Abduction Framework for Repairing Incomplete $\mathcal{EL}$ Ontologies: Complexity Results and Algorithms (Extended version)

Fang Wei-Kleiner and **Zlatan Dragisic** and **Patrick Lambrix**
Department of Computer and Information Science
Swedish e-Science Research Centre
Linköping University, Sweden

Abstract

In this paper we consider the problem of repairing missing is-a relations in ontologies. We formalize the problem as a generalized TBox abduction problem (GTAP). Based on this abduction framework, we present complexity results for the existence, relevance and necessity decision problems for the GTAP with and without some specific preference relations for ontologies that can be represented using a member of the $\mathcal{EL}$ family of description logics. Further, we present an algorithm for finding solutions, a system as well as experiments.

This is an extended version of (Wei-Kleiner, Dragisic, and Lambrix 2014). For ease of reading we use $\mathcal{EL}$ examples in the introduction and the description of the abduction framework, while (Wei-Kleiner, Dragisic, and Lambrix 2014) uses $\mathcal{EL}^{++}$ examples.

Introduction

Abduction is a reasoning method to generate explanations for observed symptoms and manifestations. When the application domain is described by a logical theory, it is called *logic-based abduction* (Eiter and Gottlob 1995). Logic-based abduction is widely applied in diagnosis, planning, and database updates (Kakas and Mancarella 1990), among others. Provenances in databases (Cheney, Chiticariu, and Tan 2009), specifically the *why provenance*, is an abduction process, in which information about the witnesses to the answer set of a query is provided. Recently, logic-based abduction has provided the theoretical ground for the application fields of knowledge base and ontology debugging and repairing, in which inconsistent and incomplete information of the knowledge base or ontology is discovered and repaired (Section Related Work).

In this paper, we consider ontologies that are respresented by description logics (DLs), more specifically represented by TBoxes in the $\mathcal{EL}$ family, which consist of axioms such as $Carditis \sqsubseteq Fracture$, with the intended meaning that $Carditis$ is a $Fracture$, where $Carditis$ and $Fracture$ are *concepts* and the relationship is an *is-a* relation. (For detailed syntax see Section Preliminaries.) A set of such terminological axioms is a TBox. The $\mathcal{EL}$ family of description logics is highly relevant for the representation of lightweight ontologies. For instance, several of the major ontologies in the biomedical domain, e.g., SNOMED (http://www.ihtsdo.org/snomed-ct/) and Gene Ontology (Ashburner et al. 2000), can be represented in $\mathcal{EL}$ or small extensions thereof (Baader, Brandt, and Lutz 2005).

Defects in ontologies can take different forms (e.g. (Kalyanpur et al. 2006b)). The more interesting and severe defects are the modeling defects which require domain knowledge to detect and resolve, and semantic defects such as unsatisfiable concepts and inconsistent ontologies. In this paper we tackle a particular kind of modeling defects: defects in the is-a structure in ontologies. In addition to its importance for the correct modeling of a domain, the structural information in ontologies is also important in applications. Missing is-a structure leads to valid conclusions to be missed and therefore affects the quality of the application results. For instance, querying a ontology with missing is-a relations leads to incomplete results (according to the intended model) for the queries (e.g. (Lambrix and Liu 2013)). Debugging defects in ontologies consists of two phases, detection and repair. In this paper we assume that the detection phase has been performed and focus on the repairing phase. There are many approaches to detect missing is-a relations (see Section Related Work as well as Section Experiments). However, in general, these approaches do not detect *all* missing is-a relations and in several cases even only few. Therefore, we assume that we have obtained a set of missing is-a relations for a given ontology (but not necessarily all). In the case where our set of missing is-a relations contains *all* missing is-a relations, the repairing phase is easy. We just add all missing is-a relations to the ontology and a reasoner can compute all logical consequences. However, when the set of missing is-a relations does not contain all missing is-a relations - and this is the common case - there are different ways to repair the ontology. The easiest way is still to just add the missing is-a relations to the ontology. For instance, T in Figure 1 represents a small ontology inspired by Galen ontology (http://www.co-ode.org/galen/), that is relevant for our discussions. Assume that we have detected that Endocarditis $\sqsubseteq$ PathologicalPhenomenon and GranulomaProcess $\sqsubseteq$ NonNormalProcess are missing is-a relations (M in Figure 1). Obviously, adding these relations to the ontology will repair the missing is-a structure. However, there are other more interesting possibilities. For instance, adding Carditis $\sqsubseteq$ CardioVascularDisease and GranulomaProcess $\sqsubseteq$ PathologicalProcess also repairs the missing is-a structure. Further, these is-a relations are correct according to the domain and constitute new is-a relations (e.g. Carditis $\sqsubseteq$ CardioVascularDisease) that were not derivable from the ontology and not originally detected by the detection algorithm.[1]

We also note that from a logical point of view, adding Carditis $\sqsubseteq$ Fracture and GranulomaProcess $\sqsubseteq$ NonNormalProcess also repairs the missing is-a structure. However, from the point of view of the domain, this solution is not correct. Therefore, as for all approaches for debugging modeling defects, a domain expert needs to validate the logical solutions.

The above example shows that the framework of TBox abduction defined in (Elsenbroich, Kutz, and Sattler 2006) catches the basic semantics of repairing is-a relations. Let T denote the current ontology based on a certain formalism. The set of identified missing is-a relations M (atomic concept subsumptions) represents the manifestation. To repair the ontology, the ontology should be extended with a set S of atomic concept subsumptions (repair) such that the extended ontology is consistent and the missing is-a relations in M are derivable from the extended ontology. That is, $T \cup S \models M$ holds.

However, there are several properties of ontology repairing of missing is-a relations which distinguish themselves from the classic abduction framework. We summarize them as P1 and P2, and give the intuition behind them.

P1: Oracle function Or instead of hypothesis H.

In the classic abduction framework there is a hypothesis H from which the solution S is chosen such that $S \subseteq H$ holds. The corresponding component is the set of atomic concept subsumptions that should be correct according to the domain. In general, this set is not known beforehand. In the repairing scenario, a domain expert decides whether an atomic concept subsumption is correct according to the domain, and can return $true$ or $false$ like an oracle. Consequently, we formulate this function as Or that when given an atomic concept subsumption, returns $true$ or $false$. It is then required that for every atomic concept subsumption $s \in S$, we have that $Or(s) = true$.

P2: Informativeness as one of the preference criteria.

Ontology repairing of missing is-a relations follows different preference criteria from the logic-based abduction framework, in the sense that a more *informative* solution is preferred to a less informative one. Note that the informativeness is a measurement for how much information the added subsumptions (i.e. solution S) can derive. (See Definition 2 for the precise formulation.) This is in contrast to the criteria of minimality (e.g. subset minimality, cardinality minimality) from the abduction framework. In principle this difference on the preference stems from the original purpose of the two formalisms. The abduction framework is often used for diagnostic scenarios, thus the essential goal is to confine the cause of the problem as small as possible. Whilst for ontology repairing, the goal is to add more subsumptions to enrich the ontology. As long as the added rules are correct, a more informative repairing means more enrichment to the ontology. However, there are technical difficulties in finding the most informative solution as such. A brute-force method to create a most informative solution is to check for each pair of atomic concepts A and B, whether $Or(A \sqsubseteq B) = true$ and if so, add $A \sqsubseteq B$ to the ontology. In practice, for large ontologies this is infeasible. Therefore, it is not clear how to *generate* such a solution due to the missing hypothesis H. Further, we might obtain a solution with redundancy.

For this purpose, we would like to add another minimality preference, namely subset minimality to the informativeness preference. That is, we prefer a solution which is both semantically maximal (most informative) and subset minimal. Combining these two preferences drives us to three distinct interpretations, depending on what kind of priority we assign for the single preferences. The first interpretation (maxmin) implies higher priority for semantic maximality than subset minimality and thus favors semantically maximal solutions with no redundancy. A second interpretation (minmax) implies a higher priority on subset minimality than the semantic maximality, and thus favors solutions without redundancy which are the most informative solutions within these solutions without redundancy. In the third interpretation (skyline) we treat both preferences equally and the chosen solution is such that there does not exist another solution which is preferable on both criteria.

In this paper we focus on the formalization of the problems and conduct complexity analysis on the decision problems regarding the various preference criteria for $\mathcal{EL}^{++}$ ontologies. We prove the complexity results on all the decision problems (see Table 2) and obtain interesting findings. While it is not surprising that with either of the single preferences of subset minimality and semantic maximality, the complexity remains the same as the case without any preference (NP-complete), it is interesting to observe that combining the two preferences yields different complexity results. The combinations maxmin and skyline do not increase the complexity, while for minmax the complexity is higher which is at the second level of polynomial hierarchy. The intuition behind that can be explained informally as follows: for maxmin and skyline, the checking of both preference criteria can be conducted sequentially, while for minmax it is not possible. The complexity results provide a guideline on the choosing of suitable preference criteria for designing repairing algorithms in practice. As a result, the final part of the paper is dedicated to a concrete algorithm for finding *one* skyline optimal solution, together with a system based on the algorithm as well as experiments.

The contributions of this paper are the following.

- We formalize the repairing of the missing is-a structure in an ontology as a generalized version of the TBox abduction problem (GTAP).
- We present complexity results for the existence, relevance and necessity decision problems for GTAP in $\mathcal{EL}^{++}$ with and without the preference relations subset minimality and semantic maximality as well as three ways of combining these (maxmin, minmax, skyline). Subset minimality is a

[1] Therefore, the approach in this paper can also be seen as a detection method that takes already found missing is-a relations as input.

C = { GranulomaProcess, CardioVascularDisease, PathologicalPhenomenon, Fracture, Endocarditis, Carditis, InflammationProcess, PathologicalProcess, NonNormalProcess}

T = { CardioVascularDisease $\sqsubseteq$ PathologicalPhenomenon, Fracture $\sqsubseteq$ PathologicalPhenomenon, $\exists$hasAssociatedProcess.PathologicalProcess $\sqsubseteq$ PathologicalPhenomenon, Endocarditis $\sqsubseteq$ Carditis, Endocarditis $\sqsubseteq$ $\exists$hasAssociatedProcess.InflammationProcess, PathologicalProcess $\sqsubseteq$ NonNormalProcess }

M = { Endocarditis $\sqsubseteq$ PathologicalPhenomenon, GranulomaProcess $\sqsubseteq$ NonNormalProcess }

The following is-a relations are correct according to the domain, i.e., Or returns $true$ for:
GranulomaProcess $\sqsubseteq$ InflammationProcess, GranulomaProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ NonNormalProcess, CardioVascularDisease $\sqsubseteq$ PathologicalPhenomenon, Fracture $\sqsubseteq$ PathologicalPhenomenon, Endocarditis $\sqsubseteq$ PathologicalPhenomenon, Endocarditis $\sqsubseteq$ Carditis, Endocarditis $\sqsubseteq$ CardioVascularDisease, Carditis $\sqsubseteq$ PathologicalPhenomenon, Carditis $\sqsubseteq$ CardioVascularDisease, InflammationProcess $\sqsubseteq$ PathologicalProcess, InflammationProcess $\sqsubseteq$ NonNormalProcess, PathologicalProcess $\sqsubseteq$ NonNormalProcess.

