Simplifying Description Logic Ontologies

Nadeschda Nikitina[1] and Sven Schewe[2]

[1] University of Oxford, UK
[2] University of Liverpool, UK

Abstract. We discuss the problem of minimizing TBoxes expressed in the lightweight description logic $\mathcal{EL}$, which forms a basis of some large ontologies like SNOMED, Gene Ontology, NCI and Galen. We show that the minimization of TBoxes is intractable (NP-complete). While this looks like a bad news result, we also provide a heuristic technique for minimizing TBoxes. We prove the correctness of the heuristics and show that it provides optimal results for a class of ontologies, which we define through an acyclicity constraint over a reference relation between equivalence classes of concepts. To establish the feasibility of our approach, we have implemented the algorithm and evaluated its effectiveness on a small suite of benchmarks.

1 Introduction

It is well-known that the same facts can be represented in many different ways, and that the size of these representations can vary significantly. This is also reflected in ontology engineering, where the syntactic form of ontologies can be more complex than necessary. For instance, throughout the development (and the life-cycle) of an ontology, the way in which concepts and the relationship between them are represented within the ontology are constantly changing. For example, a name for a complex concept expression is often introduced only after it has been used several times and has proved to be important. Another example are dependencies between concepts that evolve over time, resulting in new subsumption relations between concepts ($A_1 \sqsubseteq A_2$). As a result, previously reasonable concept expressions may become unnecessarily complex. In the given example, $A_1 \sqcap A_2$ becomes equivalent to A_1.

Clearly, unnecessary complexity impacts on the maintenance effort as well as the usability of ontologies. For instance, keeping track of dependencies between complex concept expressions and relationships between them is more cumbersome when it contains unnecessarily complex or unnecessarily many different concept expressions. As a result, the chance of introducing unwanted consequences is higher. Moreover, unintended redundancy decreases the overall quality of the ontology.

Removing unnecessary syntactic complexity from ontologies by hand is a difficult task: for the average ontology, it is almost impossible to obtain the minimal representation without tool support. Thus, automated methods that help to assess the current succinctness of an ontology and generate suggestions on how to increase it would be highly valued by ontology engineers.

It is easy to envision scenarios that demonstrate the usefulness of rewriting for reducing the cognitive complexity of axioms. For instance, when a complex concept C is

H. Alani et al. (Eds.): ISWC 2013, Part I, LNCS 8218, pp. 411–426, 2013.

frequently used in the axioms of an ontology and there is an equivalent atomic concept A_C, the ontology will diminish in size when occurrences of C are replaced by A_C.

Example 1. Consider the following excerpt from the ontology Galen [1]:

$$\begin{aligned}\mathtt{Clotting} \sqsubseteq\ & \exists\,\mathtt{actsSpecificallyOn}.(\mathtt{Blood} \sqcap \exists\,\mathtt{hasPhysicalState}. \\ & \qquad (\mathtt{PhysicalState} \sqcap \exists\,\mathtt{hasState.liquid})) \sqcap \\ & \exists\mathtt{hasOutcome}.(\mathtt{Blood} \sqcap \exists\,\mathtt{hasPhysicalState.solidState}) \qquad (1)\end{aligned}$$

$$\mathtt{LiquidState} \equiv \mathtt{PhysicalState} \sqcap \exists\,\mathtt{hasState.liquid} \qquad (2)$$

$$\mathtt{LiquidBlood} \equiv \mathtt{Blood} \sqcap \exists\,\mathtt{hasPhysicalState.LiquidState} \qquad (3)$$

Given concepts defined in Axioms 2 and 3 above, we can easily rewrite Axiom 1 to obtain the following, simpler axiom containing only 6 references to concepts and roles (as opposed to 10 references in Axiom 1):

$$\begin{aligned}\mathtt{Clotting} \sqsubseteq\ & \exists\,\mathtt{actsSpecificallyOn.LiquidBlood} \sqcap \\ & \exists\mathtt{hasOutcome}.(\mathtt{Blood} \sqcap \exists\,\mathtt{hasPhysicalState.solidState}) \qquad (4)\end{aligned}$$

In description logics [2], few results towards simplifying ontologies have been obtained so far. Grimm et al. [3] propose an algorithm for eliminating semantically redundant axioms from ontologies. In the above approach, axioms are considered as atoms that cannot be split into parts or changed in any other way. With the specific goal of improving reasoning efficiency, Bienvenu et al. [4] propose a normal form called prime implicates normal form for $\mathcal{ALC}$ ontologies. However, as a side-effect of this transformation, a doubly-exponential blowup in concept size can occur.

In this paper, we investigate the succinctness for the lightweight description logic $\mathcal{EL}$. The tractable OWL 2 EL profile [5] of the W3C-specified OWL Web Ontology Language [6] is based on DLs of the $\mathcal{EL}$ family [7]. We consider the problem of finding a minimal equivalent representation for a given $\mathcal{EL}$ ontology. First, we demonstrate that we can reduce the size of a representation by up to an exponent even in the case that the ontology does not contain any redundant axioms. We show that the related decision problem (is there an equivalent ontology of size $\leq k$?) is NP-complete by a reduction from the set cover problem, which is one of the standard NP-complete problems. We also show that, just as for other reasoning problems in $\mathcal{EL}$, ontology minimization becomes simpler under the absence of a particular type of cycles. We identify a class of TBoxes, for which the problem can be solved in PTIME instead of NP and implement a tractable algorithm that computes a minimal TBox for this class of TBoxes. The algorithm can also be applied to more expressive and most cyclic TBoxes[1], however without a guarantee of minimality. We apply an implementation of the algorithm to various existing ontologies and show that their succinctness can be improved. For instance, in case of Galen, we managed to reduce the number of complex concepts occurrences by 955 and the number of references to atomic concepts and roles by 1130.

The paper is organized as follows: In Section 2, we recall the necessary preliminaries on description logics. Section 3 demonstrates the potential of minimization. In the

[1] The extension to general TBoxes is a trivial modification of the algorithm.

same section, we also introduce the basic definitions of the size of ontologies and formally state the corresponding decision problem. In Section 4, we derive the complexity bounds for this decision problem. Section 5 defines the class of TBoxes, for which the problem can be solved in PTIME instead of NP and presents a tractable algorithm that computes a minimal TBox for this class of TBoxes. In Section 6, we present experimental results for a selection of ontologies. Finally, we discus related approaches in Section 7 before we conclude and outline future work in Section 8. Further details and proofs can be found in the extended version of this paper.

