Towards Parallel Repair: An Ontology Decomposition-based Approach*

Yue Ma[1] and Rafael Peñaloza[1,2]

[1] Theoretical Computer Science, TU Dresden, Germany
[2] Center for Advancing Electronics Dresden, Germany
{mayue,penaloza}@tcs.inf.tu-dresden.de

Abstract. Ontology repair remains one of the main bottlenecks for the development of ontologies for practical use. Many automated methods have been developed for suggesting potential repairs, but ultimately human intervention is required for selecting the adequate one, and the human expert might be overwhelmed by the amount of information delivered to her. We propose a decomposition of ontologies into smaller components that can be repaired in parallel. We show the utility of our approach for ontology repair, provide algorithms for computing this decomposition through standard reasoning, and study the complexity of several associated problems.

1 Introduction

One of the main challenges in real-world Description Logic (DL) based applications is to maintain the ontologies consistent with the intended domain modeling, a task often involving interactions with domain experts. It is thus desirable to allow experts to work distributively and in parallel for verifying and correcting unexpected logical consequences. This task is of a particular importance in scenarios where information is precious, large, and complex, and where manual verification of a repair plan is a necessary but labour-consuming task.

Current research on ontology repair (see e.g. [5, 6, 8, 10, 15, 17, 20]) focuses on pinpointing the axiomatic causes, called *MinAs* of an unintended consequence and using them to compute a repair plan. For example, consider the ontology

$$\mathcal{O} = \{A \sqsubseteq B_i, B_i \sqsubseteq C \mid 1 \leq i \leq n+1\} \cup \{B_{i+1} \sqsubseteq B_i \mid 1 \leq i \leq n\}.$$

Clearly, $\mathcal{O}$ implies $A \sqsubseteq C$, and there are diverse reasons for this entailment: $\{A \sqsubseteq B_i, B_i \sqsubseteq C\}$ and $\{B_i \sqsubseteq C, B_{i+1} \sqsubseteq B_i, A \sqsubseteq B_{i+1}\}$ for each $i, 1 \leq i \leq n$. This situation is depicted in Figure 1, where each axiom of $\mathcal{O}$ is represented with a node, and the different MinAs for $A \sqsubseteq C$ are surrounded by an ellipse.

To repair the ontology, minimal hitting sets [18], represented with squared nodes in each subfigure from Figure 1, are the commonly used. The reason for

* This work is supported by the DFG Research Unit FOR 1513, project B1 and within the Cluster of Excellence 'cfAED'.

Fig. 1. Two minimal hitting sets (sets of squared dots) of the illustrative ontology $\mathcal{O}$

Fig. 2. A decomposition of $\mathcal{O}$ (thickened circles)

this choice is that a minimal hitting set of all MinAs corresponds to a minimal set of elements that need to be removed to avoid this consequence. That is, it provides a minimal repair plan, which can be computed automatically.

When multiple experts are available for authenticating a repair plan, it would be desirable to divide this task in a manner that allows experts to work in parallel. One way is to distribute one MinA to each expert. However, due to the overlaps between the MinAs, this may lead to unnecessary extra re-validations from different experts. Alternatively, we could distribute the minimal hitting sets; it is then unclear which hitting set should be sent from the many available. Consider the case where several hitting sets are precomputed and each one of them is sent to an expert. Besides the problem of possible overlaps among hitting sets in general, the large sizes of these sets (e.g. $n + 1$ in the example ontology $\mathcal{O}$), means a high work load for each expert. Moreover, if a given hitting set is partitioned and distributed among different experts, there is a need of a communication mechanism for all experts to be informed of any decision made by the others, to avoid successive clashes and unnecessary effort.

Based on these observations, we propose a novel methodology for decomposing ontologies that allows experts to validate and apply a repair plan in parallel. Generally speaking, we are interested in a decomposition that guarantees that:

- no communication is required among experts and no axiom is submitted to more than one expert; and
- the union of the repairs returned is free of the unintended consequence.

For the example ontology $\mathcal{O}$, a possible such decomposition is shown in Figure 2, where each thickened ellipse is a component delivered to an expert for repair. This decomposition allows n experts to work distributively, analysing two axioms each. Stated in a general ontology language, our methodology is applicable to all logic-based applications that have a necessity to have experts assisting the repair process. Moreover, to keep the modeling convention of a domain, we assume that subontologies should be delivered in their original format. All these features distinguish the present work from the existing research efforts on ontology decompositions, such as ontology modularization [7], ontology masking [8], decompositions based on minimal unsatisfiable sets (MUSs) [9], and root and derived axioms for unsatisfiable concepts [12], among many others. Due to space limitations, all proofs are left in a technical report [14].

