EL}$-concepts, which plays a central role in our approach. We define the $\int$-size of an $\mathcal{EL}$ concept inductively by $\int(\top) = \int(A) = 1$ for all $A \in N_c$, $\int(\exists r.C) = \int(C) + 1$ and $\int(C_1 \sqcap C_2) = \int(C_1) + \int(C_2)$. Informally, the size of a concept corresponds to the number occurrences of concept and role symbols, as well as of the $\top$ concept.

The notion of minimal concepts is captured in the following definition:

Definition 1. *Given an $\mathcal{EL}$ ontology $\mathcal{T}$ and an $\mathcal{EL}$ concept C, C is* minimal *in $\mathcal{T}$ if there is no concept C' with $\mathcal{T} \models C \equiv C'$ and $\int(C') < \int(C)$. For an ontology $\mathcal{T}$ and a concept C, we denote by $min_c(C, \mathcal{T})$ the set of minimal concepts C' with $\mathcal{T} \models C \equiv C'$. For an ontology $\mathcal{T}$ and two $\mathcal{EL}$ concepts C, C_c, we denote by $min_c(C, \mathcal{T}, C_c)$ the set of* conditioned minimal concepts *C' for which $\mathcal{T} \models C_c \sqcap C \equiv C_c \sqcap C'$ and for which there is no concept C'' with $\int(C'') < \int(C')$ and $\mathcal{T} \models C_c \sqcap C \equiv C_c \sqcap C''$.*

Intuitively, minimising an ontology involves minimising the concepts occurring in it. Whereas $min_c(C, \mathcal{T})$ contains all minimal concepts equivalent to C, the set $min_c(C, \mathcal{T}, C_c)$ of conditioned minimal concepts contains concepts that are minimal in conjunction with a fixed concept C_c. Conditioned minimal concepts can be used to determine minimal concept inclusion axioms. Take as example the axiom $\exists r.A \sqsubseteq B \sqcap \exists r.\top$. The concept $B \sqcap \exists r.\top$ is minimal with respect to the empty ontology. However, since $\exists r.A \sqsubseteq \exists r.\top$ already follows from the empty ontology, the axiom $\exists r.A \sqsubseteq B$ is equivalent and uses only B on the right hand side. By fixing $\exists r.A$, as a conjunct, we obtain the desired concept for the axiom, since $\models \exists r.A \sqcap (\exists r.\top \sqcap B) \equiv \exists r.A \sqcap B$, and $min_c(B \sqcap \exists r.\top, \emptyset, \exists r.A) = \{B\}$.

The decision problem corresponding to minimising $\mathcal{EL}$-concepts—is there a concept C_2 for a given $\mathcal{T}$, C_1 and k such that $\mathcal{T} \models C_1 \equiv C_2$ and $\int(C_1) \leq k$—is NP-complete [2]. Since equivalence between general $\mathcal{EL}$ ontologies can be decided in polynomial time, from this follows also the NP-completeness of the corresponding decision problems for minimising $\mathcal{EL}$ concepts with arbitrary $\mathcal{EL}$ ontologies, as well as for minimising cyclic or acyclic $\mathcal{EL}$ ontologies. The upper-bound follows since we can non-deterministically guess a solution of size k and test for equivalence in polynomial time (see also [22]). Our method for computing minimal or conditioned minimal equivalent $\mathcal{EL}$ concepts makes use of regular tree grammars. Despite the exponential worst-case complexity of this approach, practicality can be achieved by using best-first search and further optimisations, which is briefly discussed in the evaluation section.

We define $\mathcal{F}^{\mathcal{EL}}$ as the symbols constituting the logic $\mathcal{EL}$, that is, $\mathcal{F}^{\mathcal{EL}} = N_c \cup N_r \cup \{\sqcap, \exists\}$. For a given ontology $\mathcal{T}$ and concept C_s, we define the set $\mathcal{N}^{\mathcal{T},C_s} = \{\mathfrak{n}_C \mid C \in sub(\mathcal{T}) \cup sub(C_s) \cup \{\top\}\}$ which contains a non-terminal symbol $\mathfrak{n}_C$ for every concept C occurring in $\mathcal{T}$ or C_s. Given an ontology $\mathcal{T}$ and a concept C_s, the grammar $G^{\sqsubseteq}(\mathcal{T}, C_s)$ is then given by $\langle \mathfrak{n}_{C_s}, \mathcal{N}^{\mathcal{T},C_s}, \mathcal{F}^{\mathcal{EL}}, \mathcal{R}\rangle$, where $\mathcal{R}$ contains the following derivation rules:

(R1): $\mathfrak{n}_C \rightarrow_R C$ for all $\mathfrak{n}_C \in \mathcal{N}^{\mathcal{T},C_s}$
(R2): $\mathfrak{n}_C \rightarrow_R \exists r.\mathfrak{n}_D$ for all $\mathfrak{n}_C, \mathfrak{n}_D \in \mathcal{N}^{\mathcal{T},C_s}$ such that $C = \exists r.D$
(R3): $\mathfrak{n}_C \rightarrow_R \mathfrak{n}_{C_1} \sqcap \ldots \sqcap \mathfrak{n}_{C_n}$ for all $\mathfrak{n}_C \in \mathcal{N}^{\mathcal{T},C_s}$ and sets $\{\mathfrak{n}_{C_1}, \ldots, \mathfrak{n}_{C_n}\} \subseteq \mathcal{N}^{\mathcal{T},C_s}$ such that $\mathcal{T} \models C \sqsubseteq C_i$, $1 \leq i \leq n$.