2 Preliminaries

We recall the basic notions in description logics [2] required in this paper. 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, E 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 $D \sqsubseteq C$. The signature of an $\mathcal{EL}$ concept C or an axiom α, denoted by sig(C) or sig(α), 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 $\text{sig}_C(C)$ and $\text{sig}_R(C)$, respectively. The signature of a TBox $\mathcal{T}$, in symbols sig$(\mathcal{T})$ (correspondingly, $\text{sig}_C(\mathcal{T})$ and $\text{sig}_R(\mathcal{T})$), is defined analogously. Additionally, we denote the set of subconcepts occurring in a concept C as $\mathtt{sub}(C)$ and the set of all subconcepts including part-conjunctions as $\mathtt{sub}_\sqcap(C)$. For instance, for $C = \exists r.(A_1 \sqcap A_2 \sqcap A_3)$ we obtain $\mathtt{sub}(C) = \{\exists r.(A_1 \sqcap A_2 \sqcap A_3), A_1 \sqcap A_2 \sqcap A_3, A_1, A_2, A_3\}$ and $\mathtt{sub}_\sqcap(C) = \{\exists r.(A_1 \sqcap A_2 \sqcap A_3), A_1 \sqcap A_2 \sqcap A_3, A_1 \sqcap A_2, A_1 \sqcap A_3, A_2 \sqcap A_3, A_1, A_2, A_3\}$. Accordingly, we denote the set of subconcepts occurring in a TBox $\mathcal{T}$ as $\mathtt{sub}(\mathcal{T})$ and the set of all subconcepts including part-conjunctions as $\mathtt{sub}_\sqcap(\mathcal{T})$.

Next, we recall the semantics of the above introduced DL constructs, which is defined by 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 $\mathcal{T}$. A TBox $\mathcal{T}$ entails another TBox $\mathcal{T}'$, in symbols $\mathcal{T} \models \mathcal{T}'$, if $\mathcal{T} \models \alpha$ for all $\alpha \in \mathcal{T}'$. $\mathcal{T} \equiv \mathcal{T}'$ is a shortcut for $\mathcal{T} \models \mathcal{T}'$ and $\mathcal{T}' \models \mathcal{T}$.

3 Reducing the Complexity of Ontologies

The size of a TBox $\mathcal{T}$ is often measured by the number of axioms contained in it ($|\mathcal{T}|$). This is, however, a simplified view of the size, which neither reflects cognitive complexity, nor the reasoning complexity. In this paper, we measure the size of a concept, an axiom, or a TBox by the number of references to signature elements as stated in the definition below.

Definition 1. *The* size *of an $\mathcal{EL}$ concept D is defined as follows:*

- *for $D \in sig(\mathcal{T}) \cup \{\top\}$, $\int(D) = 1$;*
- *for $D = \exists r.C$, $\int(D) = \int(C) + 1$ where $r \in sig_R(\mathcal{T})$ and C is an arbitrary concept;*
- *for $D = C_1 \sqcap C_2$, $\int(D) = \int(C_1) + \int(C_2)$ where C_1, C_2 are arbitrary concepts;*

The size *of an $\mathcal{EL}$ axiom (one of $C_1 \sqsubseteq C_2$, $C_1 \equiv C_2$) and a TBox $\mathcal{T}$ is accordingly defined as follows:*

- $\int(C_1 \sqsubseteq C_2) = \int(C_1) + \int(C_2)$ *for concepts C_1, C_2;*
- $\int(C_1 \equiv C_2) = \int(C_1) + \int(C_2)$ *for concepts C_1, C_2.*
- $\int(\mathcal{T}) = \sum_{\alpha \in \mathcal{T}} \int(\alpha)$ *for a TBox $\mathcal{T}$.*

The above definition, for instance, can serve as a basis for computing the average size of axioms ($\int(\mathcal{T}) \div |\mathcal{T}|$) within an ontology. In addition to the above measure of size, the number of distinct complex concept expressions $\mathtt{sub}(\mathcal{T})$ and the overall number of occurrences of such concept expressions (with the corresponding values related to $|\mathcal{T}|$) can serve as an indication of how complex are concept expressions within the ontology. In the following example, we demonstrate the difference between the two measures $|\mathcal{T}|$ and $\int(\mathcal{T})$ and show how the complexity of an ontology can be reduced in principle (by up to an exponent for ontologies without redundant axioms, i.e., axioms that can be omitted without losing any logical consequences).

Example 2. Let concepts C_i be inductively defined by $C_0 = A$, $C_{i+1} = \exists r.C_i \sqcap \exists s.C_i$. Intuitively, C_i of concepts have the shape of binary trees with exponentially many leaves. Clearly, the concepts grow exponentially with i, since $\int(C_i) = 2 + 2 \cdot \int(C_{i-1})$. For a natural number n, consider the TBox $\mathcal{T}_n$:

$$\begin{aligned} C_{n-1} &\sqsubseteq B \\ B_i &\equiv C_i \qquad 1 \leq i \leq n-1 \end{aligned}$$

While $\mathcal{T}_n$ does not contain any redundant axioms, it can easily be represented in a more compact way by recursively replacing each C_i by the corresponding B_i, yielding $\mathcal{T}_n'$:

$$\begin{aligned} B_{n-1} &\sqsubseteq B \\ B_1 &\equiv C_1 \\ B_{i+1} &\equiv \exists r.B_i \sqcap \exists s.B_i \qquad 1 \leq i \leq n-1 \end{aligned}$$

While the number of axioms is the same in both cases, the complexity of $\mathcal{T}_n$ is clearly lower. E.g., for $n = 5$, we obtain $\int(\mathcal{T}_n) = 134$ and $\int(\mathcal{T}_n') = 24$.

We now consider the problem of finding the minimal equivalent $\mathcal{EL}$ representation for a given TBox. The corresponding decision problem can be formulated as follows:

Definition 2 (P1). *Given an $\mathcal{EL}$ TBox $\mathcal{T}$ and a natural number k, is there an $\mathcal{EL}$ TBox $\mathcal{T}'$ with $\int(\mathcal{T}') \leq k$ such that $\mathcal{T}' \equiv \mathcal{T}$.*

In general, the corresponding minimal result is not unique. We denote the set $\{\mathcal{T}' \mid \mathcal{T}' \equiv \mathcal{T}\}$ by $[\mathcal{T}]$. Note that the minimality of the result is trivially checked by deciding **P1** for a decreasing number k until the answer is negative.

In literature, there are different variations of the ontology minimization problem that cover specific cases. Perhaps the simplest examples for avoidable non-succinctness are axioms that follow from other axioms and that can be removed from the ontology without losing any logical consequences. While some axioms including the last axiom in the above example can be removed in any representations, in general, subsets of axioms can be exchangeable.

Example 3. Consider the ontology $\mathcal{T}$:

$$\begin{array}{cc} C \sqsubseteq \exists r.C & \exists r.D \sqsubseteq D \\ C \sqsubseteq D & \exists r.C \sqsubseteq \exists r.D \end{array}$$

$\mathcal{T}$ has two subset ontologies, $\mathcal{T}_1$ and $\mathcal{T}_2$:

$$\mathcal{T}_1 = \{C \sqsubseteq \exists r.C, \exists r.C \sqsubseteq \exists r.D, \exists r.D \sqsubseteq D\}$$

$$\mathcal{T}_2 = \{C \sqsubseteq \exists r.C, C \sqsubseteq D, \exists r.D \sqsubseteq D\}$$

Neither of the two contains any axioms that are entailed by the remainder of the ontology. There are also no sub-expressions that can be removed. However, $\mathcal{T}_2$ is less complex than $\mathcal{T}_1$, because $C \sqsubseteq D$ is simpler (shorter) than $\exists r.C \sqsubseteq \exists r.D$.