2 Preliminaries

To remain as general as possible, we do not fix any specific knowledge representation formalism, but assume that we have an *ontology language* that describes two classes of well-formed formulas called *axioms* and *consequences*, respectively. An *ontology* $\mathcal{O}$ is a finite set of axioms, and a subset of $\mathcal{O}$ is called a *subontology* of $\mathcal{O}$. For a fixed ontology language $\mathcal{L}$, a *monotone consequence relation* $\models$ is a binary relation between ontologies $\mathcal{O}$ and consequences α of $\mathcal{L}$ such that, if $\mathcal{O} \models \alpha$ and $\mathcal{O} \subseteq \mathcal{O}'$, then $\mathcal{O}' \models \alpha$. If $\mathcal{O} \models \alpha$, we say that $\mathcal{O}$ *entails* α. For the rest of this paper, we denote as $\mathfrak{C}$ the complexity of deciding entailments in $\mathcal{L}$.

Two examples of ontological languages are $\mathcal{HL}$ and $\mathcal{EL}$, among other DLs [2]. In $\mathcal{HL}$ axioms and consequences are *Horn clauses* $p_0 \leftarrow p_1 \wedge \ldots \wedge p_n$, $n \geq 0$, where each p_i is a propositional variable $0 \leq i \leq n$. In this case, the standard logical entailment relation $\vdash$ is a monotone consequence relation. In $\mathcal{EL}$, concepts are built from disjoint sets $\mathsf{N_C}$ and $\mathsf{N_R}$ using the rule $C ::= A \mid \top \mid C \sqcap C \mid \exists r.C$, where $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. Axioms and consequences in this logic are *GCIs* $C \sqsubseteq D$, where C, D are concepts, and *subsumption* is a monotone consequence relation. In $\mathcal{HL}$ and $\mathcal{EL}$, the consequence relation can be decided in polynomial time [1, 3, 4].

If an unwanted consequence α follows from an ontology, we are interested in *repairing* it, by finding an appropriate weaker ontology that does not entail α anymore. Formally, a *repair* for $\mathcal{O}$ w.r.t. α is an ontology $\mathcal{R}$ such that (i) $\mathcal{R} \not\models \alpha$, (ii) $\mathcal{R}$ is weaker than $\mathcal{O}$; that is, for every consequence β, if $\mathcal{R} \models \beta$, then $\mathcal{O} \models \beta$, and (iii) $\mathcal{R}$ is a *minimal change* of $\mathcal{O}$. We simply say a *repair* for $\mathcal{O}$ when the consequence is clear from context. If α is an erroneous consequence, then a repair describes a minimal change of $\mathcal{O}$ that removes it. The notion of minimal change depends on the ontological knowledge and the desired application. For example, one can define repairs to be maximal subontologies that avoid the consequence (see e.g. [13]) or allow for a more fine-grained decomposition of the axioms [8]. We remain as general as possible, and allow any notion to be considered. As there is typically more than one repair for any given consequence, usually a human expert is needed to identify the best one. To aid in the repair process, it is often helpful to identify the axiomatic causes of the consequence, called MinAs.[3]

Definition 1 (MinA). *Let $\mathcal{O}$ be an ontology and α a consequence with $\mathcal{O} \models \alpha$. A subontology $\mathcal{M} \subseteq \mathcal{O}$ is a* MinA *for $\mathcal{O}$ w.r.t. α if $\mathcal{M} \models \alpha$ and for every $\mathcal{M}' \subsetneq \mathcal{M}$, $\mathcal{M}' \not\models \alpha$.*

An axiom is said to be *consequence-free* w.r.t. α if it is not in any MinA for $\mathcal{O}$ w.r.t. α. When the consequence is clear from context, we simply call it *free axiom*. Our goal is to divide the ontology into different components that can be repaired in parallel. Since free axioms are never responsible for the occurrence of the erroneous consequence α, for the rest of the paper we assume that the ontology $\mathcal{O}$ does not contain any consequence free axiom w.r.t. α.

[3] MinAs receive several names. They are e.g. called MUS in propositional logic, and MUPS [19] or justifications [8, 11] in DLs.

3 General Ontology Decompositions

We now formalize the notion of parallel ontology repair via decompositions. We start with the ideal case where an ontology can be fully decomposed into different components and repaired in parallel. However, the existence of this case can not be guaranteed. Hence, we subsequently propose a series of ontology decompositions that do not necessarily partition the whole ontology, but still allow repairs to be performed independently on each component.

Definition 2 (Perfect Partition). *Given an ontology $\mathcal{O}$ and a consequence α, a* perfect partition *of $\mathcal{O}$ w.r.t. α is a partition $\{K_1, \ldots, K_n\}$ of $\mathcal{O}$ such that, if $\mathcal{R}_i$ is a repair of K_i for $i, 1 \leq i \leq n$, then $\bigcup_{i=1}^{n} \mathcal{R}_i$ is a repair of $\mathcal{O}$ w.r.t. α. In this case, the perfect partition has* size *n.*

A perfect partition provides a way to break down an ontology into multiple disjoint subontologies that can be resolved independently, and the union of their repairs will lead to a repair of the whole ontology.[4] This characterizes the intuition of repairing an ontoloy in parallel.