Let $\mathcal{P}$ = GTAP(T, C, Or, M).

Figure 1: Small $\mathcal{EL}$ example.

preference criterion that is often used in abductive reasoning problems. Semantic maximality is a new criterion that is important for GTAP.

- We provide algorithms for finding a skyline optimal solution to GTAP in $\mathcal{EL}$ and $\mathcal{EL}^{++}$. Although in theory, maxmin optimal solutions are normally preferred, in practice, they cannot be guaranteed and skyline optimal solutions are the best we can do.
- We provide a system and show its usefulness through experiments.

Preliminaries

Proposition logic and Horn theory

We assume a finite propositional language built from a set $V = \{v_1, \ldots, v_n\}$ of atoms and the usual Boolean connectives. A clause is a disjunction $\lambda = \bigvee_{v_i \in Pos(\lambda)} v_i \vee \bigvee_{v_i \in Neg(\lambda)} \neg v_i$ where $Pos(\lambda)$ and $Neg(\lambda)$ are the sets of atoms which appear positively and negatively in λ and $Pos(\lambda) \cap Neg(\lambda) = \emptyset$. We say that a clause λ is *Horn* if $|Pos(\lambda)| \leq 1$. A Horn theory is a set of Horn clauses.

The description logics $\mathcal{EL}^{++}$ and $\mathcal{EL}$

Concept descriptions are constructed inductively from a set N_C of atomic concepts and a set N_R of atomic roles and (possibly) a set N_I of individual names. The concept constructors are the top concept $\top$, bottom concept $\bot$, nominals, conjunction, and existential restriction, and a restricted form of concrete domains. In this paper, we consider the version of $\mathcal{EL}^{++}$ without concrete domains. Note that this simplification does not affect the complexity results presented later on. For the syntax of the different constructors see Table 1. An interpretation $\mathcal{I}$ consists of a non-empty set $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ which assigns to each atomic concept $A \in N_C$ a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, to each atomic role $r \in N_R$ a relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to each individual name $a \in N_I$ an individual $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The interpretation function is straightforwardly extended to complex concepts. An $\mathcal{EL}^{++}$ TBox (named CBox in (Baader, Brandt, and Lutz 2005)) is a finite set of *general concept inclusions* (GCIs) and *role inclusions* (RIs) whose syntax can be found in the lower part of Table 1. Note that a finite set of GCIs is called a *general TBox*. An interpretation $\mathcal{I}$ is a *model* of a TBox T if for each GCI and RI in T, the conditions given in the third column of Table 1 are satisfied. $\mathcal{EL}$ has the restricted form of $\mathcal{EL}^{++}$ which allows for concept constructors of top concept $\top$, conjunction and existential restriction. An $\mathcal{EL}$ TBox contains only GCIs. The main reasoning task for description logics is subsumption in which the problem is to decide for a TBox T and concepts C and D whether $T \models C \sqsubseteq D$. Subsumption in $\mathcal{EL}^{++}$ is polynomial even w.r.t. general TBoxes (Baader, Brandt, and Lutz 2005).

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
RI	$r_1 \circ \ldots \circ r_k \sqsubseteq r$	$r_1^{\mathcal{I}} \circ \ldots \circ r_k^{\mathcal{I}} \subseteq r^{\mathcal{I}}$

Table 1: $\mathcal{EL}^{++}$ Syntax and Semantics

We note that every Horn theory can be represented by a general $\mathcal{EL}^{++}$ TBox (Bienvenu 2008).

Computational Complexity

We recall some basic definitions from computational complexity (cf. (Papadimitriou 1994)). The class P comprises all problems which can be decided in polynomial time by a deterministic Turing machine. The class NP contains all problems which can be decided in polynomial time by a nondeterministic Turing machine. The class co-NP is defined to be the set of all problems whose complement belongs to NP. The class $\Sigma_2^P = \mathrm{NP}^{\mathrm{NP}}$ consists of those problems which can be decided in polynomial time by a non-deterministic Turing machine which can query an NP oracle. The class Π_2^P comprises all problems whose complement is in Σ_2^P.

A quantified Boolean formula (QBF) is a sentence of the form $Q_1x_1 \ldots Q_nx_nE, n \geq 0$, where E is a propositional formula whose variables are from $x_1, \ldots, x_n$ and where each $Q_i, 1 \leq i \leq n$, is one of the quantifiers $\forall$, $\exists$ ranging over $\{true, false\}$. Such a formula is said to have a quantifier alternation for Q_1 and for each $Q_i, i > 1$, such that $Q_i \neq Q_{i-1}$. The set of valid QBFs with k quantifier alternations and $Q_1 = \exists$ (resp. $Q_1 = \forall$) is denoted by $\text{QBF}_{k,\exists}$ (resp. $\text{QBF}_{k,\forall}$). It is well-known that deciding whether a QBF Φ satisfies $\Phi \in \text{QBF}_{k,\exists}$ (resp. $\Phi \in \text{QBF}_{k,\forall}$) is Σ_k^P-complete (resp. Π_k^P-complete).

Abduction Framework

In the following we explain how the problem of finding possible ways to repair the missing is-a structure in a ontology is formalized as a generalized version of the TBox abduction problem as defined in (Elsenbroich, Kutz, and Sattler 2006). We assume that our ontology is represented using a TBox T in $\mathcal{EL}^{++}$. The identified missing is-a relations are then represented by a set M of atomic concept subsumptions. To repair the ontology, the ontology should be extended with a set S of atomic concept subsumptions (repair) such that the extended ontology is consistent and the missing is-a relations are derivable from the extended ontology. However, the added atomic concept subsumptions should be correct according to the domain. In general, the set of all atomic concept subsumptions that are correct according to the domain are not known beforehand. Indeed, if this set were given then we would only have to add this to the ontology. The common case, however, is that we do not have this set, but instead can rely on a domain expert that can decide whether an atomic concept subsumption is correct according to the domain. In our formalization the domain expert is represented by an oracle Or that when given an atomic concept subsumption, returns true or false. It is then required that for every atomic concept subsumption $s \in S$, we have that $Or(s) = true$. The following definition formalizes this.

Definition 1 *(GENERALIZED TBOX ABDUCTION) Let T be a TBox in $\mathcal{EL}^{++}$ and C be the set of all atomic concepts in T. Let $M = \{A_i \sqsubseteq B_i \mid A_i, B_i \in C\}$ be a finite set of TBox assertions. Let $Or : \{C_i \sqsubseteq D_i \mid C_i, D_i \in C\} \rightarrow \{true, false\}$. A solution to the generalized TBox abduction problem (GTAP) (T, C, Or, M) is any finite set $S = \{E_i \sqsubseteq F_i \mid E_i, F_i \in C \ \wedge \ Or(E_i \sqsubseteq F_i) = true\}$ of TBox assertions, such that $T \cup S$ is consistent and $T \cup S \models M$. The set of all such solutions is denoted as $\mathcal{S}(T, C, Or, M)$.*

As noted before, in the classic abduction problem there is usually no oracle Or, but a set of abduciles H (e.g. (Eiter and Gottlob 1995)) that restricts the solution space. A major difference is that H is usually given, and finding solutions can therefore start from H. In GTAP on the other hand this is not possible, but (partial) solutions are validated using Or.

As an example, consider GTAP $\mathcal{P}$ as defined in Figure 1. Then {Carditis $\sqsubseteq$ CardioVascularDisease, InflammationProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ InflammationProcess} is a solution for $\mathcal{P}$. Another solution is {Carditis $\sqsubseteq$ CardioVascularDisease, GranulomaProcess $\sqsubseteq$ PathologicalProcess} as shown in Section Introduction.

There can be many solutions for a GTAP and, as explained in Section Introduction, not all solutions are equally interesting. Therefore, we propose two preference criteria on the solutions as well as different ways to combine them. The first criterion is a criterion that is not used in other abduction problems, but that is particularly important for GTAP. In GTAP it is important to find solutions that add to the ontology as much information as possible that is correct according to the domain. Therefore, the first criterion prefers solutions that imply more information.

Definition 2 *(MORE INFORMATIVE) Let S and S' be two solutions to the GTAP (T, C, Or, M). S is said to be* more informative *than S' iff $T \cup S \models S'$ and $T \cup S' \not\models S$.*

Further, we say that S is equally informative *as S' iff $T \cup S \models S'$ and $T \cup S' \models S$.*

Consider two solutions to $\mathcal{P}$, S_1 = {InflammationProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ InflammationProcess}[2] and S_2 = {InflammationProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ PathologicalProcess}. In this case solution S_1 is more informative than S_2.

Definition 3 *(SEMANTIC MAXIMALITY) A solution S to the GTAP (T, C, Or, M) is said to be semantically maximal iff there is no solution S' which is more informative than S. The set of all semantically maximal solutions is denoted as $\mathcal{S}^{max}(T, C, Or, M)$.*

An example of a semantically maximal solution to $\mathcal{P}$ is {InflammationProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ InflammationProcess, Carditis $\sqsubseteq$ CardioVascularDisease}.

The second criterion is a classical criterion in abduction problems. It requires that no element in a solution is redundant.

Definition 4 *(SUBSET MINIMALITY) A solution S to the GTAP (T, C, Or, M) is said to be subset minimal iff there is no proper subset $S' \subsetneq S$ such that S' is a solution. The set of all subset minimal solutions is denoted as $\mathcal{S}_{min}(T, C, Or, M)$.*

An example of a subset minimal solution for $\mathcal{P}$ is {InflammationProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ InflammationProcess}. On the other hand, solution {Carditis $\sqsubseteq$ CardioVascularDisease, InflammationProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ InflammationProcess} is not subset minimal as it contains Carditis $\sqsubseteq$ CardioVascularDisease which is redundant for repairing the missing is-a relations.