While the above problem is already known to be non-tractable and can have many solutions, the ability to rewrite axioms of the ontology can further increase the difficulty and the number of possible solutions: While in the above cases a minimal ontology contains only subconcepts $\mathtt{sub}(\mathcal{T})$ of the original ontology $\mathcal{T}$, in general, a minimal ontology can introduce new concept expressions as demonstrated in the following example.

Example 4. Consider the following TBox $\mathcal{T}$:

$$\begin{array}{cc} C_1 \sqsubseteq A_2 & A_2 \sqsubseteq C_3 \\ \exists r.D \sqsubseteq D & \exists s.C_1 \sqsubseteq D \end{array}$$

$$\exists s.C_3 \sqsubseteq \exists r.(\exists s.C_1)$$

Assume that $\int(C_1)$ and $\int(C_3)$ are large. Then the axiom $\exists s.C_1 \sqsubseteq D$ needs to be exchanged by $\exists s.A_2 \sqsubseteq D$ to obtain a smaller TBox. The TBox $\mathcal{T}_m$ given below is a minimal representation of $\mathcal{T}$.

$$\begin{array}{cc} C_1 \sqsubseteq A_2 & A_2 \sqsubseteq C_3 \\ \exists r.D \sqsubseteq D & \exists s.A_2 \sqsubseteq D \end{array}$$

$$\exists s.C_3 \sqsubseteq \exists r.(\exists s.C_1)$$

We notice that the original ontology $\mathcal{T}$ does not contain the expression $\exists s.A_2 \in \mathtt{sub}(\mathcal{T}_m)$.

We can conclude that considering subsumption relations between subconcepts $\mathtt{sub}(\mathcal{T})$ of $\mathcal{T}$ is not sufficient when looking for a minimal equivalent representation. In the next section, we show that the corresponding decision problem **P1** is in fact NP-complete.

4 NP-Completeness

In this section, we first show the NP-hardness of the problem and then establish its NP-completeness. We show NP-hardness by a reduction from the set cover problem, which is one of the standard NP-complete problems. For a given set $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ with carrier set $S = \bigcup_{i=1}^{n} S_i$, a *cover* $\mathcal{C} \subseteq \mathcal{S}$ is a subset of $\mathcal{S}$, such that the union of the sets in $\mathcal{C}$ covers S, i.e., $S = \bigcup_{C \in \mathcal{C}} C$.

The *set cover problem* is the problem to determine, for a given set $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ and a given integer k, if there is a cover $\mathcal{C}$ of $\mathcal{S}$ with at most $k \geq |\mathcal{C}|$ elements. We will use a restricted version of the set cover problem, which we call the *dense set cover problem* (DSCP). In the dense set cover problem, we require that

- neither the carrier set S nor the empty set is in $\mathcal{S}$,
- all singleton subsets (sets with exactly one element) of S are in $\mathcal{S}$, and
- if a non-singleton set S is in $\mathcal{S}$, so is some subset $S' \subseteq S$, which contains only one element less than S ($|S \smallsetminus S'| = 1$).

Lemma 1. *The dense set cover problem is NP-complete.*

Proof Sketch. For the full version of the proof, see extended version of the paper. The proof shows how to convert the cover of the non-dense set into a cover of the corresponding dense set and vise versa. □

Given the above NP-completeness result, we show that the size of minimal equivalents specified in **P1** is a linear function of the size of the minimal cover. To this end, we use the lemma below to obtain a lower bound on the size of equivalents. Intuitively, it states that for each entailed non-trivial equivalence $C \equiv A$, the TBox must contain at least one axiom that is at least as large as $C' \equiv A$ for some C' with $\mathcal{T} \models C \equiv C'$:

Lemma 2. *Let $\mathcal{T}$ be an $\mathcal{EL}$ TBox, $A \in sig(\mathcal{T})$ and C, D $\mathcal{EL}$ concepts such that $\mathcal{T} \models C \equiv A$, $\mathcal{T} \models A \sqsubseteq D$ (the latter is required for induction). Then, one of the following is true:*

1. *A is a conjunct of C (including the case $C = A$);*
2. *there exists an $\mathcal{EL}$ concept C' such that $\mathcal{T} \models C \equiv C'$ and $C' \bowtie A \in \mathcal{T}$ or $C' \bowtie A \sqcap D' \in \mathcal{T}$ for some $\bowtie \in \{\equiv, \sqsubseteq\}$ and some concept D'.*

Proof Sketch. For the full version of the proof, see extended version of the paper. We use the sound and complete proof system for general subsumption in $\mathcal{EL}$ terminologies introduced in [8] and prove the lemma by induction on the depth of the derivation of $C \sqsubseteq A \sqcap D$. We assume that the proof has minimal depth and consider the possible rules that could have been applied last to derive $C \sqsubseteq A \sqcap D$. In each case the lemma holds. □

We now show how to encode the dense set cover problem as an ontology minimization problem. Consider an instance of the dense set cover problem with the carrier set $A = \{B_1, \dots, B_n\}$, the set $\mathcal{S} = \{A_1, \dots, A_m, \{B_1\}, \dots, \{B_n\}\}$ of subsets that can be used to form a cover. By interpreting the set and element names as atomic concepts, we can construct $\mathcal{T}_{\mathcal{S}\mathtt{base}}$ as follows:

$$\mathcal{T}_{\mathcal{S}\mathtt{base}} = \{A'' \equiv A' \sqcap B \mid A'', A' \in \mathcal{S}, B \in A, A'' = A' \cup \{B\}, A'' \neq A'\}.$$

Observe that the size of $\mathcal{T}_{\mathcal{S}\mathtt{base}}$ is at least $3m$. Clearly, $\mathcal{T}_{\mathcal{S}\mathtt{base}} \models A_i \equiv \bigsqcap_{B \in A_i} B$. Let $\mathcal{T}_{\mathcal{S}} = \mathcal{T}_{\mathcal{S}\mathtt{base}} \cup \{A \equiv \bigsqcap_{B \in A} B\}$. We establish the connection between the size of $\mathcal{T}_{\mathcal{S}}$ equivalents and the size of the cover of $\mathcal{S}$ as follows:

Lemma 3. *$\mathcal{T}_{\mathcal{S}}$ has an equivalent of size $\int(\mathcal{T}_{\mathcal{S}\mathtt{base}}) + k + 1$ if, and only if, $\mathcal{S}$ has a cover of size k.*

Proof. For the if-direction, assume that $\mathcal{S}$ has a cover of size k. We construct $\mathcal{T}'_{\mathcal{S}}$ of size $\int(\mathcal{T}_{\mathcal{S}\mathtt{base}}) + k + 1$ as follows: $\mathcal{T}'_{\mathcal{S}} = \mathcal{T}_{\mathcal{S}\mathtt{base}} \cup \{A \equiv \bigsqcap_{A' \in \mathcal{C}} A'\}$. Clearly, $\mathcal{T}'_{\mathcal{S}} \equiv \mathcal{T}_{\mathcal{S}}$.