Example 3. Let $\mathcal{O} = \{A \sqsubseteq B_i, B_i \sqsubseteq C \mid 1{\leq}i{\leq}n\} \cup \{A \sqsubseteq D_i, D_i \sqsubseteq C \mid 1{\leq}i{\leq}m\}$. The sets $K_i = \{A \sqsubseteq B_i, B_i \sqsubseteq C\}$ and $L_i = \{A \sqsubseteq D_i, D_i \sqsubseteq C\}$, produce a perfect partition of $\mathcal{O}$ w.r.t. $A \sqsubseteq C$. Repairing each K_i and L_i w.r.t. $A \sqsubseteq C$ leads to a repair of $\mathcal{O}$ w.r.t. the same consequence.

In this example, the set of mutually disjoint MinAs produces a perfect partition of $\mathcal{O}$. However, this is not true in general, as illustrated by the following example.

Example 4. Consider the ontology

$$\mathcal{O} = \{A \sqsubseteq B_1, B_1 \sqsubseteq C, A \sqsubseteq B_1 \sqcap B_2, B_2 \sqsubseteq C\}.$$

$\mathcal{M}_1 = \{A \sqsubseteq B_1, B_1 \sqsubseteq C\}$ and $\mathcal{M}_2 = \{A \sqsubseteq B_1 \sqcap B_2, B_2 \sqsubseteq C\}$ are two disjoint MinAs for $A \sqsubseteq C$, and $\{\mathcal{M}_1, \mathcal{M}_2\}$ is a partition of $\mathcal{O}$. However, they do not form a perfect partition: $\{B_1 \sqsubseteq C\}$ and $\{A \sqsubseteq B_1 \sqcap B_2\}$ are repairs of $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively, but their union $\{B_1 \sqsubseteq C, A \sqsubseteq B_1 \sqcap B_2\}$ is not a repair of $\mathcal{O}$.

The ontology $\mathcal{O}$ from this example does not have any perfect partition of size ≥ 2. Clearly, a perfect partition of size 1 always exists: the whole ontology itself is such one. Since only decompositions of size larger than 1 are meaningful for parallel repair, in the rest of the paper, by a perfect partition we mean its size is equal or greater than 2 unless explicitly stated otherwise. The following theorem provides a simple sufficient condition for an ontology to have a perfect partition.

Proposition 5. *Let $\mathcal{M}_1, \ldots, \mathcal{M}_n$ be* all *the MinAs of $\mathcal{O}$ w.r.t. a consequence α. If these MinAs are all pairwise disjoint, $\mathcal{O}$ has a perfect partition of size n.*

In general the condition of partitioning the ontology is too strong. In some cases, a decomposition of larger size can be obtained if some axioms are not included in any of the components. We formalize this idea next.

[4] A subontology is the repair of itself if it does not contain a MinA.

Definition 6 (Perfect Partial-Partition). *A set of mutually disjoint subontologies* $\{K_1, \ldots, K_n\}$ *is a* perfect partial partition *of* $\mathcal{O}$ *w.r.t. a consequence* α *if it satisfies the following condition: given repairs* $\mathcal{R}_i$ *of* K_i *for each* $i, 1 \leq i \leq n$, $\bigcup_{i=1}^{n} \mathcal{R}_i$ *is a repair for* $\bigcup_{i=1}^{n} K_i$ *w.r.t.* α.

To characterize these partial partitions, we introduce the notion of decomposition. This is inspired by the work [9] and the insight that the different subontologies submitted to experts should be *inner-connected*, but *outer-isolated*.

Definition 7 (Decomposition). *Let* $\mathcal{O}$ *be an ontology entailing a consequence* α. *A* decomposition *of* $\mathcal{O}$ *w.r.t.* α *is a set* $\mathcal{D} := \{K_1,\ldots,K_n\}$ *of mutually disjoint subsets of* $\mathcal{O}$ *such that: (i) for every* $i, 1 \leq i \leq n$, $K_i \models \alpha$, *and (ii) for every MinA* $\mathcal{M}$ *for* $\bigcup_{i=1}^{n} K_i$ *w.r.t.* α, *there is an* $i, 1 \leq i \leq n$ *with* $\mathcal{M} \subseteq K_i$.

The decomposition $\mathcal{D}$ *has* size n, *or is an* n-decomposition*; each* K_i *is a* component *of* $\mathcal{D}$; *and* $\mathfrak{D}_n(\mathcal{O}, \alpha)$ *denotes the set of* n*-decompositions of* $\mathcal{O}$ *w.r.t. the consequence* α.

Intuitively, the condition (i) characterizes the inner-connection such that each expert is assigned with a subontology containing some causes of the consequence. And the condition (ii) guarantees the outer-isolation as illustrated by the following example.