In practice, instances of rule (R3) can be determined by flattening and classifying the ontology using any standard description logic reasoner. The generation of the other rules is trivial. We extend the notion of $\int$-size to terms generated by $G^{\sqsubseteq}(\mathcal{T}, C_s)$ by setting $\int(\mathfrak{n}_C) = 1$ for all $\mathfrak{n}_C \in \mathcal{N}^{\mathcal{T},C_s}$. Furthermore, we denote by $Con(t, G)$ the result of saturating the term t with derivation rules of type (R1), that is, by replacing every non-terminal with the corresponding concept. Note that there is exactly one rule of type (R1) for every non-terminal, so that $Con(t, G)$ is always uniquely defined.

Example 1. Consider the following ontology $\mathcal{T}_1$:

$$B \sqcap \exists r.\exists s.A_1 \sqsubseteq \exists r.\exists t.A_2 \qquad \exists t.A_2 \sqsubseteq A_3 \qquad A_3 \sqsubseteq \exists s.A_1$$

The set $\mathcal{N}^{\mathcal{T}_1}$ contains non-terminals for the concepts $B \sqcap \exists r.\exists s.A_1$, B, $\exists r.\exists s.A_1$, $\exists s.A_1$, A_1, $\exists r.\exists t.A_2$, $\exists t.A_2$, A_2, A_3 and $\top$. We show an example derivation in the grammar $G^{\sqsubseteq}(\mathcal{T}_1, B \sqcap \exists r.\exists s.A_1)$.

$$\begin{aligned} \mathfrak{n}_{B \sqcap \exists r.\exists s.A_1} &\xrightarrow{(R3)}_G \mathfrak{n}_B \sqcap \mathfrak{n}_{\exists r.\exists t.A_2} \xrightarrow{(R1)}_G B \sqcap \mathfrak{n}_{\exists r.\exists t.A_2} \\ &\xrightarrow{(R2)}_G B \sqcap \exists r.\mathfrak{n}_{\exists t.A_2} \xrightarrow{(R3)}_G B \sqcap \exists r.\mathfrak{n}_{A_3} \xrightarrow{(R1)}_G B \sqcap \exists r.A_3 \end{aligned}$$

The applied rules of type (R3) are due to the entailments $\mathcal{T} \models B \sqcap \exists r.\exists s.A_1 \sqsubseteq B$, $\mathcal{T} \models B \sqcap \exists r.\exists s.A_1 \sqsubseteq \exists r.\exists t.A_2$ and $\mathcal{T} \models \exists t.A_2 \sqsubseteq A_3$. One can show that $\mathcal{T} \models B \sqcap \exists r.\exists s.A_1 \equiv B \sqcap \exists r.A_3$. In fact, one cannot derive a smaller equivalent concept.

It is shown in [21] that for every pair of $\mathcal{EL}$ concepts C, D such that no concept occurs twice in a conjunction in D, $\mathcal{T} \models C \sqsubseteq D$ iff D is generated by $G^{\sqsubseteq}(\mathcal{T}, C)$. Therefore, $G^{\sqsubseteq}(\mathcal{T}, C)$ generates all subsumers of C that are candidates for minimal equivalent concepts of C in $\mathcal{T}$. In order to compute an element of $min_c(C, \mathcal{T})$ or $min_c(C, \mathcal{T}, C_c)$, we search the space of concepts generated by $G^{\sqsubseteq}(\mathcal{T}, C)$. In order to limit the search space we make use of the following properties, which can be shown by inspection of the derivation rules.

Lemma 1. *Let $\mathcal{T}$ be an acyclic $\mathcal{EL}$ ontology, C an $\mathcal{EL}$ concept and $G = G^{\sqsubseteq}(C, \mathcal{T})$. Further, let t_1, t_2 be terms over $\mathcal{N}^{\mathcal{T}} \cup \mathcal{F}^{\mathcal{T}}$.*

- *If $t_1 \rightarrow_G t_2$, then $\mathcal{T} \models Con(t_1) \sqsubseteq Con(t_2)$. ($\sqsubseteq$-monotonicity)*
- *If $t_1 \rightarrow_G t_2$, then $\int(t_1) \leq \int(t_2)$. ($\int$-monotonicity)*

Due to the $\int$-monotonicity, we do not have to follow derivations from concepts C' for which $\int(C') > \int(C^m)$, where C^m is the currently known smallest equivalent concept of C. Note that this way, we have to check at most exponentially many derivations. Due to the $\sqsubseteq$-monotonicity, we do not have to follow derivations of concepts C' such that $\mathcal{T} \not\models C' \sqsubseteq C$, where C is the concept to minimise.

4 Minimising Acyclic Ontologies

We describe our method for minimising acyclic $\mathcal{EL}$ ontologies. The notion of $\int$-size is extended as follows to $\mathcal{EL}$ axioms and TBoxes: $\int(C \sqsubseteq D) = \int(C) + \int(D)$, $\int(C_1 \equiv \ldots \equiv C_n) = \Sigma_{1 \leq i \leq n} C_i$, $\int(\mathcal{T}) = \Sigma_{\alpha \in \mathcal{T}} \int(\alpha)$. An $\mathcal{EL}$ ontology $\mathcal{T}^m$ is *minimal* if there is no ontology $\mathcal{T}$ such that $\mathcal{T} \equiv \mathcal{T}^m$ and $\int(\mathcal{T}) < \int(\mathcal{T}^m)$.

We focus on computing minimal equivalent ontologies for a class of $\mathcal{EL}$ ontologies that is characterized by the following definition.