[2] Observe that both missing is-a relations are derivable using S_1. GranulomaProcess $\sqsubseteq$ NonNormalProcess is derivable as GranulomaProcess $\sqsubseteq$ InflammationProcess (S_1), InflammationProcess $\sqsubseteq$ PathologicalProcess (S_1), and PathologicalProcess $\sqsubseteq$ NonNormalProcess (T). Endocarditis $\sqsubseteq$ PathologicalPhenomenon is derivable as Endocarditis $\sqsubseteq$ $\exists$hasAssociatedProcess.InflammationProcess (T), $\exists$hasAssociatedProcess.InflammationProcess $\sqsubseteq$ $\exists$hasAssociatedProcess.PathologicalProcess (S_1), and $\exists$hasAssociatedProcess.PathologicalProcess $\sqsubseteq$ PathologicalPhenomenon (T).

In practice, both of the above two criteria are desirable. We therefore define ways to combine these criteria depending on what kind of priority we assign for the single preferences.

Definition 5 *(COMBINING WITH PRIORITY FOR SEMANTIC MAXIMALITY) A solution S to the GTAP (T, C, Or, M) is said to be maxmin optimal iff S is semantically maximal and there does not exist another semantically maximal solution S' such that S' is a proper subset of S. The set of all maxmin optimal solutions is denoted as $\mathcal{S}^{\mathbf{max}}_{min}(T, C, Or, M)$.*

As an example, {InflammationProcess ⊑ PathologicalProcess, GranulomaProcess ⊑ InflammationProcess, Carditis ⊑ CardioVascularDisease} is a maxmin optimal solution for $\mathcal{P}$. The advantage of maxmin optimal solutions is that a maximal body of correct information is added to the ontology and without redundancy. For GTAP these are the most attractive solutions, but as mentioned before it is not clear how to generate such a solution.

Definition 6 *(COMBINING WITH PRIORITY FOR SUBSET MINIMALITY) A solution S to the GTAP (T, C, Or, M) is said to be minmax optimal iff S is subset minimal and there does not exist another subset minimal solution S' such that S' is more informative than S. The set of all minmax optimal solutions is denoted as $\mathcal{S}^{max}_{\mathbf{min}}(T, C, Or, M)$.*

As an example, {InflammationProcess ⊑ PathologicalProcess, GranulomaProcess ⊑ InflammationProcess} is a minmax optimal solution for $\mathcal{P}$. In practice, minmax optimal solutions ensure fewer is-a relations to be added, thus avoiding redundancy. This is desirable if the domain expert would prefer to look at as small solutions as possible. The disadvantage is that there may be correct relations that are not derivable when they are not included in the solution.

For the skyline interpretation, we consider the subset minimality and the semantic maximality as two dimensions for a solution S (see (Lambrix et al. 2013) for an explanation of how the definition satisfies the skyline interpretation).

Definition 7 *(SKYLINE OPTIMAL) A solution S to the GTAP (T, C, Or, M) is said to be skyline optimal iff there does not exist another solution S' such that S' is a proper subset of S and S' is equally informative as S. The set of all skyline optimal solutions is denoted as $\mathcal{S}^{max}_{min}(T, C, Or, M)$.*

All subset minimal, minmax optimal and maxmin optimal solutions are also skyline optimal solutions. However, there are semantically maximal solutions that are not skyline optimal. For example, {InflammationProcess ⊑ PathologicalProcess, GranulomaProcess ⊑ InflammationProcess, Carditis ⊑ CardioVascularDisease, Endocarditis ⊑ CardioVascularDisease} is a semantically maximal solution for $\mathcal{P}$, but it is not skyline optimal as its subset {InflammationProcess ⊑ PathologicalProcess, GranulomaProcess ⊑ InflammationProcess, Carditis ⊑ CardioVascularDisease} is equally informative. There also exist skyline optimal solutions that are not subset minimal solutions. For instance, {InflammationProcess ⊑ PathologicalProcess, GranulomaProcess ⊑ InflammationProcess, Carditis ⊑ CardioVascularDisease} is a skyline optimal solution that is not subset minimal as removing Carditis ⊑ CardioVascularDisease would still yield a solution (although not as informative). Skyline optimal is a relaxed criterion. It requires subset minimality for some level of informativeness. Although maxmin solutions are preferred, in practice, it is not clear how to generate a maxmin solution, except for a brute-force method that would query the oracle with, for larger ontologies, unfeasibly many questions. Therefore, a skyline solution is the next best thing and, in the case solutions exist, it is easy to generate *a* skyline optimal solution. However, the difficulty lies in reaching an as high level of informativeness as possible.

Further, in addition to finding solutions, traditionally, there are three main decision problems for logic-based abduction: existence, relevance and necessity.

Definition 8 *Given a GTAP (T, C, Or, M) we define the following decision problems:*

Existence *$\mathcal{S}(T, C, Or, M) \neq \emptyset$?*

Relevance *Given ψ, does a solution $S \in \mathcal{S}(T, C, Or, M)$ exist such that $\psi \in S$?*

Necessity *Given ψ, do all the solutions in $\mathcal{S}(T, C, Or, M)$ contain ψ?*

If we replace $\mathcal{S}$ in Definition 8 with $\mathcal{S}_{min}$, $\mathcal{S}^{max}$, $\mathcal{S}^{max}_{\mathbf{min}}$ $\mathcal{S}^{\mathbf{max}}_{min}$ and $\mathcal{S}^{max}_{min}$, respectively, we obtain the GTAP decision problems under the criteria of subset minimality, semantic maximality and the combinations.

Dispensability Given ψ, does a solution $S \in \mathcal{S}(T, C, Or, M)$ exist such that $\psi \notin S$?

For convenience in Section Complexity Results we primarily deal with dispensability rather than with necessity. Results for necessity are easy corollaries to our results on dispensability.

Complexity Results

In this section, we present complexity results for deciding the existence and relevance of GTAP under several preference criteria for both $\mathcal{EL}$ and $\mathcal{EL}^{++}$. The summary of the results is shown in Table 2.

Since it holds that every definite Horn theory can be represented by a general $\mathcal{EL}$ TBox and every Horn theory can be represented by a general $\mathcal{EL}^{++}$ TBox (Bienvenu 2008), some existing complexity results on the abduction of Horn theory can be adapted here for the case of general existence and subset minimality case. Note that this applies to the hardness proofs.

Complexity - $\mathcal{EL}^{++}$

General Case

Theorem 1 *To decide if $\mathcal{S}(T, C, Or, M) \neq \emptyset$ for a given GTAP (T, C, Or, M) is NP-complete.*

Proof. The entailment problem of $\mathcal{EL}^{++}$ is tractable (Baader, Brandt, and Lutz 2005). Therefore the membership in NP follows.

NP-hardness of this problem is shown by a transformation from well-known satisfiability problem (SAT), cf. (Garey

	$\mathcal{EL}$			$\mathcal{EL}^{++}$		
Decision problems	Existence	Relevance	Necessity	Existence	Relevance	Necessity
General	in P	in P	in P	NP-complete	NP-complete	co-NP-complete
Subset Minimality	in P	NP-complete	in P	NP-complete	NP-complete	co-NP-complete
Semantic Maximality	in P	in P	in P	NP-complete	NP-complete	co-NP-complete
Minmax	in P	NP-complete	in P	NP-complete	Σ_2^P-complete	Π_2^P-complete
Maxmin	in P	in P	in P	NP-complete	NP-complete	co-NP-complete
Skyline	in P	NP-complete	in P	NP-complete	NP-complete	co-NP-complete

Table 2: Complexity Results of GTAP

and Johnson 1979). Let $Cl = \{Cl_1, \ldots, Cl_m\}$ be a set of propositional clauses on $X = \{x_1, \ldots, x_n\}$. Let $X' = \{x'_1, \ldots, x'_n\}$, $G = \{g_1, \ldots, g_m\}$, $R = \{r_1, \ldots, r_n\}$ be sets of new concepts and c be a new concept. Then, the GTAP (T, C, Or, M) is constructed as follows.

Note that in order to simplify the presentation, for the definition of the oracle, we write Or as a set containing the subsumptions that are $true$ according to the oracle. We also apply this simplification in the other proofs of the paper.

$$\begin{array}{l} C = X \cup X' \cup G \cup R \cup c \\ M = \{c \sqsubseteq r_i : 1 \leq i \leq n,\ \ c \sqsubseteq g_j : 1 \leq j \leq m\} \\ Or = \{c \sqsubseteq x_i : 1 \leq i \leq n,\ \ c \sqsubseteq x'_i : 1 \leq j \leq n\} \\ T = \{x_i \sqcap x'_i \sqsubseteq \bot, x_i \sqsubseteq r_i, x'_i \sqsubseteq r_i : 1 \leq i \leq n\} \cup \{c \sqsubseteq \top, \top \sqsubseteq c\} \\ \qquad \bigcup_{i=1}^{m} (\{x_j \sqsubseteq g_i : x_j \in Cl_i\} \cup \{x'_j \sqsubseteq g_i : \neg x_j \in Cl_i\}) \end{array}$$

Next we prove that Cl is satisfiable iff (T, C, Or, M) has a solution. We first observe that for each $S \in \mathcal{S}(T, C, Or, M)$, either $c \sqsubseteq x_i \in S$ or $c \sqsubseteq x'_i \in S$ (but not both) must hold, for $1 \leq i \leq n$, since otherwise $T \cup S \not\models c \sqsubseteq r_i$.

Assume Cl is satisfiable. Let ψ be the truth assignment such that $\psi(Cl)$ is $true$. Define the solution S as

$$\begin{array}{rcl} S & = & \{c \sqsubseteq x_i : \psi(x_i) = true, 1 \leq i \leq n\} \cup \\ & & \{c \sqsubseteq x'_i : \psi(x_i) = false, 1 \leq i \leq n\} \end{array}$$

Clearly $T \cup S \models c \sqsubseteq r_1 \wedge \ldots \wedge c \sqsubseteq r_n$. Moreover, because for every $Cl_i (1 \leq i \leq m)$ $\psi(Cl_i)$ is $true$, we have $T \cup S \models c \sqsubseteq g_1 \wedge \ldots \wedge c \sqsubseteq g_m$. Therefore $T \cup S \models M$ holds.

Consider Cl is not satisfiable. For a solution S, either x_i or x'_i must exist in S. Since there does not exist any truth assignment such that $\psi(Cl)$ is $true$, there does not exist such S such that $T \cup S \models c \sqsubseteq g_1 \wedge \ldots \wedge c \sqsubseteq g_m$. Therefore $\mathcal{S}(T, C, Or, M) = \emptyset$.