For the only-if-direction, we assume that k is minimal and argue that no equivalent $\mathcal{T}' \in [\mathcal{T}_{\mathcal{S}}]$ of size $\leq \int(\mathcal{T}_{\mathcal{S}\mathtt{base}}) + k$ can exist. Assume that $\mathcal{T}$ is a minimal TBox with $\mathcal{T} \in [\mathcal{T}_{\mathcal{S}}]$. With the observation, that the $m + n$ atomic concepts that represent elements of $\mathcal{S}$ are pairwise not equivalent with each other or the concept A that represents the carrier set, we can conclude that no two atomic concepts are equivalent. From Lemma 2 it follows that, for each A_i with $i \in \{1, \dots, m\}$, there is an axiom $C_i \equiv C'_i \in \mathcal{T}$ or $C_i \sqsubseteq C'_i \in \mathcal{T}$ such that $\mathcal{T} \models C_i \equiv A_i$ and A_i is a conjunct of C'_i or $A_i = C'_i$. Since there are no equivalent atomic concepts and $C_i \neq A_i$ due to the minimality of $\mathcal{T}$, the size of each such axiom is at least 3 and none of these axioms coincide. Additionally, since $\mathcal{T}_{\mathcal{S}} \not\models A_i \sqsubseteq A$, A cannot occur as a conjunct of C_i or as a conjunct of C'_i;

Finally, we estimate the size of the remaining axioms and show that their cumulative size is $> k$. It also follows from Lemma 2 that there exists an axiom $C \equiv C' \in \mathcal{T}$ or $C \sqsubseteq C' \in \mathcal{T}$ such that $\mathcal{T} \models C \equiv A$ and A is a conjunct of C' or $A = C'$. It holds that $\mathcal{T} \models C \equiv \bigsqcap_{B \in A} B$. We also know that for no proper subset $S' \subsetneq A$ holds $\mathcal{T} \models \bigsqcap_{B \in S'} B \sqsubseteq C$. Thus, we have found a cover of S and the size of the axiom must be $\geq k + 1$. Thus, the overall size of $\mathcal{T}$ must be $\geq \int(\mathcal{T}_{\mathcal{S}\mathtt{base}}) + k + 1$. □

Theorem 1. P1 *is in* NP.

Proof. We ask the non-deterministic algorithm to guess a TBox of the size $\leq k$. It remains to verify $\mathcal{T}' \equiv \mathcal{T}$, which can be done in PTIME [7]. □

Theorem 2. P1 *is* NP*-complete.*

Proof. The problem is NP-hard as an immediate consequence of Lemmas 3 and 1. Given the result of Theorem 1, we establish NP-completeness of the problem. □

5 Minimizing Acyclic TBoxes

In this section, we develop an algorithm for minimizing TBoxes in polynomial time, which is guaranteed to provide a minimal TBox for a class of $\mathcal{EL}$ TBoxes satisfying a certain type of acyclicity conditions. The algorithm can also be applied to more expressive and some cyclic TBoxes, however without the guarantee of minimality.

5.1 Acyclicity Conditions

In this subsection, we introduce equivalence classes on concepts and discuss cyclic dependencies between equivalence classes and their impact on computing minimal representations. Let $\mathcal{T}$ be an $\mathcal{EL}$ TBox and let C be a concept in $\mathtt{sub}(\mathcal{T})$. We use the notation $[C]_\mathcal{T} = \{C' \in \mathtt{sub}(\mathcal{T}) \mid \mathcal{T} \models C \equiv C'\}$ to denote the *equivalence class* of the concept C and $\mathcal{C}_\mathcal{T} = \{[C]_\mathcal{T} \mid C \in \mathtt{sub}(\mathcal{T})\}$ to denote the set of all equivalence classes over the set $\mathtt{sub}(\mathcal{T})$. In case $\mathcal{T}$ is clear from the context, we omit the index. We base the acyclicity conditions on the following reference relations, which use both syntactic and semantic dependencies between equivalence classes:

Definition 3. *Let $\mathcal{T}$ be an $\mathcal{EL}$ TBox. The* reference relations $\prec_\sqsubseteq, \prec_\sqsupseteq$ *and $\prec_s$, all subsets of $\mathcal{C} \times \mathcal{C}$, are given as follows:*

- *$[C] \prec_s [C']$ if, for some $C_1 \in [C], C_2 \in [C']$, it holds that C_2 occurs in C_1;*
- *$[C] \prec_\sqsubseteq [C']$ if, for some $C_1 \in [C], C_2 \in [C']$, it holds that $[C_1] \prec_s [C_2]$ or $\mathcal{T} \models C_1 \sqsubseteq C_2$;*
- *$[C] \prec_\sqsupseteq [C']$ if, for some $C_1 \in [C], C_2 \in [C']$, it holds that $[C_1] \prec_s [C_2]$ or $\mathcal{T} \models C_1 \sqsupseteq C_2$.*

We call a TBox *cyclic*, if any of the above relations $\prec_\sqsubseteq, \prec_\sqsupseteq, \prec_s$ is cyclic. We say that a TBox $\mathcal{T}$ is *strongly cyclic* if $\prec_s$ is cyclic. The algorithm presented in this paper is applicable for TBoxes not containing strong cycles. Most of the large bio-medical ontologies including Galen, Gene Ontology and NCI do not contain strong cycles. This was also the case for earlier versions of SNOMED, e.g., the one dated 09 February 2005 [9]. Note that asking for the absence of cycles in $\prec_s$ is a weaker requirement than for $\prec_\sqsubseteq$ or $\prec_\sqsupseteq$, as $\prec_s \subseteq \prec_\sqsubseteq \cap \prec_\sqsupseteq$. But the reverse relationship between the conditions holds.

In some cases, TBoxes contain cycles that are caused by redundant conjuncts and can easily be removed.

Example 5. $\{A \sqcap B \sqsubseteq C, A \sqsubseteq B\}$ has a cyclic $\prec_\sqsupseteq$ relation due to a cycle between $A \sqcap B$ and A. It can be transformed into an acyclic TBox $\{A \sqsubseteq C, A \sqsubseteq B\}$.

We call conjunctions $C' \sqcap C''$ in $\mathtt{sub}(\mathcal{T})$ such that $\mathcal{T} \models C' \sqsubseteq C''$ *subsumer-containing conjunctions*. We can easily eliminate subsumer-containing conjunctions in TBoxes before applying the algorithm: for each subsumer-containing conjunction $C' \sqcap C''$ in $\mathtt{sub}(\mathcal{T})$ with $\mathcal{T} \models C' \sqsubseteq C''$, we replace $C' \sqcap C''$ in $\mathcal{T}$ by C', and add the axiom $C' \sqsubseteq C''$ to $\mathcal{T}$. We can show that the closure of each equivalence class $[C]$ of an acyclic TBox $\mathcal{T}$ is finite if we exclude subsumer-containing conjunctions. We denote such a closure with $[C]^* = \{C' \mid \mathcal{T} \models C \equiv C'$ and C' is not a subsumer-containing conjunction$\}$. We denote the extended set of subconcepts of $\mathcal{T}$ by $\mathtt{sub}(\mathcal{T})^* = \bigcup_{[C] \in \mathcal{C}} [C]^*$.