Example 8. Consider again the ontology $\mathcal{O}$ and the MinAs $\mathcal{M}_1$ and $\mathcal{M}_2$ from Example 4. $\{\mathcal{M}_1, \mathcal{M}_2\}$ is not a decomposition because there $\{A \sqsubseteq B_1 \sqcap B_2, B_1 \sqsubseteq C\}$ is also a MinA contained in $\mathcal{M}_1 \cup \mathcal{M}_2$, violating the condition (ii) of Definition 7. Suppose that $\mathcal{M}_1$ and $\mathcal{M}_2$ were distributed to two experts, which return the repairs $B_1 \sqsubseteq C \in \mathcal{M}_1$ and $A \sqsubseteq B_1 \sqcap B_2 \in \mathcal{M}_2$, respectively. Then, these repairs together would still entail the unwanted consequence.

Theorem 9. *If* $\mathcal{D}$ *is an* n*-decomposition, then* $\mathcal{D}$ *is a perfect partial-partition of size* n.

Theorem 9 tells that the inconsistencies in each component of a decomposition can be resolved distributively and the merged repair becomes consistent.

Notice that the union of the components needs not to be the whole ontology, so a decomposition defined Defintion 7 is not necessarily a perfect partition. Let $\mathcal{U} = \mathcal{O} \setminus \bigcup_{i=1}^{n} K_i$. If $\mathcal{U} \neq \emptyset$, we cannot guarantee that $\bigcup_{i=1}^{n} \mathcal{R}_i \cup U$ does not entail the consequence. Consider again the example from the introduction with $n = 2$. For the decomposition $\{K_1, K_2\}$ with $K_i = \{A \sqsubseteq B_i, B_i \sqsubseteq C\}$ for $i = 1, 2$, the axiom $B_2 \sqsubseteq B_1 \in \mathcal{U}$. The subontologies $\mathcal{R}_1 = \{B_1 \sqsubseteq C\}$ and $\mathcal{R}_2 = \{A \sqsubseteq B_2\}$ are repairs for K_1 and K_2, respectively. However, $\bigcup_{i=1}^{n} \mathcal{R}_i \cup \mathcal{U} \models A \sqsubseteq C$. To solve this issue, one can either drop the unconfirmed axiom $B_2 \sqsubseteq B_1$ because the other axioms that can form a MinA with it have been verified by experts; or, repeat the same process by constructing a new decomposition to resolve the remaining inconsistencies. To parallelize the effort of repairing the ontology, we are interested in decompositions of maximal size. The corresponding decision problem is the following.

Problem: MAX-DECOM
Input: an ontology $\mathcal{O}$, a consequence α, an integer m
Question: is there an m-decomposition of $\mathcal{O}$ w.r.t. α?

As we show next, this problem can be solved using a special kind of decomposition made of MinAs only.

Definition 10 (MinA Decomposition). *A decomposition $\mathcal{D} = \{K_1,...,K_n\}$ of $\mathcal{O}$ w.r.t. α is a* MinA decomposition *if for every $i, 1 \leq i \leq n$, K_i is a MinA for $\mathcal{O}$ w.r.t. α.*

Lemma 11. *$\mathcal{O}$ has an m-decomposition if and only if $\mathcal{O}$ has a MinA decomposition of size m.*

Based on this lemma, we can decide MAX-DECOM by guessing m disjoint subsets $K_1, \ldots, K_m$ of $\mathcal{O}$ in polynomial time, and verifying that they form a MinA decomposition; that is, solving the following problem.

Problem: IS-(MINA)-DECOM
Input: ontology $\mathcal{O}$, consequence α, $\mathcal{D} = \{K_1, \ldots, K_m\}$
Question: is $\mathcal{D}$ a (MinA) decomposition of $\mathcal{O}$ w.r.t. α?

To decide this problem, we can guess a MinA $\mathcal{M}$ that violates the second condition of Definition 7. To verify that $\mathcal{M}$ is indeed a MinA, polynomially many entailment tests are required. Thus we get the following bound.

Lemma 12. IS-DECOM *is in co*$\mathrm{NP}^{\mathfrak{C}}$.

This lemma provides only an upper bound for the problem; in particular, it also shows that IS-MINA-DECOM is in $\mathrm{coNP}^{\mathfrak{C}}$. In general, these upper bounds do not need to be tight. For example, in $\mathcal{HL}$ it is possible to decide in polynomial time whether the set $\mathcal{D} = \{\mathcal{M}_1, \ldots, \mathcal{M}_m\}$ is exactly the set of all MinAs for an ontology $\mathcal{O}$ [16]. It is easy to see that the set $\mathcal{D}$ is a MinA decomposition iff $\mathcal{D}$ is the set of all MinAs for $\bigcup_{i=1}^{m} \mathcal{M}_i$. Thus, IS-MINA-DECOM for $\mathcal{HL}$ can be solved in polynomial time. However, it is still an open question whether the same holds for IS-DECOM, or for the more expressive logic $\mathcal{EL}$, or for other logics with polynomial time entailment problems.