Theorem 2 *To decide if a given ψ is relevant for a given GTAP (T, C, Or, M) is NP-complete. To decide if a given ψ is dispensable for a given GTAP (T, C, Or, M) is NP-complete.*

Proof. Guess a solution S which contains ψ (resp. does not contain ψ). Since the checking if $S \in \mathcal{S}(T, C, Or, M)$ is in P, the membership in NP follows.

Hardness can be proven by a slight modification of the reduction for the existence problem in Theorem 1. Define the GTAP (T', C', Or', M') as

$$\begin{array}{rcl} C' & = & C \cup e \cup e' \\ M' & = & M \cup h \\ Or' & = & Or \cup \{c \sqsubseteq e, c \sqsubseteq e'\} \\ T' & = & T \setminus \{x_i \sqsubseteq r_i, x'_i \sqsubseteq r_i : 1 \leq i \leq n\} \cup \\ & & \{x_i \sqcap e \sqsubseteq r_i, x'_i \sqcap e \sqsubseteq r_i : 1 \leq i \leq n\} \cup \\ & & \{e' \sqsubseteq r_i : 1 \leq i \leq n,\ \ e' \sqsubseteq g_j : 1 \leq j \leq m\} \cup \\ & & \{e \sqcap e' \sqsubseteq \bot, e \sqsubseteq h, e' \sqsubseteq h\} \end{array}$$

where e, e', h are new concepts not occurring in C.

We show that Cl is satisfiable if and only if (T', C', Or', M') has a solution containing $c \sqsubseteq e$ and not containing $c \sqsubseteq e'$.

Assume Cl is satisfiable. Let ψ be the truth assignment such that $\psi(Cl)$ is $true$. Define the solution S as

$$\begin{array}{rcl} S & = & \{c \sqsubseteq x_i : \psi(x_i) = true, 1 \leq i \leq n\} \cup \\ & & \{c \sqsubseteq x'_i : \psi(x_i) = false, 1 \leq i \leq n\} \cup \{c \sqsubseteq e\} \end{array}$$

Then $T' \cup S \models M'$ holds. Note that one and only one of $c \sqsubseteq e$ and $c \sqsubseteq e'$ is in any solution to (T', C', Or', M'). Therefore, $c \sqsubseteq e' \notin S$ holds.

Assume Cl is not satisfiable. Then the solution S is $\{c \sqsubseteq e'\}$. Clearly $c \sqsubseteq e \notin S$ holds. This concludes the proof.

Subset Minimality

Theorem 3 *To decide if $\mathcal{S}_{min}(T, C, Or, M) \neq \emptyset$ for a given GTAP (T, C, Or, M) is NP-complete.*

Proof. We show that the problem is equivalent to the existence problem in general case. That is, $\mathcal{S}_{min}(T, C, Or, M) \neq \emptyset$ iff $\mathcal{S}(T, C, Or, M) \neq \emptyset$. The 'only if' direction is trivial. Now we prove the 'if' direction. We show that if there is a solution $S \in \mathcal{S}(T, C, Or, M)$, then there is a solution $S' \in \mathcal{S}_{min}(T, C, Or, M)$ and $S' \subseteq S$. If S is subset minimal, then $S' = S$. Otherwise, let $\mathcal{W}$ be the set of all solutions S'' such that $S'' \subset S$. Since the empty set is not a solution, there exists an $S' \in \mathcal{W}$, such that $\forall P \in \mathcal{W}$, $P \not\subset S'$ holds. Clearly S' is a subset minimal solution.

Theorem 4 *To decide if a given ψ is min-relevant for a given GTAP (T, C, Or, M) is NP-complete. To decide if a given ψ is min-dispensable for a given GTAP (T, C, Or, M) is NP-complete.*

Proof. Membership: guess a set S which contains (resp. does not contain) ψ. Note that $S \in \mathcal{S}_{min}(T, C, Or, M)$ iff $S \in \mathcal{S}(T, C, Or, M)$ and $\{\forall h \in S : S \setminus h \notin$

$\mathcal{S}(T, C, Or, M)\}$ holds. This is due to the monotonicity of $\models$ in $\mathcal{EL}^{++}$. Clearly the checking is in P, hence the membership in NP follows.

Hardness under the restrictions follows immediately by Theorem 2.

Semantic Maximality

Theorem 5 *To decide if $\mathcal{S}^{max}(T, C, Or, M) \neq \emptyset$ for a given GTAP (T, C, Or, M) is NP-complete.*

Proof. The proof is analogous to that of Theorem 3: we show that the problem is equivalent to the existence problem of the general case. That is, $\mathcal{S}^{max}(T, C, Or, M) \neq \emptyset$ iff $\mathcal{S}(T, C, Or, M) \neq \emptyset$. The 'only if' direction is trivial. Now we prove the 'if' direction. We show that if there is a solution $S \in \mathcal{S}(T, C, Or, M)$, then there is a solution $S' \in \mathcal{S}^{max}(T, C, Or, M)$ and $S \subseteq S'$. Let $\mathcal{W}$ be the set of all solutions S'' that $S \subseteq S''$. Then there exists $S' \in \mathcal{W}$, such that $\forall P \in \mathcal{W}$, $S' \not\subset P$ holds. It is easy to show that S' is semantically maximal. Assume the opposite. There is another solution S_1 which is more informative than S'. That is, there is a ψ such that $T \cup S_1 \models S' \cup \psi$ and $T \cup S' \not\models \psi$. Then $S' \cup S_1$ should be a solution and it is a superset of S'. $\Rightarrow$ Contradiction.

Theorem 6 *To decide if a given ψ is max-relevant for a given GTAP (T, C, Or, M) is NP-complete. To decide if a given ψ is max-dispensable for a given GTAP (T, C, Or, M) is NP-complete.*

Proof. Membership: guess a set S which contains (resp. does not contain) ψ. $S \in \mathcal{S}^{max}(T, C, Or, M)$ iff $S \in \mathcal{S}(T, C, Or, M)$ and $\forall h \in Ors.t.T \cup S \not\models h : T \cup S \cup h \models M\}$ holds. This is due to the monotonicity of $\models$ in $\mathcal{EL}^{++}$. The checking can be done in polynomial time since the number of possible TBox assertions is polynomial to C. Hence the membership follows.

Hardness under the restrictions follows immediately by Theorem 2.

Skyline Due to the fact that the set of skyline optimal solutions contains all subset minimal solutions, the existential problem follows trivially. That is, if there exists a subset minimal solution, then there exists a skyline optimal solution.

Theorem 7 *To decide if $\mathcal{S}^{max}_{min}(T, C, Or, M) \neq \emptyset$ for a given GTAP (T, C, Or, M) is NP-complete.*

Theorem 8 *To decide if a given ψ is skyline-relevant for a given GTAP (T, C, Or, M) is NP-complete. To decide if a given ψ is skyline-dispensable for a given GTAP (T, C, Or, M) is NP-complete.*

Proof. Membership: guess a set S which contains (resp. does not contain) ψ. Note that $S \in \mathcal{S}^{max}_{min}(T, C, Or, M)$ iff $S \in \mathcal{S}(T, C, Or, M)$ and $\{\forall h \in S : T \cup (S \setminus h) \not\models S\}$ holds. This is due to the monotonicity of $\models$ in $\mathcal{EL}^{++}$. Clearly the checking is in P, hence the membership in NP follows. Hardness under the restrictions follows immediately by Theorem 2.

Maxmin

Theorem 9 *To decide if $\mathcal{S}^{\mathbf{max}}_{min}(T, C, Or, M) \neq \emptyset$ for a given GTAP (T, C, Or, M) is NP-complete.*

Proof. Again, we show that the problem is equivalent to the existence problem of the general case. Since the existence problem of $\mathcal{S}^{max}(T, C, Or, M)$ is shown to be equivalent to the general case, there exists $\mathcal{S}^{max}(T, C, Or, M)$. Since $\mathcal{S}^{\mathbf{max}}_{min}(T, C, Or, M) \subseteq \mathcal{S}^{max}(T, C, Or, M)$ holds, we need to remove from $\mathcal{S}^{max}(T, C, Or, M)$ those solutions $\{S | \exists S', s.t. S' \subset S : T \cup S' \models S\}$. Given a maximal solution S, we call such an S' the witness of S. Note that if $S \in \mathcal{S}^{max}(T, C, Or, M)$, then all the witnesses of S as defined above are also in $\mathcal{S}^{max}(T, C, Or, M)$. Therefore, during the removing process, if S is removed, S must have a witness S' and S' is still in $\mathcal{S}^{max}(T, C, Or, M)$. As a result, there will be at least one solution remaining in $\mathcal{S}^{max}(T, C, Or, M)$ after the removal process. This concludes the proof.

Theorem 10 *To decide if a given ψ is maxmin-relevant for a given GTAP (T, C, Or, M) is NP-complete. To decide if a given ψ is maxmin-dispensable for a given GTAP (T, C, Or, M) is NP-complete.*

Proof. Membership: guess a set S which contains (resp. does not contain) ψ. Note that $S \in \mathcal{S}^{\mathbf{max}}_{min}(T, C, Or, M)$ iff $S \in \mathcal{S}^{max}(T, C, Or, M)$ and $\{\forall h \in S : T \cup (S \setminus h) \not\models S\}$ holds. To check whether $S \in \mathcal{S}^{max}(T, C, Or, M)$ is feasible in polynomial time as shown in Theorem 6. The minimality check is also feasible in polynomial time as shown in Theorem 8, hence the membership in NP follows. Hardness under the restrictions follows immediately by Theorem 2.

Minmax

Theorem 11 *To decide if $\mathcal{S}^{max}_{\mathbf{min}}(T, C, Or, M) \neq \emptyset$ for a given GTAP (T, C, Or, M) is NP-complete.*

Proof. We show that the problem is equivalent to the existence problem of the general case. That is, $\mathcal{S}^{max}_{\mathbf{min}}(T, C, Or, M) \neq \emptyset$ iff $\mathcal{S}(T, C, Or, M) \neq \emptyset$. If there is a solution $S \in \mathcal{S}(T, C, Or, M)$, then from Theorem 3 there is a solution which is subset minimal. Let $\mathcal{W}$ be the set of all the subset minimal solutions. Then we remove from $\mathcal{W}$ the solutions which are less informative, in the sense that if there is $S', S'' \in \mathcal{W}$ such that S' is more informative than S'', then S'' is removed. Since the relation *more informative* is transitive, the removal process is confluent. Then there exists a unique non-empty set $\mathcal{W}' \subseteq \mathcal{W}$, such that no solution is more informative than another. It is obvious that $\mathcal{W}'$ is $\mathcal{S}^{max}_{\mathbf{min}}(T, C, Or, M)$.