Definition 2. *Let $\mathcal{T}$ be an $\mathcal{EL}$ ontology. $\mathcal{T}$ is* acyclic *iff there are no $\mathcal{EL}$ concepts C, D such that $C \in sub(D)$ and $\mathcal{T} \models C \sqsubseteq \exists r.D$.*

Note that since the definition of acyclicity is defined purely semantically, acyclicity is robust under logical equivalence: given two equivalent ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$, $\mathcal{T}_1$ is acyclic if and only if $\mathcal{T}_2$ is acyclic.

Acyclic ontologies have the following property, which facilitates the minimisation of ontologies compared to cyclic ontologies.

Lemma 2. *Let $\mathcal{T}^m$ be an $\int$-minimal acyclic ontology. Further, let $\alpha \in \mathcal{T}^m$ be of the form $C_1 \sqsubseteq C_2$ or $C_1 \equiv \ldots \equiv C_n$. Then, every equivalent $\mathcal{EL}$ ontology $\mathcal{T}$ contains an axiom β of the form $C'_1 \sqsubseteq C'_2$ or $C'_1 \equiv \ldots \equiv C'_m$ such that $\mathcal{T} \models C_1 \equiv C'_1$.*

Due to this lemma, it is sufficient to consider axioms for the minimised ontology whose left-hand side is equivalent to the left-hand side of axioms in the input ontology. This allows to partition the axioms in the ontology based on their left-hand side concepts, and minimise the partitions one after the other.

4.1 Structuring the input ontology

To formalise this idea, we first group equivalent concepts in the ontology. For a concept C, we denote by $[C]_\mathcal{T}$ the *equivalence class of* C, the concepts that syntactically occur in $\mathcal{T}$ and are equivalent to C: $[C]_\mathcal{T} = \{C' \mid C \in sub(\mathcal{T}), \mathcal{T} \models C \equiv C'\}$. For each equivalence class $[C]_\mathcal{T}$, we define the before-mentioned partitions as $\mathcal{T}_{[C]} = \{\alpha \mid \alpha = C_1 \sqsubseteq C_2 \text{ or } \alpha = C_1 \equiv \ldots \equiv C_n,, C_1 \in [C]_\mathcal{T}\}$. By Lemma 2, each non-empty partition $\mathcal{T}^m_{[C]}$ in a minimal ontology $\mathcal{T}^m$ has a corresponding non-empty partition $\mathcal{T}_{[C]}$ in any equivalent ontology $\mathcal{T}$.

Note that in Example 1, if we replace the concept $B \sqcap \exists r.\exists s.A_1$ with its minimal equivalent $B \sqcap \exists r.A_3$, we obtain an ontology that is not equivalent. In order to minimise a partition, we can only take into account entailments from axioms outside of that partition. For example, all elements in $[C]_\mathcal{T}$ are equivalent to the same concept C^m minimal with respect to $\mathcal{T}$, but if we replace equivalence axioms in $\mathcal{T}_{[C]}$ by the tautological axiom $C^m \equiv C^m$, we do not obtain an equivalent ontology. To determine which axioms of the ontology have to be considered when minimising a partition $\mathcal{T}_{[C]}$, we structure the ontology based on an implicability relation $\rightsquigarrow_\mathcal{T}$ between axioms and equivalence classes. Intuitively, if we have $\alpha \rightsquigarrow_\mathcal{T} \beta$, for some $\alpha \in \mathcal{T}_{[C]}$, α affects the meaning of β, and we should not take β into account when minimising $\mathcal{T}_{[C]}$.

We define the relation $\rightsquigarrow_\mathcal{T}$ formally. For two $\mathcal{EL}$ axioms α, β, $\alpha \rightsquigarrow_\mathcal{T} \beta$ iff there is a TBox $\mathcal{T}'$ with $\mathcal{T} \models \mathcal{T}'$, such that $\alpha \in \mathcal{T}'$, $\mathcal{T}' \models \beta$ and $\mathcal{T}' \setminus \{\alpha\} \not\models \beta$.

Example 2. Take the ontology $\mathcal{T}_1$ used in the last example, and the two axioms $\alpha = B \sqcap \exists r.A_1 \sqsubseteq \exists r.\exists t.A_2$ and $\beta = B \sqcap \exists r.\exists s.A_1 \equiv B \sqcap \exists r.A_3$. We observed in the example that $\mathcal{T}_1 \models \beta$. Set $\mathcal{T}' = \mathcal{T}_1$. We have $\mathcal{T}_1 \models \mathcal{T}'$, $\mathcal{T}' \models \beta$ and $\mathcal{T}' \setminus \{\alpha\} \not\models \beta$. Therefore, $\alpha \rightsquigarrow_{\mathcal{T}_1} \beta$. In the same way, we can establish $\exists t.A_2 \sqsubseteq A_3 \rightsquigarrow_{\mathcal{T}_1} \beta$ and $A_3 \sqsubseteq \exists s.A_1 \rightsquigarrow_{\mathcal{T}_1} \beta$.

In acyclic $\mathcal{EL}$ ontologies, the only cycles in the implicability relation are between axioms of the same partition.