Another kind of removable cyclic dependencies are conjunctions on the right-hand side. We use a simple *decomposition*, in which all conjunctions on the right-hand side of axioms are replaced by separate inclusion axioms for each conjunct. We obtain the decomposed version $\mathcal{T}'$ of a TBox $\mathcal{T}$ by replacing each $C \sqsubseteq D_1 \sqcap D_2 \in \mathcal{T}_m$ by $C \sqsubseteq D_1, C \sqsubseteq D_2$ until a fixpoint is reached. *Composition* is the dual transformation:

we replace any two axioms $C \sqsubseteq D_1, C \sqsubseteq D_2$ by $C \sqsubseteq D_1 \sqcap D_2$ until a fixpoint is reached.

Unless we state otherwise, in the following we assume that TBoxes are decomposed and do not contain subsumer-containing conjunctions.

5.2 Uniqueness of Minimal TBoxes

Acyclic TBoxes are better behaved not only with respect to the complexity of minimization, but they also have a unique minimal TBox modulo replacement of equivalent concepts by one another (if we assume that the TBox with the lower number of equivalence axioms should be preferred in case of equally large TBoxes).

To be able to determine a unique syntactic representation of a TBox $\mathcal{T}$, we choose a representative $C' \in [C]^*$ for each equivalence class $[C] \in \mathcal{C}$ and denote it using the *representative selection function* $r : \mathcal{C} \to \mathtt{sub}(\mathcal{T})^*$ with $r([C]) = C'$. We say that r is *valid*, if for all $[C], [D] \in \mathcal{C}$ with $[C] \neq [D]$ it holds that $C' \in [C]^*$ occurs in $r([D])$ only if $C' = r([C])$, i.e., representatives can only contain other representatives, but not other elements of equivalence classes.

Definition 4. *Let $\mathcal{T}$ be a TBox and $\bowtie \in \{\equiv, \sqsubseteq\}$. We say that $\mathcal{T}$ is* aligned *with r, if for each $C \bowtie D \in \mathcal{T}$ one of the following conditions holds:*

- *if $\mathcal{T} \not\models C \equiv D$, then $C = r([C])$ and $D = r([D])$;*
- *if $\mathcal{T} \models C \equiv D$, then for each C' such that $C' \neq C$, $C' \neq D$ and C' occurs in C or D it holds that $C' = r([C'])$.*

In other words, the only axioms, in which we allow an occurrence of a non-representative C are axioms relating C with concepts equivalent to it.

Since minimal TBoxes can sometimes contain subsumption axioms relating two equivalent concepts with each other, the otherwise unique TBox result can vary in the choice between subsumption and equivalence axioms. For the sake of uniqueness, we assume that, whenever we have a choice between equivalence ($\equiv$) and subsumption axioms ($\sqsubseteq$) in the resulting TBox, we prefer subsumption axioms.

We call a TBox *non-redundant*, if there is no $\alpha \in \mathcal{T}$ such that $\mathcal{T} \setminus \{\alpha\} \models \alpha$. In order to show how to compute a minimal equivalent TBox for an acyclic initial TBox, we first show that we do not need new equivalence classes or new relations between them to obtain any non-redundant, decomposed, equivalent TBox. In other words, non-redundant, decomposed axioms encoding relations between equivalence classes are unique up to exchanging equivalent concepts.

Lemma 4. *Let $\mathcal{T}_1, \mathcal{T}_2$ be two non-redundant, acyclic $\mathcal{EL}$ TBoxes such that $\mathcal{T}_1 \equiv \mathcal{T}_2$. Let $C \sqsubseteq D \in \mathcal{T}_2$. Then there is $C' \sqsubseteq D' \in \mathcal{T}_1$ such that $\mathcal{T}_1 \models C' \equiv C$, $\mathcal{T}_1 \models D' \equiv D$.*

While the above lemma addresses relations between equivalence classes in non-redundant, decomposed TBoxes, it does not allow us to draw conclusions about axioms representing relations within equivalence classes. The purpose of the below lemma is to determine the part of the TBox that encodes relations between equivalent concepts within equivalence classes. For this, we divide the TBox into partitions: one for non-equivalence axioms $\mathcal{T}^0 = \{C \sqsubseteq D \in \mathcal{T} \mid \mathcal{T} \not\models C \equiv D\}$ and one for axioms encoding relations within each equivalence class: $\mathcal{T}^{[C']} = \{C \equiv D \in \mathcal{T} \mid C, D \in [C']\}$ for each $[C'] \in \mathcal{C}$. We denote the set of all subsumption dependencies holding within a partition by $\mathcal{T}^{\mathtt{full},[C']} = \{C \sqsubseteq D \mid C, D \in [C']\}$. In each (equivalence class) partition, a part of dependencies can be deducible from the remainder of the TBox.

Example 6. Consider the TBox $\mathcal{T} = \{A \sqsubseteq B, \exists r.A \equiv \exists r.B\}$. For the equivalence class $\{\exists r.A, \exists r.B\}$, the subsumption $\exists r.A \sqsubseteq \exists r.B$ follows from $A \sqsubseteq B$.

We denote entailed dependencies for an equivalence class $[C']$ by $\mathcal{T}^{\mathtt{red},[C']} = \{C \sqsubseteq D \mid C, D \in [C']\}$. We now consider alternative representations of each partition $\mathcal{T}^{[C']}$. We first show that, in any acyclic TBox $\mathcal{T}$ aligned with some valid r, we can determine the entailed dependencies $\mathcal{T}^{\mathtt{red},[C']}$ within each $\mathcal{T}^{\mathtt{full},[C']}$ based on $\mathcal{T}^0$.

Lemma 5. *Let $\mathcal{T}$ be a non-redundant, acyclic $\mathcal{EL}$ TBox aligned with a valid representative selection function r. Then, for each non-singleton equivalence class $[C'] \in \mathcal{C}(\mathcal{T})$ and each pair $C, D \in [C']$, it holds that $C \sqsubseteq D \in \mathcal{T}^{\mathtt{red},[C']}$ exactly if one of the following conditions is true:*

1. *$D = \top$*
2. *there are concepts C', D' such that $C = \exists r.C', D = \exists r.D'$ and $\mathcal{T} \models C' \sqsubseteq D'$, $\mathcal{T} \not\models C' \equiv D'$.*

As a consequence, each equivalence class partition can be considered independently from other equivalence class partitions. In particular, this implies that, for any syntactic representation $\mathcal{T}^{[C]}$ of a partition for equivalence class $[C']$, we can obtain $\mathcal{T}^{\mathtt{full},[C']}$ from $\mathcal{T}^{[C]} \cup \mathcal{T}^{\mathtt{red},[C]}$ by computing its transitive closure [2].

Lemma 6. *Let $\mathcal{T}$ be a non-redundant, acyclic $\mathcal{EL}$ TBox aligned with a valid representative selection function r. Then, for each equivalence class $[C] \in \mathcal{C}(\mathcal{T})$ it holds that $(\mathcal{T}^{[C]} \cup \mathcal{T}^{\mathtt{red},[C]})^* = \mathcal{T}^{\mathtt{full},[C']}$.*

Since our implementation operates on ontologies represented in the OWL Web Ontology Language, we consider here an important detail of this language. In addition to constructs mentioned in preliminaries, OWL Web Ontology Language allows for *OwlEquivalentClassesAxioms* - axioms, in which we can specify a set of equivalent concepts. With the exception of equivalence classes containing $\top$, for which there exists an equally small representation without an OwlEquivalentClassesAxiom, this is clearly the smallest representation for equivalence class partitions.