Algorithm 1 uses all these ideas to decide MAX-DECOM. The procedure NOT-MINA receives as input an ontology $\mathcal{O}$ and a consequence α, and answers "yes" if $\mathcal{O}$ is *not* a MinA w.r.t. α. This is the case if either $\mathcal{O} \not\models \alpha$ (line 18), or there is a strict subset of $\mathcal{O}$ that still entails α (line 17). The other two procedures perform a non-deterministic guess; MAX-DECOM guesses the components, while IS-DECOM guesses a MinA that violates the second condition of Definition 7.

Theorem 13. MAX-DECOM *is in* $(\Sigma_2^P)^{\mathfrak{C}}$.

In particular, this theorem shows that MAX-DECOM is in Σ_2^P for both $\mathcal{HL}$ and $\mathcal{EL}$. As before, the bound needs not be tight; indeed, using the arguments described above, it is easy to see that this problem is in NP for $\mathcal{HL}$.

Algorithm 1 Deciding MAX-DECOM

procedure MAX-DECOM($\mathcal{O}$, α, m)
  for $1 \leq i \leq m$ **do**
    guess $K_i \subseteq \mathcal{O}$
    if NOT-MINA(K_i, α) **then return** no
  if $K_i \cap K_j = \emptyset$ for all $i, 1 \leq i < j \leq m$ **then**
    return IS-DECOM($\{K_1, \ldots, K_m\}$, α)
  else return no
end procedure
procedure IS-DECOM($\mathcal{D}$, α)
  guess $\mathcal{M} \subseteq \bigcup_{K \in \mathcal{D}} K$
  if $\mathcal{M} \not\subseteq K$ for all $K \in \mathcal{D}$ **then**
    return NOT-MINA($\mathcal{M}$, α)
  else return yes
end procedure
procedure NOT-MINA($\mathcal{O}$, α)
  for all $t \in \mathcal{O}$ **do**
    if $\mathcal{O} \setminus \{t\} \models \alpha$ **then return** yes
  return $\mathcal{O} \not\models \alpha$
end procedure

Algorithm 2 Deciding PERFECT-PART

procedure FULL-DECOM($\mathcal{O}$, α, m)
  for $1 \leq i \leq m$ **do**
    guess $K_i \subseteq \mathcal{O}$
  if $\{K_1, \ldots, K_m\}$ is a partition of $\mathcal{O}$ **then**
    return IS-DECOM($\{K_1, \ldots, K_m\}$, α)
  else return no
end procedure

We are mainly interested in decompositions of maximal size since they allow for a more efficient parallelization of the repairing procedure: each component can be repaired independently, and the properties of the decomposition guarantee that the union of these repairs does not entail the consequence. However, we are interested in finding a repair for the whole input ontology $\mathcal{O}$, not just for those axioms appearing in the decomposition. As described before, ideally we would find a perfect partition of size n, for a given natural number n. Accordingly, we want to decide whether such a decomposition exists.

Problem: PERFECT-PART
Input: an ontology $\mathcal{O}$, a consequence α, an integer m
Question: is there a perfect partition of $\mathcal{O}$ w.r.t. α of size m?

Algorithm 2 describes a method for deciding PERFECT-PART. In a nutshell, it guesses a partition $\mathcal{D}$ of $\mathcal{O}$ of size m, and then verifies, through a call to the

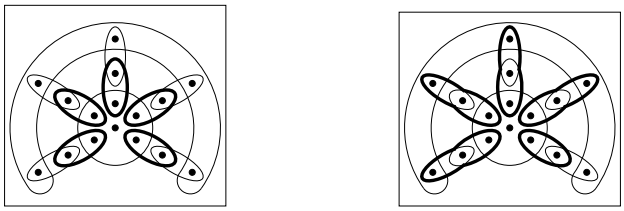

Fig. 3. Two 5-decompositions: left a MinA decomposition; right a justified decomposition with minimal remain.

procedure IS-DECOM from Algorithm 1, that $\mathcal{D}$ is a decomposition. This yields the same complexity upper bound as for MAX-DECOM.

Theorem 14. PERFECT-PART *is in* $(\Sigma_2^P)^{\mathfrak{C}}$.

We can in general visualize the set of MinAs for a consequence α as a *hypergraph* $G_{\mathcal{O},\alpha}$. Every axiom in $\mathcal{O}$ is represented through a node, and every MinA is a hyperedge in this hypergraph. Consider the sub-hypergraph $H_{\mathcal{O},\alpha}$ of $G_{\mathcal{O},\alpha}$ that contains only nodes belonging to some hyperedge; i.e., where all free axioms have been removed. It is easy to see that there is a perfect partition of size m iff there are at least m maximally connected subgraphs of $H_{\mathcal{O},\alpha}$. This is usually not a desired behavior for parallelization, since the number of components will be usually small. For example, the ontology depicted in Figure 3 allows for a decomposition of size 5, but its only perfect partition has size 1. On the other hand, if $\mathcal{D}$ is an n-decomposition of $\mathcal{O}$ w.r.t. α, then $\mathcal{D}$ is a perfect partition of $\bigcup_{K\in\mathcal{D}} K$ w.r.t. α of size n.