Theorem 12 *To decide if a given ψ is minmax-relevant for a given GTAP (T, C, Or, M) is Σ_2^P-complete. To decide if a given ψ is minmax-dispensable for a given GTAP (T, C, Or, M) is Σ_2^P-complete.*

Proof. Membership can be shown by first guessing a solution S containing (resp. not containing) ψ, then verifying if $S \in \mathcal{S}^{max}_{\mathbf{min}}(T, C, Or, M)$. That is, to check whether there does not exist a subset minimal solution which is more informative than S. The check can be done by a co-NP oracle, since checking that there does exist such a solution can be done in NP (we guess a solution S'. Checking S' is subset minimal and S' is more informative than S can be done in polynomial time). Therefore, the membership in Σ_2^P follows.

Σ_2^P-hardness of this problem is shown by a transformation from deciding $\Phi \in \mathrm{QBF}_{2,\exists}$. Let Φ without loss of generality be a QBF $\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m E$. Let E be in disjunctive normal form $D_1 \vee \ldots \vee D_l$ where $D_i (1 \leq i \leq l)$ is a conjunction of literals. Let $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_m\}$, $X' = \{x'_1, \ldots, x'_n\}$, and $Y' = \{y'_1, \ldots, y'_m\}$. Let further $G = \{g_1, \ldots, g_m\}$, $R = \{r_1, \ldots, r_n\}$ be sets of new concepts and h, e, e', c be new concepts. Then, the GTAP (T, C, Or, M) is constructed as follows.

$$\begin{array}{rcl}
C & = & X \cup X' \cup Y \cup Y' \cup G \cup R \cup h \cup c \cup e \cup e' \\
M & = & \{c \sqsubseteq h\} \\
Or & = & \{c \sqsubseteq e, c \sqsubseteq e', c \sqsubseteq x_i : 1 \leq i \leq n, \\
 & & c \sqsubseteq x'_i : 1 \leq i \leq n, c \sqsubseteq y_j : 1 \leq j \leq m, \\
 & & c \sqsubseteq y'_j : 1 \leq j \leq m\}
\end{array}$$

$$\begin{array}{rcl}
T & = & \{c \sqsubseteq \top, \top \sqsubseteq c\} \\
 & & \cup \{x_i \sqcap x'_i \sqsubseteq \bot, x_i \sqcap e \sqsubseteq r_i, x'_i \sqcap e \sqsubseteq r_i : 1 \leq i \leq n\} \\
 & & \cup \{r_1 \sqcap \ldots \sqcap r_n \sqsubseteq h\} \\
 & & \cup \{y_i \sqcap y'_i \sqsubseteq \bot, y_i \sqcap e' \sqsubseteq g_i, y'_i \sqcap e' \sqsubseteq g_i : 1 \leq i \leq m\} \\
 & & \cup \{g_1 \sqcap \ldots \sqcap g_m \sqsubseteq e\} \cup T' \cup T''
\end{array}$$

$$\begin{array}{rcl}
T' & = & \bigcup_{i=1}^{l} \bigcup_{j=1}^{s} (\{y_{i_1} \sqcap \ldots \sqcap y_{i_p} \sqcap y'_{i_{p+1}} \sqcap \ldots \sqcap y'_{i_q} \\
 & & \sqcap_{k=1,k\neq j}^{s} x_{i_k} \sqcap_{k=s+1}^{t} x'_{i_k} \quad \sqsubseteq \quad x'_{i_j} : \\
 & & D_i = y_{i_1} \wedge \ldots \wedge y_{i_p} \wedge \neg y_{i_{p+1}} \wedge \ldots \wedge \neg y_{i_q} \\
 & & \wedge x_{i_1} \wedge \ldots \wedge x_{i_s} \wedge \neg x_{i_{s+1}} \wedge \ldots \wedge \neg x_{i_t}\}) \\
T'' & = & \bigcup_{i=1}^{l} \bigcup_{j=s+1}^{t} (\{y_{i_1} \sqcap \ldots \sqcap y_{i_p} \sqcap y'_{i_{p+1}} \sqcap \ldots \sqcap y'_{i_q} \\
 & & \sqcap_{k=1}^{s} x_{i_k} \sqcap_{k=s+1,k\neq j}^{t} x'_{i_k} \quad \sqsubseteq \quad x_{i_j} : \\
 & & D_i = y_{i_1} \wedge \ldots \wedge y_{i_p} \wedge \neg y_{i_{p+1}} \wedge \ldots \wedge \neg y_{i_q} \\
 & & \wedge x_{i_1} \wedge \ldots \wedge x_{i_s} \wedge \neg x_{i_{s+1}} \wedge \ldots \wedge \neg x_{i_t}\})
\end{array}$$

Intuitively, for each disjunct D_i in E, for each x literal in D_i, T' and T'' consists of a subsumption where the negated form of x is at the right hand side. More precisely, if x is of the form x_i, then x'_i occurs at the right hand side; if x is of the form $\neg x_i$, then x_i occurs at the right hand side. For instance, assume $D_i = y_1 \wedge \neg y_2 \wedge x_1 \wedge \neg x_2$. Then T' consists of the subsumption $y_1 \sqcap y'_2 \sqcap x'_2 \sqsubseteq x'_1$, and T'' consists of $y_1 \sqcap y'_2 \sqcap x_1 \sqsubseteq x_2$.

Note that T is consistent and that (T, C, Or, M) is constructible in polynomial time. We show that $\Phi \in \mathrm{QBF}_{2,\exists}$ holds iff $(c \sqsubseteq e) \in S$ (resp. $(c \sqsubseteq e') \notin S$) such that $S \in \mathcal{S}_{\mathbf{min}}^{max}(T, C, Or, M)$.

"Only if": Assume $\Phi \in \mathrm{QBF}_{2,\exists}$ holds. Hence, there exists a truth assignment $\phi(X)$ such that $\forall y_1 \ldots \forall y_m E_\phi(X) \in \mathrm{QBF}_{1,\forall}$ holds. Define the solution S as $S = \{c \sqsubseteq x_i : \phi(x_i) = true, 1 \leq i \leq n\} \cup \{c \sqsubseteq x'_i : \phi(x_i) = false, 1 \leq i \leq n\} \cup \{c \sqsubseteq e\}$.

Clearly $T \cup S \models M$. Moreover, S is subset minimal. Next we show there is no other subset minimal solution which is more informative than S. Other than ϕ, there are $2^n - 1$ possible truth assignments over X. For each such truth assignment ψ, we can obtain the corresponding solution S', analogously to the way obtaining S by replacing ϕ with ψ. Clearly every such S' is a subset minimal solution. However, it is obvious that $T \cup S' \not\models S$, since $S \neq S'$ and there is at least one variable x_i such that $\phi(x_i) \neq \psi(x_i)$.

Let μ be an arbitrary truth assignment over Y. Define S' as $S' = \{c \sqsubseteq y_i : \mu(y_i) = true, 1 \leq i \leq m\} \cup \{c \sqsubseteq y'_i : \mu(y_i) = false, 1 \leq i \leq m\} \cup \{c \sqsubseteq e'\}$. Clearly any other subset minimal solution S'' which does not contain $c \sqsubseteq e$ must contain such an S'. Note that we do not fix S' since μ is arbitrary. To prove S is a minmax solution, we need to show that there does not exist such a subset minimal solution S'' such that $T \cup S'' \models S$ holds. In the following we show that for every such a possible solution S'', $T \cup S'' \cup S$ is inconsistent.

Since $\forall y_1 \ldots \forall y_m E_\phi(X) \in \mathrm{QBF}_{1,\forall}$ holds, there exists a disjunct $D_i \in E$, such that $D_{i\phi,\mu}(X, Y)$ is $true$. That is, for every $z \in D_i$, $c \sqsubseteq z \in S \cup S''$ and for every $\neg z \in D_i$, $c \sqsubseteq z' \in S \cup S''$. Let ρ be a rule in $T' \cup T''$ regarding D_i (w. l. o. g.) with the form:

$$\begin{array}{c}
y_{i_1} \sqcap \ldots \sqcap y_{i_p} \sqcap y'_{i_{p+1}} \sqcap \ldots \sqcap y'_{i_q} \\
\sqcap_{k=1,k\neq j}^{s} x_{i_k} \sqcap_{k=s+1}^{t} x'_{i_k} \sqsubseteq x'_{i_j}
\end{array}$$

Since $\rho \in T$, we have $T \cup S'' \cup S \models c \sqsubseteq x'_{i_j}$. On the other hand, $T \cup S'' \cup S \models c \sqsubseteq x_{i_j}$ holds too, because $x_{i_j} \in D_i$. Therefore $T \cup S'' \cup S$ is not consistent, hence $T \cup S'' \not\models S$.

"If": Assume $\Phi \in \mathrm{QBF}_{2,\exists}$ does not hold. Hence, for every truth assignment $\phi(X)$, there exists a truth assignment $\mu(Y)$, such that $E_{\phi,\mu}(X, Y)$ is $false$. That is, each $D_{i\phi,\mu}(X, Y)$ $(1 \leq i \leq l)$ is $false$. We prove that there does not exist a minmax solution which contains $c \sqsubseteq e$ (resp. does not contain $c \sqsubseteq e'$). Define the solution S as $S = \{c \sqsubseteq x_i : \phi(x_i) = true, 1 \leq i \leq n\} \cup \{c \sqsubseteq x'_i : \phi(x_i) = false, 1 \leq i \leq n\} \cup \{c \sqsubseteq e\}$. Clearly $T \cup S \models M$. Moreover, S is subset minimal. Next we show that there exists another subset minimal solution which is more informative than S. Define S' as $S' = \{c \sqsubseteq y_i : \mu(y_i) = true, 1 \leq i \leq m\} \cup \{c \sqsubseteq y'_i : \mu(y_i) = false, 1 \leq i \leq m\} \cup \{c \sqsubseteq e'\}$. First we show that $T \cup S \cup S'$ is consistent. From the construction of T, we notice that inconsistency can only occur if there is an $x_j \in X$ (resp. $x'_j \in X'$) such that $c \sqsubseteq x_j \in S$ (resp. $c \sqsubseteq x'_j \in S$), and $T \cup S \cup S' \models c \sqsubseteq x'_j$ (resp. $T \cup S \cup S' \models c \sqsubseteq x_j$) also holds.