Let $[C]^{\mathtt{nonred}} = [C] \setminus \{C' \in [C] \mid C' \sqsubseteq D'_1$ and $C' \sqsupseteq D'_2 \in \mathcal{T}^{\mathtt{red},[C]}$ for some $D'_1, D'_2\}$. Let $\mathcal{T}^{\mathtt{nonred},[C]}$ be the corresponding OWLEquivalentClassesAxiom with $[C]^{\mathtt{nonred}}$ as the set of equivalent concepts. Note that, according to the semantics of OwlEquivalentClassesAxioms, it holds that $\mathcal{T}^{\mathtt{nonred},[C]} \models \mathcal{T}^{\mathtt{full},[C]^{\mathtt{nonred}}}$. Thus, $\mathcal{T}^{\mathtt{nonred},[C]} \cup \mathcal{T}^{\mathtt{red},[C]} \models \mathcal{T}^{\mathtt{full},[C]}$. Note that $\int(\mathcal{T}^{\mathtt{nonred},[C]}) = \sum_{C' \in [C]^{\mathtt{nonred}}} \int(C')$.

Lemma 7. *Let $\mathcal{T}$ be a non-redundant, acyclic $\mathcal{EL}$ TBox aligned with a valid representative selection function r. Then, $\int(\mathcal{T}^{\mathtt{nonred},[C]}) \leq \int(\mathcal{T}^{[C]})$ for each equivalence class $[C] \in \mathcal{C}(\mathcal{T})$.*

[2] For a set $\mathcal{T}$ of axioms, the transitive closure $(\mathcal{T})^*$ is obtained by including $C \sqsubseteq D$ for any C, D such that there exists C' with $\mathcal{T} \models \{C \sqsubseteq C', C' \sqsubseteq D\}$.

Algorithm 1. Rewriting $\mathcal{T}_{\mathtt{in}}$

```
   Data: $\mathcal{T}_{\mathtt{in}}$ acyclic decomposed TBox
   Result: $\mathcal{T}_{\mathtt{out}}$ minimal equivalent TBox
1  $\mathcal{C}_{\mathtt{all}} \leftarrow \mathcal{C}$;
2  $\mathcal{C}_{\mathtt{TODO}} \leftarrow \mathcal{C}_{\mathtt{all}}$;
3  $\mathcal{T}_{\mathtt{out}} \leftarrow$ remove equivalence axioms from $\mathcal{T}_{\mathtt{in}}$;
4  while $\mathcal{C}_{\mathtt{TODO}} \neq \emptyset$ do
5      for $[C] \in \mathit{leaves}(\mathcal{C}_{\mathtt{TODO}})$ do
6          choose minimal representative $r([C])$;
7          replace $C' \in [C]$ in $\mathcal{T}_{\mathtt{out}}$ by $r([C])$;
8          replace $C' \in [C]$ in $\mathcal{C}_{\mathtt{TODO}} \setminus \{[C]\}$ by $r([C])$;
9          replace $C' \in [C]$ in $\mathcal{C}_{\mathtt{all}} \setminus \{[C]\}$ by $r([C])$;
10         $\mathcal{C}_{\mathtt{TODO}} \leftarrow \mathcal{C}_{\mathtt{TODO}} \setminus \{[C]\}$;
11 $\mathcal{T}_e \leftarrow \bigcup_{[C]\in\mathcal{C}_{\mathtt{all}},|[C]|\geq 2} \mathcal{T}^{\mathtt{nonred},[C]}$;
12 for $\alpha \in \mathcal{T}_{\mathtt{out}}$ do
13     if $\mathcal{T}_{\mathtt{out}} \cup \mathcal{T}_e \setminus \{\alpha\} \models \alpha$ then
14         $\mathcal{T}_{\mathtt{out}} \leftarrow \mathcal{T}_{\mathtt{out}} \setminus \{\alpha\}$;
15 $\mathcal{T}_{\mathtt{out}} \leftarrow \mathcal{T}_{\mathtt{out}} \cup \mathcal{T}_e$ ;
16 $\mathcal{T}_{\mathtt{out}} \leftarrow$ compose($\mathcal{T}_{\mathtt{out}}$);
```

Based on the above two lemmas, we can show that, in the acyclic case, we can compute a minimal TBox by eliminating redundant axioms, fixing the representative selection function r to some minimal value, constructing the core representation $\mathcal{T}^{\mathtt{nonred},[C]}$ for each non-singleton equivalence class $[C]$ and composing $\mathcal{T}$ again.

Definition 5. *Let $\mathcal{T}$ be an $\mathcal{EL}$ TBox and r a corresponding valid representative selection function. We say that r is* minimal, *if for each $[C] \in \mathcal{C}$ holds: there is no $C \in [C]^*$ such that $\int(C) < \int(r([C]))$.*

We can now state the minimality of the composed TBox containing $\mathcal{T}^0$ and a partition $\mathcal{T}^{\mathtt{nonred},[C]}$ for each non-singleton equivalence class $[C] \in \mathcal{C}$.

Theorem 3. *Let $\mathcal{T}$ be a non-redundant, acyclic $\mathcal{EL}$ TBox and r a minimal, valid representative selection function. Let the TBox $\mathcal{T}_n = \mathcal{T}^0 \cup \bigcup_{[C]\in\mathcal{C},|[C]|\geq 2} \mathcal{T}^{\mathtt{nonred},[C]}$ be aligned with r. Let $\mathcal{T}_n'$ be a composed version of $\mathcal{T}_n$. Then, for any minimal TBox $\mathcal{T}_m$ with $\mathcal{T}_m \equiv \mathcal{T}$ it holds that $\int(\mathcal{T}_m) = \int(\mathcal{T}_n')$.*

Algorithm 1 implements the iterative computation of r and the minimal TBox $\mathcal{T}_n'$. It takes an acyclic decomposed TBox $\mathcal{T}_{\mathtt{in}}$ and computes the corresponding minimal equivalent TBox $\mathcal{T}_{\mathtt{out}}$. Line 3 is not strictly necessary, but allows for a more efficient processing. In Lines 4-10, a minimal representative selection function r is iteratively determined – for one equivalence class at a time – and all data structures are aligned with r. We distinguish two versions of equivalence classes: $\mathcal{C}_{\mathtt{TODO}}$ contains equivalence classes, for which the minimal representative has not been selected yet. In each iteration, we process the leaves in $\mathcal{C}_{\mathtt{TODO}}$ ordered with the reference relation $\prec_s$ and remove those equivalence classes from $\mathcal{C}_{\texttt{TODO}}$. $\mathcal{C}_{\texttt{all}}$ contains all equivalence classes that are stepwise aligned with a minimal representative selection function r. In each step, we also align axioms $\mathcal{T}_{\texttt{out}}$ corresponding to the partition $\mathcal{T}^0$ with r by replacing concepts with the representative $r([C])$ fixed in Line 6.