To maximize the number of components, and hence the degree of parallelization, we are willing to ignore some axioms, as described by the notion of decomposition. However, we should try to submit to the experts as much information from the original ontology as possible, to ensure an effective repair process. This will be the focus of the next section.

4 Maximally Informative Decompositions

While MinA decompositions are useful for deciding the existence of a decomposition of a given size, they, by construction, ignore a large amount of axioms from the ontology. Consider again the example in Figure 3. The maximal size of a decomposition of this ontology is 5, as shown on the left-hand-side graph through a MinA decomposition. In total, the components contain only 10 out of the 16 axioms from the ontology. Moreover, there is a whole MinA for the consequence that is left out of the decomposition; even if all the components are corrected, the obtained ontology would still entail the error. On the right-hand-side, we can observe a decomposition of the same size 5, whose components extend those of the MinA decomposition, and uses 14 out of the 16 axioms.

Algorithm 3 Finding a minimal remain decomposition

```
1: procedure FIND-MIN-REM-DEC(𝒪, α, m)
2:     𝒟 ← FIND-DECOM(𝒪, α, m)
3:     ℛ ← ⋃_{K∈𝒟} K
4:     for all t ∈ 𝒪 \ ℛ do
5:         if PERFECT-PART(ℛ ∪ {t}, α, m) then
6:             ℛ ← ℛ ∪ {t}
7:     return FIND-PERFECT-PART(ℛ, α, m)
8: end procedure
```

When we send a subontology for repair, it should be as informative as possible, to ensure that no simple errors are ignored. Clearly, the less axioms that are removed to build the decomposition, the more information that is gathered and used during the parallel repair. Thus, we are interested in finding, among all decompositions of maximal size, those that include the most axioms possible.

Definition 15 (Minimal Remain Decomposition). *Let n be a natural number and $\mathcal{O} \models \alpha$. The* remain *of a decomposition $\mathcal{D}$ is $\mathcal{O} \setminus (\bigcup_{K\in\mathcal{D}} K)$. A decomposition $\mathcal{D} \in \mathfrak{D}_n(\mathcal{O}, \alpha)$ is a* minimal remain decomposition *if there is no $\mathcal{D}' \in \mathfrak{D}_n(\mathcal{O}, \alpha)$ with $\bigcup_{K\in\mathcal{D}} K \subsetneq \bigcup_{K\in\mathcal{D}'} K$.*

In other words, a decomposition $\mathcal{D}$ has minimal remain if it is not possible to decompose a proper superset of $\bigcup_{K\in\mathcal{D}} K$ in the same number of components. Consider again the example in Figure 3. The decomposition on the right-hand-side has a remain with two axioms. It is a minimal remain decomposition, since adding any of these two axioms would destroy the properties of decompositions.

To find a minimal remain decomposition, we can recursively try to add axioms from the remainder of a previously known n-decomposition, until none can be added, as described in Algorithm 3. More precisely, let $\mathcal{D}$ be an n-decomposition, for instance, a MinA decomposition of size n that was constructed through Algorithm 1, and let $\mathcal{R} = \bigcup_{K\in\mathcal{D}} K$; i.e., $\mathcal{R}$ is the complement of the remain of $\mathcal{D}$. For each axiom t in the remain of $\mathcal{D}$, we decide whether $\mathcal{R} \cup \{t\}$ has a full n-decomposition. If so, then t is added to $\mathcal{R}$. At the end of this iteration, $\mathcal{R}$ has a maximal subontology that allows for a perfect partition of size n. Any perfect partition of this subontology is hence guaranteed to have minimal remain.

The internal subprocedure in lines 3 to 6 of Algorithm 3 can be easily adapted to verify that the decomposition $\mathcal{D}$ has minimal remain.

Problem: IS-MIN-REM-DECOM
Input: ontology $\mathcal{O}$, consequence α, $\mathcal{D} = \{K_1, \ldots, K_m\}$
Question: is $\mathcal{D}$ a minimal remain decomposition of $\mathcal{O}$ w.r.t. α?

In the variant algorithm, one only has to check that $\mathcal{R} \cup \{t\}$ has no perfect partition, for every $t \in \mathcal{O} \setminus \mathcal{R}$. If that is the case, then $\mathcal{D}$ has minimal remain.

Theorem 16. IS-MIN-REM-DECOM *is in* $(\Pi_2^P)^{\mathfrak{C}}$.

Fig. 4. A Pareto decomposition without minimal remain

Notice that the decomposition obtained through Algorithm 3 may have no resemblance with the first decomposition found at line 2. Indeed, the only requirement is that there is a perfect partition of all the axioms used, which could differ greatly from the original one. In some cases, e.g. when the first decomposition was constructed from some specific MinAs that should remain connected, it is desirable to only add axioms to the existing components.