Consider any subsumption $\rho = Q \sqsubseteq p$ in $T' \cup T''$. Assume ρ is regarding the disjunct D_i. If for every $z \in Q$, $(c \sqsubseteq z) \in S \cup S'$ holds, then except for one literal (we call it z_1), the truth assignments enable all other literals in D_i to be $true$. Since $D_{i\phi,\mu}(X, Y)$ is $false$, z_1 has to be $false$. If z_1 is a positive literal with the form of x, then x is assigned as $false$ in ϕ. Therefore $c \sqsubseteq x'$ is in S. From the construction of ρ we obtain that p is in fact x'. Thus $T \cup S \cup S' \models c \sqsubseteq x'$ holds, and $T \cup S \cup S'$ is consistent. Analogously, if z_1 is a negative literal with the form of $\neg x$, then x is assigned as $true$ in ϕ. Therefore $c \sqsubseteq x$ is in S. From the construction of ρ we obtain that p is in fact x. Thus $T \cup S \cup S' \models c \sqsubseteq x$ holds, and $T \cup S \cup S'$ is consistent.

Now that $T \cup S \cup S'$ is consistent, $T \cup S \cup S' \models S$ holds. Clearly $(S \cup S' \setminus c \sqsubseteq e)$ is a subset minimal solution. Moreover, it is straightforward to verify that $T \cup (S \cup S' \setminus c \sqsubseteq e) \models S$. This concludes the proof.

Complexity - $\mathcal{EL}$

In the following proofs we define the solution S_{or} as $S_{or} = \{P_i \sqsubseteq Q_i \mid \forall P_i, Q_i \in C : Or(P_i \sqsubseteq Q_i) = true\}$ with the intended meaning that S_{or} consists of all the subsumptions that are $true$ according to the domain expert.

General Case

Theorem 13 *To decide if $\mathcal{S}(T, C, Or, M) \neq \emptyset$ for a given GTAP (T, C, Or, M) is in P.*

Proof. To decide the existence problem, we need to test whether $T \cup S_{or} \models M$, and the entailment problem of $\mathcal{EL}$ is tractable (Baader, Brandt, and Lutz 2005). Note that $T \cup S_{or}$ is consistent, thus if $T \cup S_{or} \not\models M$, then there does not exist a solution.

Theorem 14 *To decide if a given ψ is relevant for a given GTAP (T, C, Or, M) is in P.*

Proof. We assume $Or(\psi)$ is $true$. Otherwise the the relevant problem returns $false$. The problem is equivalent to the existence problem. That is, if there exists a solution S, then $S \cup \psi$ is also a solution. If there does not exist a solution, then ψ is not relevant, since $\psi \in S_{or}$.

Theorem 15 *To decide if a given ψ is in all the solutions for a given GTAP (T, C, Or, M) is in P.*

Proof. Two entailment tests are called: (1) $T \cup S_{or} \models M$ and (2) $T \cup (S_{or} \setminus \psi) \not\models M$. If both (1) and (2) holds, then ψ is in every solution. Otherwise, either there does not exist a solution ((1) does not hold), or there is a solution that does not contain ψ ($T \cup (S_{or} \setminus \psi)$).

Subset Minimality

Theorem 16 *To decide if $\mathcal{S}_{min}(T, C, Or, M) \neq \emptyset$ for a given GTAP (T, C, Or, M) is in P.*

Proof. The problem is equivalent to the existence problem in general case. Detailed proof see Theorem 3.

Theorem 17 *To decide if a given ψ is min-relevant for a given GTAP (T, C, Or, M) is NP-complete.*

Proof. Hardness follows immediately due to the fact that the min-relevant problem for definite Horn theory problem is NP-complete (Friedrich, Gottlob, and Nejdl 1990), (Bienvenu 2008). For the upper bound, we can guess a solution S which contains ψ, and test whether $S \in \mathcal{S}_{min}(T, C, Or, M)$. Note that $S \in \mathcal{S}_{min}(T, C, Or, M)$ iff $T \cup S \models M$ and $\{\forall h \in S : T \cup (S \setminus h) \not\models M\}$ holds. Thus the problem is in NP.

Theorem 18 *To decide if a given ψ is in every minimal solution for a given GTAP (T, C, Or, M) is in P.*

Proof. The upper bound follows the proof in general case in Theorem 15. That is, two entailment tests are called: (1) $T \cup S_{or} \models M$ and (2) $T \cup (S_{or} \setminus \psi) \not\models M$. If both (1) and (2) holds, then ψ is in every solution, thus also in every solution of $\mathcal{S}_{min}(T, C, Or, M)$. Otherwise, $S = T \cup (S_{or} \setminus \psi)$ is a solution which does not contain ψ. Then there is a subset minimal solution $S' \subseteq S$. Obviously S' does not contain ψ as well.

Semantic Maximality For $\mathcal{EL}$ TBox, S_{or} if $T \cup S_{or} \models M$ is the most informative solution. Therefore all the decision problems are trivial.

Minmax

Theorem 19 *To decide if $\mathcal{S}^{max}_{\mathbf{min}}(T, C, Or, M) \neq \emptyset$ for a given GTAP (T, C, Or, M) is in P.*

Proof. Follows the counterpart in $\mathcal{EL}^{++}$, see Theorem 11.

Theorem 20 *To decide if a given ψ is minmax-relevant for a given GTAP (T, C, Or, M) is NP-complete.*

Proof. Hardness follows from the NP-complete complexity of the min-relevance problem. In the following we prove the upper bound. First a subset minimal solution S that contains ψ can be guessed and tested. Given a solution S, we define $closure(S) = \{x : T \cup S \models x\}$. Next we prove that S is minmax optimal iff $\{\forall h \in S : T \cup (S_{or} \setminus closure(S)) \cup (S \setminus h) \not\models h\}$. If: if $\{\forall h \in S : T \cup (S_{or} \setminus closure(S)) \cup (S \setminus h) \not\models h\}$ holds, then no element from S can be derived from outside the closure of S. Thus no more informative solution exists. Only if: assume $\{\exists h \in S : T \cup (S_{or} \setminus closure(S)) \cup (S \setminus h) \models h\}$ holds. Then $S' = (S_{or} \setminus closure(S)) \cup (S \setminus h)$ is a solution and $T \cup S' \models S$. We first reduce S' to S'' such that $T \cup S'' \models S$ holds and S'' is subset minimal. Next we show that S'' is more informative than S. Since S is subset minimal, $T \cup (S \setminus h) \not\models h$ holds. Then from S'' we know that there must be an $h' \in S''$ such that $h' \in (S_{or} \setminus closure(S))$. Then it follows that $T \cup S \not\models h'$.

Theorem 21 *To decide if a given ψ is in every minmax solution for a given GTAP (T, C, Or, M) is in P.*

Proof. The upper bound follows the proof in minimal case in Theorem 18.

Skyline

Theorem 22 *To decide if $\mathcal{S}^{max}_{min}(T, C, Or, M) \neq \emptyset$ for a given GTAP (T, C, Or, M) is in P.*

Proof. The problem is equivalent to the existence problem in general case, thus the upper bound follows immediately.

Theorem 23 *To decide if a given ψ is skyline-relevant for a given GTAP (T, C, Or, M) is NP-complete.*

Proof. The upper bound follows the NP-completeness of the skyline-relevant problem on $\mathcal{EL}^{++}$, see Theorem 8. To prove the hardness, we construct a reduction from the relevance problem of the subset minimality for $\mathcal{EL}$ as follows. Given a GTAP (T, C, Or, M) (denoted as P1) where T is a TBox in $\mathcal{EL}$, where $M = \{A \sqsubseteq B\}$. Note that this simplification does not affect the NP hardness of the problem. We construct another GTAP (T', C, Or, M) (denoted as P2), with $T' = T \cup \{P_i \sqsubseteq A, B \sqsubseteq Q_i | P_i \sqsubseteq Q_i \in S_{or}\}$. The intuition of P2 is that if there is a solution S such that $T \cup S \models M$, then $T \cup S \models S_{or}$ holds.

In the following we prove that a given ψ is subset minimal relevant to P1 if and only if ψ is skyline relevant to P2.

If: Assume ψ is skyline relevant to P2. There exists a solution S_2 containing ψ, such that there does not exist any solution $S'_2 \subset S_2$ and S'_2 is equally informative to S_2. Now we show that S_2 is also a subset minimal solution to P1. First we prove that $T \cup S_2 \models M$. Assume the opposite: $T \cup S_2 \not\models M$ holds, then it follows $T' \cup S_2 \not\models M$, because extending T with $\{P_i \sqsubseteq A, B \sqsubseteq Q_i$ does not result in the subsumption of $A \sqsubseteq B$. Assume S_2 is not subset minimal in P1. Then there is another solution $S_2'' \subset S_2$, such that $T \cup S_2'' \models M$. Then it follows $T' \cup S_2'' \models M$, thus S_2 and S_2'' are equally informative in P2, contradiction.

Only if: ψ is subset minimal relevant to P1. Then there exist a solution S_1 containing ψ and S_1 is a minimal solution. Since $T \subseteq T'$, S_1 is also a solution to P2. Since S_1 is minimal to P1, any subset of S_1 is not a solution to P1. With the same argument in the If direction, we can conclude that any subset of S_1 is not a solution either.

Theorem 24 *To decide if a given ψ is in every skyline solution for a given GTAP (T, C, Or, M) is in P.*

Proof. Follows Theorem 18.

Maxmin

Theorem 25 *To decide if $\mathcal{S}^{\mathbf{max}}_{min}(T, C, Or, M) \neq \emptyset$ for a given GTAP (T, C, Or, M) is in P.*

Proof. The problem is equivalent to the existence problem in general case, thus the upper bound follows immediately.

Theorem 26 *To decide if a given ψ is maxmin-relevant for a given GTAP (T, C, Or, M) is in P.*

Proof. Follows Theorem 23.

Theorem 27 *To decide if a given ψ is in every maxmin solution for a given GTAP (T, C, Or, M) is in P.*

Proof. Follows Theorem 18.