Lemma 3. *Let $\mathcal{T}$ be an $\exists$-acyclic $\mathcal{EL}$ ontology and let α, β be entailments of $\mathcal{T}$ with $\alpha = C_\alpha \sqsubseteq D_\alpha$ and $\beta = C_\beta \sqsubseteq D_\beta$. Further, let $\beta \rightsquigarrow_\mathcal{T} \alpha$ and $\alpha \rightsquigarrow_\mathcal{T} \beta$. Then, $\mathcal{T} \models C_\alpha \equiv C_\beta$.*

We extend $\rightsquigarrow_\mathcal{T}$ to equivalence classes. $[C]_\mathcal{T} \rightsquigarrow_\mathcal{T} [D]_\mathcal{T}$ iff there are axioms $\alpha_C = C_1 \sqsubseteq C_2$ and $\alpha_D = D_1 \sqsubseteq D_2$, $\mathcal{T} \models C_1 \equiv C, D_1 \equiv D$, such that $\alpha_C \rightsquigarrow_\mathcal{T} \alpha_D$. As a consequence of Lemma 3, the only cycles in the implicability relation between equivalence classes are due to reflexivity of the relation. The ontology $\mathcal{T}^{in}_{[C]} = \bigcup\{\mathcal{T}_{[D]} \mid [D]_\mathcal{T} \rightsquigarrow_\mathcal{T} [C]_\mathcal{T}, [D]_\mathcal{T} \neq [C]_\mathcal{T}\}$ contains all axioms in $\mathcal{T}$ that may have an impact on the minimised version of $\mathcal{T}_{[C]}$.

Lemma 4. *Let $\mathcal{T}$ be an $\mathcal{EL}$ ontology and $C, D \in sub(\mathcal{T})$ be two $\mathcal{EL}$ concepts such that $[C]_\mathcal{T} \rightsquigarrow_\mathcal{T} [D]_\mathcal{T}$. Then, one of the following is true.*

1. *$\mathcal{T} \models D \sqsubseteq C$.*
2. *$\mathcal{T} \models D \equiv \exists r.C$.*

Algorithm 1: Algorithm for computing minimal equivalent ontologies.

Data: $\mathcal{T}$: $\exists$-acyclic $\mathcal{EL}$ ontology
Result: $\mathcal{T}^m$: $\int$-minimal equivalent ontology
$\mathcal{T}^m := \emptyset$;
$\mathcal{P}_{todo} := \{[C]_\mathcal{T} \mid C \sqsubseteq D \in \mathcal{T}\}$;
while $\mathcal{P}_{todo} \neq \emptyset$ **do**
    Choose $[C]_\mathcal{T} \in \mathcal{P}_{todo}$ with $[D]_\mathcal{T} \notin \mathcal{P}_{todo}$ for all $[D]_\mathcal{T} \leadsto_\mathcal{T} [C]_\mathcal{T}$,$[D]_\mathcal{T} \neq [C]_\mathcal{T}$;
    $\mathcal{T}^m := \mathcal{T}^m \cup \texttt{minimise}(\mathcal{T}_{[C]}, \mathcal{T}^m)$;
    $\mathcal{P}_{todo} := \mathcal{P}_{todo} \setminus \{[C]\}$;
return $\mathcal{T}^m$;

3. *There is a concept C_2 such that $[C_2]_\mathcal{T} \notin \{[C]_\mathcal{T}, [D]_\mathcal{T}\}$, $[C]_\mathcal{T} \leadsto_\mathcal{T} [C_2]_\mathcal{T}$ and $[C_2]_\mathcal{T} \leadsto_\mathcal{T} [D]_\mathcal{T}$.*

Lemma 4 allows to compute a super-relation of the implicability relation that is sufficient for our purposes. Condition 1 can be checked by flattening and classifying the ontology, Condition 2 is syntactical, and Condition 3 corresponds to the transitive closure.

Example 3. There are three non-empty partitions in $\mathcal{T}_1$, corresponding to the equivalence classes $[B \sqcap \exists r.\exists s.A_1]_{\mathcal{T}_1}$, $[\exists t.A_2]_{\mathcal{T}_1}$ and $[A_3]_{\mathcal{T}_1}$. Based on the observations in Example 2, we have $[\exists t.A_1]_{\mathcal{T}_1} \leadsto_{\mathcal{T}_1} [B \sqcap \exists r.\exists s.A_1]_{\mathcal{T}_1}$ and $[A_3]_{\mathcal{T}_1} \leadsto_{\mathcal{T}_1} [B \sqcap \exists r.\exists s.A_1]_{\mathcal{T}_1}$. Using Lemma 4 and $\mathcal{T}_1 \models \exists t.A_1 \sqsubseteq A_3$, we can further establish $[A_3]_{\mathcal{T}_1} \leadsto_{\mathcal{T}_1} [\exists t.A_1]_{\mathcal{T}_1}$. The ontology $\mathcal{T}^{in}_{1[B\sqcap\exists r.\exists s.A_1]}$ contains the union of $\mathcal{T}_{1[\exists t.A_1]}$ and $\mathcal{T}_{1[A_3]}$, that is, the last two axioms of the ontology. Therefore, in order to minimise partition $\mathcal{T}_{1[B\sqcap\exists r.A_3]}$, we are only allowed to take into account entailments from the last two axioms, which means the equivalence $\mathcal{T}_1 \models B \sqcap \exists r.\exists s.A_1 \equiv B \sqcap \exists r.A_3$, which depends on all axioms, cannot be used.