In Line 11, we build partitions for non-singleton equivalence classes. In Lines 12-14, we compute the non-redundant part of $\mathcal{T}_{\texttt{out}}$. The function $\texttt{compose}(\mathcal{T}_{\texttt{out}})$ in Line 16 composes subsumption axioms with identical left-hand sides into a single axiom.

Clearly, Algorithm 1 runs in PTIME. $\texttt{sub}(\mathcal{T})$ is polynomially large in the size of $\mathcal{T}$ and $\mathcal{C}$ can be computed in PTIME due to tractable reasoning in $\mathcal{EL}$. Equivalence axioms can be removed in linear time. Lines 4-10 are executed $|\mathcal{C}|$ times and can be executed in PTIME. The same holds for building partitions for non-singleton equivalence classes and computing the non-redundant part of $\mathcal{T}_{\texttt{out}}$. Composition can be performed in linear time. Note that the algorithm remains tractable only assuming the tractability of reasoning in the underlying logic. Otherwise, the complexity of reasoning dominates. In principle, the result could be obtained after computing the representatives for each equivalence class by simply selecting all subsumption relations between classes. However, this would result in a less efficient implementation with large intermediary results.

Theorem 4. *Let $\mathcal{T}$ be an acyclic $\mathcal{EL}$ TBox. Algorithm 1 computes a minimal equivalent TBox in* PTIME.

Minimality is a consequence of Theorem 3. Equivalence follows from $\mathcal{T}^{\texttt{nonred},[C]} \cup \mathcal{T}^{\texttt{red},[C]} \models \mathcal{T}^{\texttt{full},[C]}$ for each non-singleton equivalence class $[C]$ and from Lemma 4.

6 Experimental Results

For our evaluation, we have implemented the algorithm using the latest version of OWL API and Hermit reasoner. We have used an optimized version of Algorithm 1, where entailment checking is done in two phases, the first of which can be run by several threads.

A selection of publicly available ontologies (as shown in Table 1) that vary in size and expressivity have been used in the experiments[3]. Table 2 shows the number $|\texttt{CON}_o(\mathcal{T})|$ of occurrences of complex concepts $\texttt{CON}(\mathcal{T}) = \texttt{sub}(\mathcal{T}) \setminus \text{sig}_C(\mathcal{T})$ in the first two columns (the original value followed by the new value relative to the original one). The two subsequent columns show the number of pairwise different complex concepts $|\texttt{CON}(\mathcal{T})|$. The last two columns show $\int(\mathcal{T})$ – the size of each ontology measured as the number of occurrences of entities in $\text{sig}(\mathcal{T})$.

The implementation was first applied to Snomed [10]. However, the available fully-fledged reasoners Pellet and Hermit run out of heap space when classifying the ontology even with 10 GB memory assigned to the corresponding Java process. The ELK reasoner [11] is capable of classifying Snomed, but it does not currently implement entailment, which is essential for our implementation.

[3] The wine ontology can be retrieved from `http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine`. All other ontologies used can be found in the TONES ontology repository at `http://owl.cs.manchester.ac.uk/repository`

Table 1. Properties of ontologies used in experiments

	$\lvert\mathcal{T}\rvert$	$\int(\mathcal{T})/\lvert\mathcal{T}\rvert$	$\mathtt{CON}(\mathcal{T})/\lvert\mathcal{T}\rvert$	$\mathtt{CON}_o(\mathcal{T})/\lvert\mathcal{T}\rvert$	Logic
Snomed	83,259	4.99	1.14	2.57	$\mathcal{EL}++$
Gene Ontology	42656	3.37	1.20	0.27	$\mathcal{EL}++$
NCI	97811	1.10	0.00	0.14	$\mathcal{ALCH}(\mathcal{D})$
Galen	4735	2.81	0.52	1.13	$\mathcal{ALEHIF}+$
Adult Mouse	3464	2.48	0.15	0.48	$\mathcal{EL}++$
Wine	657	1.03	0.21	0.40	$\mathcal{SHOIN}(\mathcal{D})$
Nautilus	38	2.18	0.29	0.40	$\mathcal{ALCHF}(\mathcal{D})$
Cell	1264	2.16	0.09	0.16	$\mathcal{EL}++$
DOLCE-lite	351	1.42	0.13	0.14	$\mathcal{SHIF}$
Software	238	25.21	2.60	7.26	$\mathcal{ALHN}(\mathcal{D})$
Family Tree	36	6.19	1.02	1.33	$\mathcal{SHIN}(\mathcal{D})$
General Ontology	8803	0.48	0.03	0.03	$\mathcal{ALCHOIN}(\mathcal{D})$
Substance	609	2.33	0.22	0.36	$\mathcal{ALCHO}(\mathcal{D})$
Generations	38	1.87	0.58	1.21	$\mathcal{ALCOIF}$
Periodic Table	58	1.38	0.38	0.43	$\mathcal{ALU}$

From the ontologies used in our experiments, only Snomed did not satisfy the acyclicity conditions for $\prec_s$ sufficient to guarantee termination of our algorithm. On the one hand, Snomed contains cyclic concept definitions. For instance, Mast_cell_leukemia is defined by means of the corresponding equivalence axiom as

$$\begin{array}{l}
\texttt{Leukemia\_disease} \ \sqcap \\
\texttt{Mast\_cell\_malignancy} \ \sqcap \\
\exists\,\texttt{RoleGroup}. \\
\quad (\exists\,\texttt{Associated\_morphology}.\texttt{Mast\_cell\_leukemia} \ \sqcap \\
\quad \exists\,\texttt{Finding\_site}.\texttt{Hematopoietic\_system\_structure})) \ \sqcap \\
\exists\,\texttt{RoleGroup}.(\\
\quad \exists\,\texttt{Has\_definitional\_manifestation}.\texttt{White\_blood\_cell\_finding})
\end{array}$$

On the other hand, Snomed contains a cyclic reference relation between the concepts Wound and Wound_finding, which is the only cyclic dependency with more than one element.

We have manually evaluated how the rewriting has affected ontologies. In all cases where concepts became smaller, the improvement has been achieved by either elimination of redundant axioms or exchanging complex expressions by atomic concepts.

In case of the Galen ontology [1], the algorithm managed to reduce the number of occurrences of complex concepts by 955, which is 17%. The size of the ontology in number of references was reduced by 1130 (9%). The number of distinct complex concepts used in the ontology was reduced by 76 (3%). The situation is similar for the NCI [12] ontology.

The other large medical ontology – Gene Ontology [13] – does not contain any equivalent concepts, i.e., each equivalence class has only one element. The current algorithm did not find any possibility to rewrite the ontology. The same holds for Adult Mouse and Periodic Table ontologies.