Definition 17 (Pareto Decomposition). *Given $n \in \mathbb{N}$, $\mathcal{O} \models \alpha$, and decompositions $\mathcal{D}, \mathcal{D}' \in \mathfrak{D}_n(\mathcal{O}, \alpha)$, $\mathcal{D}$ is* contained *in $\mathcal{D}'$, denoted by $\mathcal{D} \subseteq \mathcal{D}'$ if, for every $K \in \mathcal{D}$ there is a $K' \in \mathcal{D}'$ such that $K \subseteq K'$. $\mathcal{D}$ is a* Pareto decomposition *if there is no $\mathcal{D}' \neq \mathcal{D}$ with $\mathcal{D} \subseteq \mathcal{D}'$.*

Clearly, every minimal remain decomposition is also Pareto. The converse, however, does not hold. Consider the situation depicted in Figure 4, where the ellipses represent the different MinAs for a given consequence. This ontology can be decomposed into a perfect partition of size two, simply by considering its connected subgraphs. It is easy to see that the 2-decomposition $\mathcal{D}$, where one component is formed by the diamond-shaped axioms, and the other by the triangle-shaped axioms is a Pareto decomposition. However, the dot-shaped axiom is in the remain of $\mathcal{D}$. This implies that $\mathcal{D}$ is not a minimal-remain decomposition.

To find a Pareto decomposition, we can use the same ideas of Algorithm 3. We first find a decomposition, and then try to add each of the remaining axioms to one of the components, as long as this addition still yields a decomposition. It can also be restricted to decide whether an input decomposition is already Pareto or not. Notice, however, that in line 5 the algorithm for deciding Pareto decomposition does not need to verify whether a set of axioms accepts a full decomposition, but rather whether a set of subontologies forms a decomposition, which, as seen before, is a simpler problem. Thus, we have the following.

Problem: IS-PARETO-DECOM
Input: ontology $\mathcal{O}$, consequence α, $\mathcal{D} = \{K_1, \ldots, K_m\}$
Question: is $\mathcal{D}$ a Pareto decomposition of $\mathcal{O}$ w.r.t. α?

Theorem 18. IS-PARETO-DECOM *is in* $\mathrm{NP}^{\mathfrak{C}}$.

We have considered decompositions that maximize the information stored in the components in two different ways: either by minimizing the elements that remain out of the decomposition, or by maximizing the components in a Pareto optimal manner. Notice, however, that some of the axioms included in a component might be irrelevant for the repair of that specific component. For that reason, we might also be interested in *justified* decompositions.

Table 1. Instantiation of complexity results for decomposition to different DLs.

Language	Consequence $\models$		IS-DEC	MAX-DEC	PERF-PART	IS-MIN-REM-D	IS-PARETO-D	IS-JUST-D
general	general	$\mathfrak{C}$	*co*NP$^{\mathfrak{C}}$	$(\Sigma_2^P)^{\mathfrak{C}}$	$(\Sigma_2^P)^{\mathfrak{C}}$	$(\Pi_2^P)^{\mathfrak{C}}$	NP$^{\mathfrak{C}}$	$(\Sigma_2^P)^{\mathfrak{C}}$
$\mathcal{HL}$	entailment	P	P	NP	NP	*co*NP	P	NP
$\mathcal{EL}$	subsumption	P	*co*NP	Σ_2^P	Σ_2^P	Π_2^P	NP	Σ_2^P
$\mathcal{ALC}$	consistency	ExpTime	ExpTime	ExpTime	ExpTime	ExpTime	ExpTime	ExpTime

Definition 19 (Justified Decomposition). *Let $\mathcal{O} \models \alpha$. A decomposition $\mathcal{D}$ is called* justified *if for every $K \in \mathcal{D}$ and every $t \in K$ there exists a MinA $\mathcal{M}$ of $\mathcal{O}$ w.r.t. α such that $t \in \mathcal{M} \subseteq K$.*

Clearly, we can combine this notion with the previous ones and obtain, e.g., Pareto justified decompositions. All the algorithms presented so far can be easily adapted to handle justified decompositions. One only needs to perform an additional check to verify that there is a full MinA for every axiom contained in a component. This test adds a new non-deterministic test, and hence the upper bounds increase to the next level of the polynomial hierarchy. Moreover, this jump in the hierarchy cannot be avoided since deciding whether an axiom is justified in a component is already NP-hard for very simple sublogics of $\mathcal{HL}$ [16].

5 Conclusions

We have introduced several notions of ontology decomposition targeted towards an efficient repair mechanism. Our motivating idea is that human experts, which are usually in demand for a correct repair of an ontology, can be easily overwhelmed by the amount of axioms provided to them. We thus suggest to divide the ontology into disjoint components that can be repaired in parallel, possibly by several different experts. Our definition of decomposition guarantees that the combination of the individual repairs for the components does not yield any new errors, hence providing an efficient parallelization of the repair process.