Theorem 1. *Let $\mathcal{T}^1$ and $\mathcal{T}^2$ be two acyclic $\mathcal{EL}$ ontologies s.t. $\mathcal{T}^1 \equiv \mathcal{T}^2$, and C be any $\mathcal{EL}$ concept. Then, the following statements are true:*

1. $(\mathcal{T}^1)^{in}_{[C]} \equiv (\mathcal{T}^2)^{in}_{[C]}$
2. $(\mathcal{T}^1)^{in}_{[C]} \cup \mathcal{T}^1_{[C]} \equiv (\mathcal{T}^1)^{in}_{[C]} \cup \mathcal{T}^2_{[C]}$

Note that Theorem 1 also holds for any minimal equivalent ontology $\mathcal{T}^2$. We can therefore construct a minimal equivalent ontology by first computing a minimal ontology equivalent to $(\mathcal{T}^1)^{in}_{[C]}$, and then a minimal extension equivalent to $(\mathcal{T}^1)^{in}_{[C]} \cup \mathcal{T}^1_{[C]}$. By starting with the concepts C for which $(\mathcal{T}^1)^{in}_{[C]}$ is empty, we can compute minimal equivalent ontologies in an incremental way. In each step, we extend the current minimal ontology with the next partition, until all partitions are processed. An overview of the corresponding top-level procedure is shown in Algorithm 1. The algorithm makes use of a procedure $\texttt{minimise}(\mathcal{T}_{[C]}, \mathcal{T}^m)$ that minimises partitions $\mathcal{T}_{[C]}$ against the already constructed ontology $\mathcal{T}^m$. This is described in the next subsection.

4.2 Computing Minimal Partitions

The concepts occurring in a minimised partition can be computed solely based on the concepts that occur in the original partition $\mathcal{T}_{[C]}$, making use of logical relations that follow from $\mathcal{T}^{in}_{[C]}$. Note that $[C]_\mathcal{T}$ contains all concepts that occur on the left-hand side of an axiom in $\mathcal{T}^{in}_{[C]}$. We further define the set $\mathcal{S}(C)_\mathcal{T} = \{C_s \mid C_1 \sqsubseteq C_s \in \mathcal{T}_{[C]} \text{ or } C_1 \equiv \ldots \equiv C_s \equiv \ldots \equiv C_n \in \mathcal{T}_{[C]}\}$, which contains the corresponding concepts on the right-hand sides. $[C]_\mathcal{T}$ and $\mathcal{S}(C)_\mathcal{T}$ encode all information in $\mathcal{T}_{[C]}$: we obtain an equivalent ontology if we replace $\mathcal{T}_{[C]}$ by $\{\equiv([C]_\mathcal{T}), C \sqsubseteq \bigsqcap \mathcal{S}(C)_\mathcal{T})\}$. In the remainder of the section, we specify how to remove all redundancy from these sets, and how to determine the shape of the minimised partition.

We first specify a minimal subset of the concepts in $[C]_\mathcal{T}$ whose equivalence has to be expressed in the partition of any equivalent ontology.

Definition 3. *A set $\mathcal{C}$ of concepts is* equivalence-reduced against $\mathcal{T}$ *if there are no distinct concepts $C_1, C_2 \in \mathcal{C}$ such that $\mathcal{T} \models C_1 \equiv C_2$.*

A set $[C]^{\equiv}_\mathcal{T}$ is a minimal set of required equivalent concepts in $\mathcal{T}_{[C]}$ *if $[C]^{\equiv}_\mathcal{T}$ is a maximal, equivalence-reduced subset of $[C]_\mathcal{T}$ against $\mathcal{T}^{in}_{[C]}$ such that there are no concepts $C_1 \in [C]^{\equiv}_\mathcal{T}$, $C_2 \in [C]_\mathcal{T}$ with $\mathcal{T}^{in}_{[C]} \not\models C_1 \equiv C_2$ and $\mathcal{T}^{in}_{[C]} \models C_1 \sqsubseteq C_2$.*

First, we only have to consider sets of concepts that are equivalence-reduced against $\mathcal{T}^{in}_{[C]}$, since all remaining equivalences already follow from $\mathcal{T}^{in}_{[C]}$. Second, we exclude all concepts from $[C]$ whose equivalence can be expressed by a single concept inclusion axiom.

Example 4. Let $\mathcal{T}_2$ extend $\mathcal{T}_1$ with the following axioms:

$$B_2 \sqcap B \sqsubseteq \exists r.\exists s.A_1 \qquad B_3 \equiv B_4 \sqcap A_3 \equiv B_4 \sqcap \exists t.A_2 \equiv B_4 \sqcap \exists s.A_1$$

We have $\mathcal{S}(B_2 \sqcap B)_{\mathcal{T}_2} = \{\exists r.\exists s.A_1\}$ and $\mathcal{S}(B_3)_{\mathcal{T}_2} = [B_3]_{\mathcal{T}_2} = \{B_3, B_4 \sqcap A_3, B_4 \sqcap \exists t.A_2, B_4 \sqcap \exists s.A_1\}$. We determine a minimal set of required equivalent concepts in $\mathcal{T}_{2[B_3]}$. We have $[B \sqcap \exists r.\exists s.A_1] \rightsquigarrow_\mathcal{T} [B_2 \sqcap B]$ and $[\exists t.A_2] \rightsquigarrow_\mathcal{T} [B_3]$, and therefore $\mathcal{T}^{in}_{2[B_3]} = \mathcal{T}_1 \cup \{B_2 \sqsubseteq \exists r.\exists s.A_1\}$. The minimal set of required equivalent concepts in $\mathcal{T}_{2[B_3]}$ is $[B_3]^{\equiv}_{\mathcal{T}_2} = \{B_3, B_4 \sqcap \exists s.A_1\}$. $B_4 \sqcap A_3$ and $B_4 \sqcap \exists t.A_2$ are not included in this set, because $\mathcal{T}^{in}_{2[B_2]} \models B_4 \sqcap A_3 \sqsubseteq B_4 \sqcap \exists s.A_1$ and $\mathcal{T}^{in}_{2[B_2]} \models B_4 \sqcap \exists t.A_2 \sqsubseteq B_4 \sqcap \exists s.A_1$.

In order to determine a minimised partition for $\mathcal{T}_{[C]}$, we have to distinguish cases based on $\mathcal{S}(C)_\mathcal{T}$ and any minimal set of required equivalent concepts in $\mathcal{T}_{[C]}$. We first give the definition of minimised partitions, and then explain it in detail.