Algorithm - $\mathcal{EL}$

In this section we present an algorithm for repairing missing is-a structure (solving GTAP (T, C, Or, M)) in ontologies that are represented in $\mathcal{EL}$ and where the TBox is normalized as described in (Baader, Brandt, and Lutz 2005). A normalized TBox T contains only axioms of the forms $A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B$, $A \sqsubseteq \exists r.B$, and $\exists r.A \sqsubseteq B$, where A, A_1, …, A_n and B are atomic concepts and r is a role. Further, based on lessons learned in (Lambrix et al. 2013), we require that the missing is-a relations are validated before the repairing. We also note that $\mathcal{EL}$ TBoxes are always consistent. Thus $\forall m \in M : Or(m) = true$, and $T \cup M$ is consistent and therefore, M is a solution. The algorithm in Algorithm 1 computes a skyline optimal solution for a GTAP (T, C, Or, M). As M is a solution, the algorithm will always return a result. The result can be a subset minimal solution that is a subset of M or a solution that is more informative than M.

```
1  Procedure RepairSingleIsa begin
       Input: $E \sqsubseteq F$, T, Or, C
       Output: Solution for GTAP (T, C, Or, $\{E \sqsubseteq F\}$)
2      Sol := $\emptyset$;
3      Source := find superconcepts of E;
4      Target := find subconcepts of F;
5      foreach $A \in$ Source do
6          foreach $B \in$ Target do
7              if A and B are atomic concepts & $A \sqsubseteq B \in Or$ then
8                  if there exists $K \sqsubseteq L \in Sol$ such that $T \models A \sqsubseteq K$ and $T \models L \sqsubseteq B$ then
9                      do nothing;
10                 else
11                     remove every $K \sqsubseteq L \in$ Sol s.t. $T \models K \sqsubseteq A$ and $T \models B \sqsubseteq L$;
12                     Sol := Sol $\cup$ $\{A \sqsubseteq B\}$;
13             else if A is of the form $\exists r.N$ & B is of the form $\exists r.O$ then
14                 Sol := Sol $\cup$ RepairSingleIsa($N \sqsubseteq O$, T, Or, C);
15     return Sol;
16 Procedure RepairMultipleIsa begin
       Input: M, T, Or, C
       Output: Solution for GTAP (T, C, Or, M)
17     foreach $E_i \sqsubseteq F_i \in M$ do
18         SingleSol$_i$ := RepairSingleIsa($E_i \sqsubseteq F_i$, T, Or, C);
19     Solution := $\bigcup_i$SingleSol$_i$;
20     remove redundancy in Solution within same level of informativeness;
21     return Solution;
22 Procedure Repair begin
       Input: M, T, Or, C
       Output: Solution for GTAP (T, C, Or, M)
23     Missing := M;
24     Solution := RepairMultipleIsa(Missing, T, Or, C);
25     Final-Solution := Solution;
26     while Solution $\neq$ Missing do
27         Missing := Solution;
28         Solution := RepairMultipleIsa(Missing, T $\cup$ Missing, Or, C);
29         Final-Solution := Final-Solution $\cup$ Solution;
30         remove redundancy in Final-Solution within same level of informativeness;
31     return Final-Solution;
```

Algorithm 1: Solving GTAP.

The basic step in the algorithm (*RepairSingleIsa*) computes a solution for a GTAP with one missing is-a relation (i.e. GTAP $(T, C, Or, \{E \sqsubseteq F\})$ in the following way. First, superconcepts of E are collected in a *Source* set and subconcepts of F are collected in a *Target* set (lines 3 and 4). *Source* contains expressions of the forms A and $\exists r.A$ while *Target* contains expressions of the forms A, $A_1 \sqcap \ldots \sqcap A_n$ and $\exists r.A$ where A, A_1, …, A_n are atomic concepts and r is a role. Adding an is-a relation between an element in Source and an element in Target to the ontology would make $E \sqsubseteq F$ derivable (and thus this gives us logical solutions, but not necessarily solutions that are correct according to the domain). As we are interested in solutions containing is-a relations between atomic concepts, we check for every pair (A,B) $\in$ Source $\times$ Target whether A and B are atomic concepts and $Or(A \sqsubseteq B) = true$ (i.e. correct according to the domain). If so, then this is a possible solution for GTAP $(T, C, Or, \{E \sqsubseteq F\})$. However, to conform to subset minimality and semantic maximality, if the current solution already contains is-a relations that would lead to the entailment of $A \sqsubseteq B$ then we do not use $A \sqsubseteq B$ (8-9). Otherwise we use $A \sqsubseteq B$ and remove elements from the current solution that would be entailed if $A \sqsubseteq B$ is used (10-12). Further, in the case where A is of the form $\exists r.N$ and B is of the form $\exists r.O$, then making $N \sqsubseteq O$ derivable would also make $A \sqsubseteq B$ derivable (13-14). It is clear that for the result of *RepairSingleIsa*, i.e. Sol, the following holds: $T \cup Sol \models$

$E \sqsubseteq F$ and $\forall s \in Sol : Or(s) = true$. Together with the fact that $\mathcal{EL}$ TBoxes are consistent, this leads to the fact that Sol is a solution of GTAP $(T, C, Or, \{E \sqsubseteq F\})$.

In *RepairMultipleIsa* the algorithm collects for each missing is-a relation a solution from *RepairSingleIsa* and takes the union of these. Therefore, the following holds for Solution in line 19: $T \cup Solution \models M$ and $\forall s \in Solution : Or(s) = true$. Together with the fact that $\mathcal{EL}$ TBoxes are consistent, this leads to the fact that Solution is a solution of GTAP (T, C, Or, M). Further, in line 20, we remove redundancy while keeping the same level of informativeness, and thus obtain a skyline optimal solution. (In the case where there are several ways to remove redundancy, one is chosen, as the extended ontologies will be equivalent in the sense that they entail the same statements.)

In *Repair* we try to improve the result from *RepairMultipleIsa* by trying to find a skyline optimal solution on a higher level of informativeness. Given that any element in the solution of *RepairMultipleIsa* that is not in M can be considered as a new missing is-a relation (which was not detected earlier), we can try to find additional more informative ways of repairing by solving a new GTAP problem for these new missing is-a relations (and continue as long as new missing is-a relations are detected). As a (skyline optimal) solution for the new GTAP is also a (skyline optimal) solution of the original GTAP, the solution found in *Repair* is a skyline optimal solution for the original GTAP.

As an example run consider the GTAP in Figure 1. For a given ontology and set of missing is-a relations, the algorithm will first find solutions for repairing individual missing is-a relations using *RepairSingleIsA*. For the missing is-a relation Endocarditis $\sqsubseteq$ PathologicalPhenomenon the following is-a relations provide logical solutions for repairing the missing is-a relation: Endocarditis $\sqsubseteq$ PathologicalPhenomenon, Endocarditis $\sqsubseteq$ Fracture, Endocarditis $\sqsubseteq$ CardioVascularDisease, Carditis $\sqsubseteq$ PathologicalPhenomenon, Carditis $\sqsubseteq$ Fracture, Carditis $\sqsubseteq$ CardioVascularDisease as well as InflammationProcess $\sqsubseteq$ PathologicalProcess. As the first one is the missing is-a relation which was already validated, only the other six is-a relations are presented to the oracle for validation. Out of these six Endocarditis $\sqsubseteq$ Fracture and Carditis $\sqsubseteq$ Fracture are not correct according to the domain and are therefore not included in solutions. Further, relations Endocarditis $\sqsubseteq$ CardioVascularDisease, Endocarditis $\sqsubseteq$ PathologicalPhenomenon, Carditis $\sqsubseteq$ PathologicalPhenomenon are removed given it is possible to entail them from the ontology together with the remaining relations. Therefore, after validation, *RepairSingleIsA* returns $\{$InflammationProcess $\sqsubseteq$ PathologicalProcess, Carditis $\sqsubseteq$ CardioVascularDisease$\}$. The same process is repeated for the second missing is-a relation GranulomaProcess $\sqsubseteq$ NonNormalProcess. In this case the following is-a relations provide logical solutions for repairing the missing is-a relation: GranulomaProcess $\sqsubseteq$ NonNormalProcess and GranulomaProcess $\sqsubseteq$ PathologicalProcess. GranulomaProcess $\sqsubseteq$ NonNormalProcess is the missing is-a relation and was already validated as correct according to the domain. GranulomaProcess $\sqsubseteq$ PathologicalProcess is presented to the oracle and validated as correct according to the domain. As GranulomaProcess $\sqsubseteq$ NonNormalProcess can be entailed from the ontology together with GranulomaProcess $\sqsubseteq$ PathologicalProcess, *RepairSingleIsA* returns $\{$GranulomaProcess $\sqsubseteq$ PathologicalProcess$\}$. The solutions for the single is-a relations are then combined to form a solution for the set of missing is-a relations. In our case, there are no redundant relations and therefore *RepairMultipleIsA* returns $\{$InflammationProcess $\sqsubseteq$ PathologicalProcess, Carditis $\sqsubseteq$ CardioVascularDisease, GranulomaProcess $\sqsubseteq$ PathologicalProcess$\}$. We note that this is a skyline optimal solution. In *Repair* the system tries to improve the acquired solution. This time the oracle is presented with a total of 13 relations for validation out of which only one is validated to be correct, i.e. GranulomaProcess $\sqsubseteq$ InflammationProcess. This is added to the solution. Given this new is-a relation, GranulomaProcess $\sqsubseteq$ PathologicalProces is removed from the solution as it can now be entailed from the ontology and GranulomaProcess $\sqsubseteq$ InflammationProcess. The new solution is $\{$InflammationProcess $\sqsubseteq$ PathologicalProcess, Carditis $\sqsubseteq$ CardioVascularDisease, GranulomaProcess $\sqsubseteq$ InflammationProcess$\}$. This is again a skyline optimal solution and it is more informative than the previous solution. As new missing is-a relations were detected, the repairing is run for the third time. However, in this run the solution is not improved and thus the algorithm outputs the final result. We note that in this example we found a skyline optimal solution that is also semantically maximal. In general, however, it is not possible to know whether the solution is semantically maximal without checking every possible is-a relation between atomic concepts in the ontology.

Algorithm - $\mathcal{EL}^{++}$

In this section we present an algorithm for repairing missing is-a structure (solving GTAP (T, C, Or, M)) in ontologies that are represented in $\mathcal{EL}^{++}$ and where the TBox is normalized as described in (Baader, Brandt, and Lutz 2005).

A normalized TBox T contains only axioms of the forms $A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B$, $A \sqsubseteq \exists r.B$, and $\exists r.A \sqsubseteq B$, as well as role inclusions of the forms $r \sqsubseteq s$ and $r_1 \circ r_2 \sqsubseteq s$ where A, A_1, ..., A_n and B are atomic concepts and r, r_1 and r_2 are roles. As in the previous section, we require that the missing is-a relations are validated before the repairing i.e. $\forall m \in M : Or(m) = true$. We note that $\mathcal{EL}^{++}$ TBoxes can be inconsistent. Thus M is a solution iff $T \cup M$ is consistent. Therefore, we also require that $T \cup M$ is consistent. The algorithm in Algorithm 2 computes a skyline optimal solution for a GTAP (T, C, Or, M). As M is a solution, the algorithm will always return a result. The result can be a subset minimal solution that is a subset of M or a solution that is more informative than M.