We have mainly focused on studying the different decision problems associated with decomposing ontologies, and their complexity. Our approach is general, considering an arbitrary monotonic consequence relation over some ontology language. Hence, our complexity analysis can only provide upper bounds; whether these bounds are tight or not is a matter of the specific language used. However, our results can be instantiated to well-known ontology languages. In Table 1 we summarize the complexity of these problems for DLs. The cells show the known upper bound for deciding the problems at each column; cells with a darker background represent tight bounds.

We plan to study the precise complexity of these problems for specific languages, in particular for light-weight DLs. We will also further consider the applicability of our decompositions for practical ontology repair. To this goal, we will implement optimized versions of our algorithms and study the viability of developing a repair plan, in which components are sent to experts in a manner that minimizes the expected total effort and time required to remove the error.

References

1. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence IJCAI-05. Morgan-Kaufmann Publishers, Edinburgh, UK (2005)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2nd edn. (2007)
3. Brandt, S.: Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In: de Mantáras, R.L., L. Saitta (eds.) Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-2004). pp. 298–302. IOS Press (2004)
4. Dowling, W.F., Gallier, J.H.: Linear-time algorithms for testing the satisfiability of propositional horn formulae. J. Log. Program. 1(3), 267–284 (1984)
5. Erdogan, H., Bodenreider, O., Erdem, E.: Finding semantic inconsistencies in umls using answer set programming. In: Fox, M., Poole, D. (eds.) Proc. of the 24th Nat. Conf. on Artificial Intelligence (AAAI'10) (2010)
6. Gebser, M., Schaub, T., Thiele, S., Veber, P.: Detecting inconsistencies in large biological networks with answer set programming. TPLP 11(2-3), 323–360 (2011)
7. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Just the right amount: Extracting modules from ontologies. In: Proceedings of the 16th International Conference on World Wide Web. pp. 717–726. WWW '07, ACM, New York, NY, USA (2007), `http://doi.acm.org/10.1145/1242572.1242669`
8. Horridge, M., Parsia, B., Sattler, U.: Justification masking in ontologies. In: Brewka, G., Eiter, T., McIlraith, S.A. (eds.) Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-12). AAAI Press (2012)
9. Jabbour, S., Ma, Y., Raddaoui, B.: Inconsistency measurement thanks to mus decomposition. In: International conference on Autonomous Agents and Multi-Agent Systems (AAMAS'14) (2014), to appear
10. Jiménez-Ruiz, E., Grau, B.C., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: Algorithms and implementation. In: Raedt, L.D., Bessière, C., Dubois, D., Doherty, P., Frasconi, P., Heintz, F., Lucas, P.J.F. (eds.) Proc. of the 20th European Conf. on Artificial Intelligence (ECAI-12). Frontiers in Artificial Intelligence and Applications, vol. 242, pp. 444–449. IOS Press (2012)
11. Kalyanpur, A.: Debugging and Repair of OWL Ontologies. Ph.D. thesis, The Graduate School of the University of Maryland (2006)
12. Kalyanpur, A., Parsia, B., Sirin, E., Hendler, J.A.: Debugging unsatisfiable classes in owl ontologies. J. Web Sem. 3(4), 268–293 (2005)
13. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant semantics for description logics. In: Hitzler, P., Lukasiewicz, T. (eds.) Proc. of the 4th Int. Conf. on Web Reasoning and Rule Systems (RR'10). Lecture Notes in Computer Science, vol. 6333, pp. 103–117. Springer (2010)
14. Ma, Y., Peñaloza, R.: Towards parallel ontology repair using decompositions. LTCS-Report 14-05, Chair for Automata Theory, Institute for Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany (2014), see http://lat.inf.tu-dresden.de/research/reports.html.
15. Meilicke, C.: Alignment incoherence in ontology matching. Ph.D. thesis, University of Mannheim, Chair of Artificial Intelligence (2011)
16. Peñaloza, R., Sertkaya, B.: On the complexity of axiom pinpointing in the el family of description logics. In: Lin, F., Sattler, U., Truszczynski, M. (eds.) Proc. of the

12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-10). AAAI Press (2010)
17. Peñaloza, R.: Axiom Pinpointing in Description Logics and Beyond. Ph.D. thesis, Institute for Theoretical Computer Science, Faculty of Computer Science, TU Dresden, Germany (2009)
18. Reiter, R.: A theory of diagnosis from first principles. Artif. Intell. 32(1), 57–95 (Apr 1987), http://dx.doi.org/10.1016/0004-3702(87)90062-2
19. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence. pp. 355–360. Morgan Kaufmann Publishers Inc. (2003), http://dl.acm.org/citation.cfm?id=1630659.1630712
20. Suntisrivaraporn, B.: Polynomial-Time Reasoning Support for Design and Maintenance of Large-Scale Biomedical Ontologies. Ph.D. thesis, Institute for Theoretical Computer Science, Faculty of Computer Science, TU Dresden, Germany (2009)