Definition 4. *Let $\mathcal{T}$ be an acyclic $\mathcal{EL}$ ontology and C an $\mathcal{EL}$ concept. Then, $\mathcal{T}^m_{[C]} = \{\alpha_C\}$ is a* minimised partition for C in $\mathcal{T}$ *iff α_C is as follows, where $[C]^{\equiv} = \{C_1, \ldots, C_n\}$ is a set of minimally required concepts of C in $\mathcal{T}$ and $C_s = \bigsqcap \mathcal{S}(C)_\mathcal{T}$:*

1. *If $|[C]^{\equiv}| \leq 1$ and $\mathcal{T}^{in}_{[C]} \models C \sqsubseteq C_s$:*

   – $\alpha_C = \emptyset$
2. *If* $|[C]^{\equiv}| \leq 1$ *and* $\mathcal{T}^{in}_{[C]} \not\models C \sqsubseteq C_s$*:*
   – $\alpha = C^m \sqsubseteq C^m_s$*, where* $C^m \in min_c(C, \mathcal{T}^{in}_{[C]})$ *and* $C^m_s \in min_c(C_s, \mathcal{T}^{in}_{[C]}, C^m)$*.*
3. *If* $|[C]^{\equiv}| > 1$ *and* $\mathcal{T}^{in}_{[C]} \cup \{C_1 \equiv \ldots \equiv C_n\} \models C_1 \sqsubseteq C_s$*:*
   – $\alpha_C = C^m_1 \equiv \ldots \equiv C^m_n$*, where* $C^m_i \in min_c(C_i, \mathcal{T}^{in}_{[C]})$
4. *If* $|[C]^{\equiv}| > 1$ *and* $\mathcal{T}^{in}_{[C]} \cup \{C_1 \equiv \ldots \equiv C_n\} \not\models C_1 \sqsubseteq C_s$*:*
   – $\alpha_C = C^m_1 \equiv \ldots \equiv C^m_n \equiv C^m_s$*, where* $C^m_i \in min_c(C_i, \mathcal{T}^{in}_{[C]})$ *and* $C^m_s \in min_c(C_1 \sqcap C_s, \mathcal{T}^{in}_{[C]} \cup \{C_1 \equiv \ldots \equiv C_n\})$*.*

The minimised partition only contains an equivalence axiom if there is more than one required equivalent concept (Condition 3 and 4). Otherwise, whether we need a concept inclusion axiom depends on whether all concept inclusions already follow from $\mathcal{T}^{in}_{[C]}$ or not (Condition 1 and 2). Note that $C_s = \bigsqcap \mathcal{S}(C)_{\mathcal{T}}$ contains all concept inclusion information for C. A minimal concept inclusion axiom is determined as discussed in Section 3. Assume we have more than one required equivalent concept and $C \sqsubseteq C_s$ does not follow solely from $\mathcal{T}^{in}_{[C]}$ and the required equivalences (Condition 4). In this case, we might just add a concept inclusion axiom as for Condition 2. However, this way we might miss the minimal solution. Observe that the axioms $C_1 \equiv C_2, C_1 \sqsubseteq C_s$ are equivalent to $C_1 \equiv C_2 \equiv C_1 \sqcap C_s$. $C_1 \sqcap C_s$ has the same size as $C_1 \sqsubseteq C_s$, but it can be equivalent to a concept that is smaller to any concept inclusion axiom for C. For simplicity, we therefore always encode concept inclusions into the equivalence axiom if Condition 4 of the definition is fulfilled.

Theorem 2. *Let* $\mathcal{T}$ *be an acyclic* $\mathcal{EL}$ *ontology,* $\mathcal{T}^m$ *a minimal equivalent ontology,* C *an* $\mathcal{EL}$ *concept and* $\mathcal{T}^m_{[C]}$ *be a minimised partition for* C *in* $\mathcal{T}$*. Denote by* $\mathcal{T}^{m2}$ *the result of replacing* $\mathcal{T}_{[C]}$ *in* $\mathcal{T}^m$ *by* $\mathcal{T}^m_{[C]}$*. Then,* $\mathcal{T} \equiv \mathcal{T}^{m2}$ *and* $\int(\mathcal{T}^m) = \int(\mathcal{T}^{m2})$*.*

The result of the method `minimise`($\mathcal{T}_{[C]}, \mathcal{T}^m$) used in Algorithm 1 is calculated by checking the cases in Definition 4. Together with Theorem 2, we can establish the correctness of our method.

Theorem 3. *For any acyclic* $\mathcal{EL}$ *ontology* $\mathcal{T}$*, Algorithm 1 terminates and returns a minimal ontology* $\mathcal{T}^m$ *such that* $\mathcal{T} \equiv \mathcal{T}^m$*.*

Example 5. We continue on the running example. As it turns out, $(\mathcal{T}_2)^{in}_{[B_2 \sqcap B]} = \mathcal{T}_1$ is already minimal. To minimise $\mathcal{T}_{[B_2 \sqcap B]}$, we note that Case 2 applies, since $[B_2 \sqcap B]_{\mathcal{T}}$ contains only one element. Based on the minimisation result in Example 1, we obtain the minimised partition $\{B_2 \sqcap B \sqsubseteq \exists r.A_3\}$. For $[B_3]_{\mathcal{T}}$, the minimal set of required equivalent concepts is $\{B_3, B_4 \sqcap \exists s.A_1\}$. Case 4 applies, which means we have to encode remaining concept inclusions from $\mathcal{S}(B_3)_{\mathcal{T}_2}$ into the equivalence axiom. The resulting minimised partition is $\{B_3 \equiv B_4 \sqcap \exists s.A_1 \equiv B_4 \sqcap \exists t.A_2\}$. As a result, we obtain the following minimal ontology:

$$B \sqcap \exists r.\exists s.A_1 \sqsubseteq \exists r.\exists t.A_2 \qquad \exists t.A_2 \sqsubseteq A_3 \qquad A_3 \sqsubseteq \exists s.A_1$$
$$B_2 \sqcap B \sqsubseteq \exists r.A_3 \qquad B_3 \equiv B_4 \sqcap \exists s.A_1 \equiv B_4 \sqcap \exists t.A_2$$