Table 2. Minimization results for different ontologies

	$\mathtt{CON}_o(\mathcal{T})$		$\lvert\mathtt{CON}(\mathcal{T})\rvert$		$\int(\mathcal{T})$	
Snomed	213,856	–	95,315	–	415,541	–
Gene Ontology	11,686	1	8,508	1	143,900	1
NCI	13,961	0.87	4,000	0.99	107,841	0.94
Galen	5,368	0.83	2,475	0.97	13,285	0.91
Adult Mouse	1,649	0.99	507	1	8,575	0.99
Wine	262	0.89	141	0.98	677	0.93
Nautilus	15	1	11	1	83	0.86
Cell	206	0.87	114	0.96	2,732	0.96
DOLCE-lite	49	0.92	46	0.98	497	0.66
Software	1,728	0.81	620	1	6,001	0.81
Family Tree	48	0.77	37	0.78	223	0.83
General Ontology	281	0.83	278	0.83	4,182	0.83
Substance	221	1	135	1	1,417	0.95
Generations	46	0.65	22	1	71	0.90
Periodic Table	25	1	22	1	80	1

Results for the other, relatively small ontologies are similar to those of Galen and in some cases more prominent (Table 2). The highest improvement (66% of $\int(\mathcal{T})$) was achieved in the DOLCE-Lite ontology [14].

7 Related Work

The work on knowledge compilation [15] is closely related to the work presented in this paper. Knowledge compilation is a family of approaches, in which a knowledge base is transformed in an off-line phase into a normal form, for which reasoning is cheaper. The hope is that the one-off cost of the initial preprocessing will be justified by the computational savings made on subsequent reasoning. One of such normal forms proposed in description logics is the prime implicates normal form for $\mathcal{ALC}$ ontologies [4]. Prime implicates of a logical formula are defined to be their strongest clausal consequences. Concepts in the prime implicates normal form are expected to be easier to read and understand. Reasoning is also expected to be more efficient for knowledge bases in this normal form. For example, concept subsumption can be tested in quadratic time. However, the problem with such normal forms is the blowup caused by the transformation. For $\mathcal{ALC}$ ontologies, a doubly-exponential blowup in the concept size can occur. Given that reasoning in $\mathcal{ALC}$ is PSpace-complete [16], such a transformation can be disadvantageous in general.

Grimm et al. [3] propose two different algorithms for eliminating semantically redundant axioms from ontologies, which is one of the sources of non-succinctness. However, as shown in Section 3, it does not guarantee that we obtain a minimal TBox in $[(]\mathcal{T})$. The advantage of this restricted approach to improving succinctness is that the result contains only axioms that are familiar to the users of the ontology.

Work on laconic and precise justifications [17] (minimal parts of the ontology implying a particular axiom or axioms) is also related to this paper. The authors propose an algorithm for computing laconic justifications – justifications that do not contain any logically superfluous parts. Laconic justifications can then be used to derive precise justifications – justifications that consist of flat, small axioms, and are important for the generation of semantically minimal repairs.

Nikitina et al. [18] propose an algorithm for an efficient handling of redundancy in inconsistent ontologies during their repair. Similarly to the approach by Grimm et al. axioms are considered as atoms that cannot be further separated into parts.

8 Summary and Outlook

We have considered the problem of finding minimal equivalent representations for ontologies expressed in the lightweight description logic $\mathcal{EL}$. We have shown that the task of finding such a representation (or rather: its related decision problem) is NP-complete. Further, we have identified a class of TBoxes for which the problem is tractable. We have implemented a polynomial algorithm for minimizing the above class of TBoxes. For general TBoxes, the algorithm can be used as a heuristic. We have implemented the algorithm and presented experimental results, which show that the complexity of various existing ontologies can be improved. For instance, in case of Galen, the number of complex concepts occurrences could be reduced by 955 and the number of references to atomic concepts and roles by 1130.

There are various natural extensions of this work. Inspired by recent results on uniform interpolation in $\mathcal{EL}$ [8], the problem can be extended to finding minimal representations for ontologies using a signature extension. The results in [8] imply that, even for the minimal equivalent representation of an ontology, an up to triple-exponentially more succinct representation can be obtained by extending its signature. Auxiliary concept symbols are therefore important contributors towards the succinctness of ontologies, e.g., used as shortcuts for complex $\mathcal{EL}$ concepts or disjunctions thereof. The results of our evaluation indicate that there are many complex concept expression that occur repeatedly in ontologies but do not have an equivalent atomic concept that could be used instead. Therefore, introducing names for such frequently used concepts could yield a further decrease of the ontology's complexity.

The results obtained within this paper can be transferred to the context of ontology reuse, where rewriting is applied to obtain a compact representation of the facts about a subset of terms [19], in particular in its extended form as suggested above.

Finally, minimizing representations is an interesting problem for knowledge representation formalisms in general, and similar questions can (and should) be asked for more expressive ontology languages.

References

1. 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 the 12th International Congress of the European Federation for Medical Informatics (MIE 1994), pp. 229–233 (1994)
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press (2003)

3. Grimm, S., Wissmann, J.: Elimination of redundancy in ontologies. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 260–274. Springer, Heidelberg (2011)
4. Bienvenu, M.: Prime implicates and prime implicants: From propositional to modal logic. Journal of Artificial Intelligence Research (JAIR) 36, 71–128 (2009)
5. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C. (eds.): OWL 2 Web Ontology Language: Profiles. W3C Recommendation (October 27, 2009), `http://www.w3.org/TR/owl2-profiles/`
6. OWL Working Group, W.: OWL 2 Web Ontology Language: Document Overview. W3C Recommendation (October 27, 2009), `http://www.w3.org/TR/owl2-overview/`
7. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), pp. 364–369 (2005)
8. Nikitina, N., Rudolph, S.: ExpExpExplosion: Uniform interpolation in general EL terminologies. In: Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012), pp. 618–623 (2012) (Shortlisted for best paper awards)
9. Konev, B., Walther, D., Wolter, F.: Forgetting and uniform interpolation in large-scale description logic terminologies. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009), pp. 830–835 (2009)
10. Spackman, K.A., Campbell, K.E., Cote, R.A.: Snomed rt: A reference terminology for health care. In: Proceedings of the AIMA Fall Symposium, pp. 640–644 (1997)
11. Kazakov, Y., Krötzsch, M., Simančík, F.: ELK reasoner: Architecture and evaluation. In: Proceedings of the OWL Reasoner Evaluation Workshop 2012 (ORE 2012). CEUR Workshop Proceedings, vol. 858. CEUR-WS.org (2012)
12. Sioutos, N., Coronado, S.D., 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)
13. Ashburner, M.: Gene ontology: Tool for the unification of biology. Nature Genetics 25, 25–29 (2000)
14. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening ontologies with dolce. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, p. 166. Springer, Heidelberg (2002)
15. Darwiche, A., Marquis, P.: A knowledge compilation map. Journal of Artificial Intelligence Research (JAIR) 17, 229–264 (2002)
16. Schmidt-Schauß, M., Smolka, G.: Attributive concept descriptions with complements. Artificial Intelligence 48(1), 1–26 (1991)
17. Horridge, M., Parsia, B., Sattler, U.: Laconic and precise justifications in owl. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 323–338. Springer, Heidelberg (2008)
18. Nikitina, N., Rudolph, S., Glimm, B.: Reasoning-supported interactive revision of knowledge bases. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011), pp. 1027–1032 (2011)
19. Nikitina, N., Glimm, B.: Hitting the sweetspot: Economic rewriting of knowledge bases. In: Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J.X., Hendler, J., Schreiber, G., Bernstein, A., Blomqvist, E. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 394–409. Springer, Heidelberg (2012)