5 Evaluation

We implemented the method in Java, using the OWL API [9]. We used the latest version of ELK [11] for reasoning, since it supports incremental reasoning, a feature required for a fast retrieval of subsumption relations for the incrementally built minimised ontologies $\mathcal{T}^{in}_{[C]}$. ELK was further used to verify the equivalence of the minimised ontologies. The implementation is available online.

The computation of minimal equivalent concepts was the most expensive part of the minimisation. Note that for each concept, there are exponentially many rules of type (R3) in the subsumer grammar, which makes an exhaustive search impossible. We therefore used several optimisations. (1) We determine the order in which rules of type (R3) are tried using a best-first strategy, where we evaluate rules based on (a) the conjunction length, (b) the size of the concepts corresponding to the non-terminals in the conjunction, and (c) the size of concepts subsuming these non-terminals. This way, we could reduce the number of entailment tests to 1 or 2 in most cases.Without a strategy like this, we were not able to compute minimal equivalent concepts in almost every case. (2) For large conjunctions, we tested whether certain conjuncts have to be included in every equivalent concept. For small concepts, this test is more expensive than just using a best-first search as described above, but it enabled us to minimise large conjunctions, which is why we only used this optimisation for large concepts.

We evaluated our method on ontologies from the NCBO BioPortal repository [23]. From this repository, we selected all ontologies that (1) could be parsed by the OWL API, (2) contained at least 75% $\mathcal{EL}$ axioms, as defined in the preliminaries of this paper, (3) contained at least one existential role restriction and one conjunction in the $\mathcal{EL}$ axioms. The resulting set contained 55 ontologies. We further included the versions of Galen [25] and NCI [26] from the Tones repository [20] and SNOMED [4]. To get an idea on how our method performs on generated ontologies, we generated 360 uniform interpolants of Galen with a signature size of 50 using the tool Lethe [15]. The syntactical structure of these interpolants was completely determined by the tool. For more information on uniform interpolation, we refer to [16].

Table 5 shows the sizes, the percentage of $\mathcal{EL}$-axioms, and the percentage of equivalence classes in the acyclic part of the input ontologies. The percentage of the size reduced by our method is shown in the column labelled *MSize1*. To compare against simple syntactic transformations, we modified the input ontologies by exhaustively applying the transformation $C_1 \sqsubseteq C_2, C_1 \sqsubseteq C_3 \Rightarrow C_1 \sqsubseteq C_2 \sqcap C_3$. The difference in size against these ontologies is shown in the column labelled *MSize2*. We were especially interested in how the amount of complex concepts changed. We therefore computed the sum of the sizes of complex concepts, as well as the sum of the sizes of existential role restrictions in each ontology. The reductions with respect to these measures are shown in the columns respectively labelled *CSize* and $\exists$*Size*. The running times per ontology are shown in the last column. We see a significant reduction for all measures, especially for complex concepts and existential restrictions, whose accumulative size was reduced by respectively 14.17% and 25.82% on average in the BioPortal repository.

Ontologies	Size	$\mathcal{EL}$	acyclic	MSize1	MSize2	CSize	∃Size	Duration
BioPortal	58,943.8	92.1%	78.5%	12.96%	7.64%	14.17%	25.82%	443 s.
Interpolants	2,192.4	79.8%	98.5%	31.50%	25.37%	48.78%	48.62%	1.3 s.
Galen	13,625	95.72%	90.53%	19.05%	13.41%	21.11%	31.15%	9.4 s.
Gene Ext.	162,950	100.0%	100.0%	31.65%	16.81%	19.68%	50.71%	483 s.
NCI	267,916	94.7%	100.0%	13.45%	3.36%	7.24%	8.71%	1,432 s.
SNOMED	444,473	100.0%	73.7%	23.11%	20.59%	26.04%	27.51%	3,954 s.

Table 1. Evaluation results.

6 Related Work

The problem of minimising $\mathcal{EL}$ concepts was first studied in [2], for the special case of acyclic terminologies. The presented method is based on unfolding, which can easy be implemented for acyclic terminologies. Due to the exponential search space of this method, they also provide a greedy version of the algorithm that has polynomial worst-case complexity, but does not guarantee optimal results. The first technique for simplifying $\mathcal{EL}$ ontologies was presented in [22]. Whereas our method explores the full space of equivalent concepts for minimisation, and determines the exact shape of minimised axioms, this method minimises axioms by replacing subconcepts based on known equivalences. Redundant axioms are removed in a last step. While this method only runs in polynomial time, it only guarantees optimality if the ontology does not use conjunctions. Comparing the results of our evaluations, we see that our method provides a significant improvement: for example, using their method, the size of NCI and Galen are respectively reduced by 6% and 9%, whereas our method provides a reduction by respectively 13% and 19%.

7 Summary

We presented a method for minimising acyclic $\mathcal{EL}$ ontologies, which might provide great help in improving existing ontologies as well as for tools that generate ontology content. Key to our method is to structure the axioms in the ontology into partitions that can be minimised one after another following a implicability relation. Minimal axioms for each partition are computed by analysing inclusion relations between concepts that syntactically occur in the ontology, and making use of a method for minimising $\mathcal{EL}$ concepts with respect to an ontology. $\mathcal{EL}$ concepts are minimised using a technique based on regular tree grammars. An evaluation on realistic and generated ontologies showed that our method reduced the overall size as well as the complexity of ontologies significantly. An open question is how to deal with cyclic ontologies. Whereas our approach could be used to minimise existing partitions in cyclic ontologies, the main challenge for cyclic ontologies is that we have to determine partitions not present in the original ontology. Apart from cyclic $\mathcal{EL}$ ontologies, we are currently investigating methods for minimising concepts in more expressive description logics.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press (2003)
2. Baader, F., Küsters, R., Molitor, R.: Rewriting concepts using terminologies. In: Proceedings of KR 2000. pp. 297–308 (2000)
3. Carral, D., Feier, C., Grau, B.C., Hitzler, P., Horrocks, I.: $\mathcal{EL}$-ifying ontologies. In: Proceedings of IJCAR 2014 (2014)
4. Cornet, R., de Keizer, N.: Forty years of SNOMED: a literature review. BMC Med. Inf. & Decision Making 8 (2008)
5. Ding, Y., Foo, S.: Ontology research and development. part 1—a review of ontology generation. Journal of information science 28(2), 123–136 (2002)
6. Du, J., Wang, K., Shen, Y.: Towards tractable and practical ABox abduction over inconsistent description logic ontologies. In: Proceedings of AAAI 2015 (2015)
7. Grau, B.C., Kharlamov, E., Zheleznyakov, D.: How to contract ontologies. In: Proceedings of OWLED 2012 (2012)
8. Grimm, S., Wissmann, J.: Elimination of redundancy in ontologies. In: Proceedings of ESWC 2011. pp. 260–274 (2011)
9. Horridge, M., Bechhofer, S.: The OWL API: A Java API for OWL ontologies. IOS Press (2011)
10. Horrocks, I., Patel-Schneider, P., van Harmelen, F.: From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. Web Semantics: Science, Services and Agents on the World Wide Web 1(1), pp.7–26 (2003)
11. Kazakov, Y., Klinov, P.: Incremental reasoning in OWL EL without bookkeeping. In: Proceedings of ISWC 2013, pp. 232–247. Springer (2013)
12. Klarman, S., Endriss, U., Schlobach, S.: ABox abduction in the description logic $\mathcal{ALC}$. Journal of Automated Reasoning 46(1), 43–80 (2011)
13. Konev, B., Lutz, C., Ozaki, A., Wolter, F.: Exact learning of lightweight description logic ontologies. In: Proceedings of KR 2013. AAAI Press (2013)
14. Koopmann, P., Nikitina, N.: Computing minimal equivalence acyclic $\mathcal{EL}$ ontologies—extended version. Tech. rep., University of Oxford (2016), `http://users.ox.ac.uk/~coml0607/DL2016.pdf`
15. Koopmann, P., Schmidt, R.A.: Lethe: Saturation-based reasoning for non-standard reasoning tasks. In: Proceedings of ORE 2015. pp. 23–30. CEUR-WS.org (2015)
16. Koopmann, P., Schmidt, R.A.: Uniform interpolation and forgetting for $\mathcal{ALC}$ ontologies with ABoxes. In: Proceedings of AAAI 2015 (2015)
17. Lehmann, J., Hitzler, P.: Concept learning in description logics using refinement operators. Machine Learning 78(1-2), 203–250 (2010)
18. Ludwig, M., Konev, B.: Practical uniform interpolation and forgetting for $\mathcal{ALC}$ TBoxes with applications to logical difference. In: Proceedings of KR 2014. AAAI Press (2014)
19. Lutz, C., Piro, R., Wolter, F.: Description logic TBoxes: Model-theoretic characterizations and rewritability. In: Proceedings of IJCAI 2011. pp. 983–988 (2011)
20. Matentzoglu, N., Bail, S., Parsia, B.: A corpus of OWL DL ontologies. In: Proceedings of DL 2013. vol. 1014, pp. 829–841 (2013)
21. Nikitina, N., Rudolph, S.: (Non-)succinctness of uniform interpolants of general terminologies in the description logic $\mathcal{EL}$. Artificial Intelligence 215(0), 120–140 (2014)

22. Nikitina, N., Schewe, S.: Simplifying description logic ontologies. In: Proceedings of ISWC 2013. LNCS, vol. 8219, pp. 411–426. Springer (2013)
23. Noy, N.F., Shah, N.H., Whetzel, P.L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D.L., Storey, M.A., Chute, C.G., Musen, M.A.: BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research (2009)
24. Qi, G., Liu, W., Bell, D.A.: Knowledge base revision in description logics. In: Logics in Artificial Intelligence, pp. 386–398. Springer (2006)
25. Rector, A., Gangemi, A., Galeazzi, E., Glowinski, A., Rossi-Mori, A.: The GALEN CORE model schemata for anatomy: Towards a re-usable application-independent model of medical concepts. In: Proceedings of MIE 2094. pp. 229–233 (1994)
26. Sioutos, N., de Coronado, S., Haber, M.W., Hartel, F.W., Shaiu, W.L., Wright, L.W.: NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. Journal of Biomedical Informatics 40(1), 30–43 (2007)
27. Wächter, T., Schroeder, M.: Semi-automated ontology generation within OBO-Edit. Bioinformatics 26(12), 88–96 (2010)