The structure of the algorithm is similar to the structure of the algorithm in the previous section. There are two main differences. First, as $\mathcal{EL}^{++}$ Tboxes may be inconsistent, we need to check whether adding a possible solution to the Tbox will result in a consistent TBox. Secondly, given that $\mathcal{EL}^{++}$ allows role inclusion, the basic step in the algorithm (*RepairSingleIsa*) was modified to take this into account when searching for solutions which are found using axioms containing $\exists$ expressions. *RepairSingleIsa* computes a solution for a GTAP with one missing is-a relation (i.e. GTAP $(T, C, Or, \{E \sqsubseteq F\})$ in the following way. First, superconcepts of E are collected in a *Source* set and subconcepts of F are collected in a *Target* set. *Source* contains expressions of the forms A and $\exists r.A$ while *Target* contains expressions of the forms A, $A_1 \sqcap \ldots \sqcap A_n$ and $\exists r.A$ where A, $A_1, \ldots,$ A_n are atomic concepts and r is a role. As before, adding an is-a relation between an element in Source and an element in Target to the ontology would make $E \sqsubseteq F$ derivable. As we are interested in solutions containing is-a relations between atomic concepts, we check for every pair (A,B) $\in$ Source $\times$ Target whether A and B are atomic concepts and $Or(A \sqsubseteq B)$ = $true$. If so, then this is a possible solution. Further, if A is of the form $\exists r.N$ and B is of the form $\exists r.O$, then making $N \sqsubseteq O$ derivable would also make $A \sqsubseteq B$ derivable. In $\mathcal{EL}^{++}$ there are two more possibilities when A is of the form $\exists r.N$ and B is of the form $\exists s.O$. If T contains $r \sqsubseteq s$, then making $N \sqsubseteq O$ derivable would also make $A \sqsubseteq B$ derivable. Further, if T contains $r \circ r_1 \sqsubseteq s$ and $N \sqsubseteq \exists r_1.P$, then making $P \sqsubseteq O$ derivable would also make $A \sqsubseteq B$ derivable.

```
Procedure RepairSingleIsa begin
    Input: $E \sqsubseteq F$, T, Or, C
    Output: Solution for GTAP (T, C, Or, $\{E \sqsubseteq F\}$)
    Sol := $\emptyset$;
    Source := find superconcepts of E;
    Target := find subconcepts of F;
    foreach $A \in$ Source do
        foreach $B \in$ Target do
            if $T \cup$ Sol $\cup \{A \sqsubseteq B\}$ is consistent then
                if A and B are atomic concepts & $A \sqsubseteq B \in$ Or then
                    if there exists $K \sqsubseteq L \in$ Sol such that $T \models A \sqsubseteq K$
                    and $T \models L \sqsubseteq B$ then
                        do nothing;
                    else
                        remove every $K \sqsubseteq L \in$ Sol s.t. $T \models K \sqsubseteq A$
                        and $T \models B \sqsubseteq L$;
                        Sol := Sol $\cup \{A \sqsubseteq B\}$;
                else if A is of the form $\exists$r.N & B is of the form $\exists$s.O then
                    Extra_Sols := FindExistsSolutions(T, r, N, s, O);
                    foreach Rel $\in$ Extra_Sols do
                        Sol := Sol $\cup$ RepairSingleIsa(Rel, T, Or, C);
    return Sol;
Procedure RepairMultipleIsa begin
    Input: M, T, Or, C
    Output: Solution for GTAP (T, C, Or, M)
    foreach $E_i \sqsubseteq F_i \in M$ do
        SingleSol$_i$ := RepairSingleIsa(E$_i \sqsubseteq$ F$_i$, T, Or, C);
    Solution := $\bigcup_i$SingleSol$_i$;
    if $T \cup$ Solution is inconsistent then
        return M;
    remove redundancy in Solution within same level of informativeness;
    return Solution;
Procedure Repair begin
    Input: M, T, Or, C
    Output: Solution for GTAP (T, C, Or, M)
    Missing := M;
    Solution := RepairMultipleIsa(Missing, T, Or, C);
    Final-Solution := Solution;
    while Solution $\neq$ Missing do
        Missing := Solution;
        Solution := RepairMultipleIsa(Missing, T $\cup$ Missing, Or, C);
        Final-Solution := Final-Solution $\cup$ Solution;
        remove redundancy in Final-Solution within same level of
        informativeness;
    return Final-Solution;
Procedure FindExistsSolutions begin
    Input: T, r, N, s, O
    Output: Set of is-a relations
    CandidateSols := $\emptyset$;
    Compositions := find all role inclusions of form r $\sqsubseteq$ s or r $\circ$ r$_1$ $\sqsubseteq$ s in
    TBox T;
    foreach Comp $\in$ Compositions do
        if Comp is of form r $\sqsubseteq$ s then
            CandidateSols := CandidateSols $\cup$ {N $\sqsubseteq$ O};
        else
            Cs := { P | T $\models$ N $\sqsubseteq$ $\exists$r1.P };
            CandidateSols := CandidateSols $\cup$ {P $\sqsubseteq$ O | P $\in$ Cs};
    return CandidateSols;
```

Algorithm 2: Algorithm for solving GTAP in $\mathcal{EL}^{++}$.

As an example run for the algorithm in $\mathcal{EL}^{++}$ consider the GTAP in Figure 2. For a given ontology and set of missing is-a relations, the algorithm will first find solutions for repairing individual missing is-a relations using *RepairSingleIsA*. For the missing is-a relation Endocarditis $\sqsubseteq$ PathologicalPhenomenon the following is-a relations provide logical solutions for repairing the missing is-a relation: Endocarditis $\sqsubseteq$ PathologicalPhenomenon, Endocarditis $\sqsubseteq$ Fracture, Endocarditis $\sqsubseteq$ CardioVascularDisease, Carditis $\sqsubseteq$ PathologicalPhenomenon, Carditis $\sqsubseteq$ Fracture, Carditis $\sqsubseteq$ CardioVascularDisease as well as InflammationProcess $\sqsubseteq$ PathologicalProcess. As the first one is the missing is-a relation which was already validated, only the other six is-a relations are presented to the oracle for validation. Out of these six Endocarditis $\sqsubseteq$ Fracture and Carditis $\sqsubseteq$ Fracture are not correct according to the domain and are therefore not included in solutions. Further, relations Endocarditis $\sqsubseteq$ CardioVascularDisease, Endocarditis $\sqsubseteq$ PathologicalPhenomenon, Carditis $\sqsubseteq$ PathologicalPhenomenon are removed given it is possible to entail them from the ontology together with the remaining relations. Therefore, after validation, *RepairSingleIsA* returns $\{$InflammationProcess $\sqsubseteq$ PathologicalProcess, Carditis $\sqsubseteq$ CardioVascularDisease$\}$. The same process is repeated for the second missing is-a relation GranulomaProcess $\sqsubseteq$ NonNormalProcess. In this case the following is-a relations provide logical solutions for repairing the missing is-a relation: GranulomaProcess $\sqsubseteq$ NonNormalProcess and GranulomaProcess $\sqsubseteq$ PathologicalProcess. GranulomaProcess $\sqsubseteq$ NonNormalProcess is the missing is-a relation and was already validated as correct according to the domain. GranulomaProcess $\sqsubseteq$ PathologicalProcess is presented to the oracle and validated as correct according to the domain. As GranulomaProcess $\sqsubseteq$ NonNormalProcess can be entailed from the ontology together with GranulomaProcess $\sqsubseteq$ PathologicalProcess, *RepairSingleIsA* returns $\{$GranulomaProcess $\sqsubseteq$ PathologicalProcess$\}$. For the missing is-a relation Wound $\sqsubseteq$ PathologicalPhenomenon relations Wound $\sqsubseteq$ PathologicalPhenomenon, SoftTissueTraumaProcess $\sqsubseteq$ Pathological-

C = { GranulomaProcess, CardioVascularDisease, PathologicalPhenomenon, Fracture, Endocarditis, Carditis, InflammationProcess, PathologicalProcess, NonNormalProcess, Wound, BurningProcess, SoftTissueTraumaProcess, TraumaticProcess}

T = { CardioVascularDisease $\sqsubseteq$ PathologicalPhenomenon, Fracture $\sqsubseteq$ PathologicalPhenomenon, $\exists$isImmediateConsequence.PathologicalProcess $\sqsubseteq$ PathologicalPhenomenon, Endocarditis $\sqsubseteq$ Carditis, Endocarditis $\sqsubseteq$ $\exists$isImmediateConsequence.InflammationProcess, PathologicalProcess $\sqsubseteq$ NonNormalProcess, hasAssociatedProcess $\sqsubseteq$ isImmediateConsequence, Wound $\sqsubseteq$ $\exists$hasAssociatedProcess.SoftTissueTraumaProcess}

M = { Endocarditis $\sqsubseteq$ PathologicalPhenomenon, GranulomaProcess $\sqsubseteq$ NonNormalProcess, Wound $\sqsubseteq$ PathologicalPhenomenon, BurningProcess $\sqsubseteq$ SoftTissueTraumaProcess, BurningProcess $\sqsubseteq$ TraumaticProcess }

The following is-a relations are correct according to the domain, i.e Or returns $true$ for:
GranulomaProcess $\sqsubseteq$ InflammationProcess, GranulomaProcess $\sqsubseteq$ PathologicalProcess, GranulomaProcess $\sqsubseteq$ NonNormalProcess, CardioVascularDisease $\sqsubseteq$ PathologicalPhenomenon, Fracture $\sqsubseteq$ PathologicalPhenomenon, Endocarditis $\sqsubseteq$ PathologicalPhenomenon, Endocarditis $\sqsubseteq$ Carditis, Endocarditis $\sqsubseteq$ CardioVascularDisease,
Carditis $\sqsubseteq$ PathologicalPhenomenon, Carditis $\sqsubseteq$ CardioVascularDisease, InflammationProcess $\sqsubseteq$ PathologicalProcess, InflammationProcess $\sqsubseteq$ NonNormalProcess, PathologicalProcess $\sqsubseteq$ NonNormalProcess, Wound $\sqsubseteq$ PathologicalPhenomenon, TraumaticProcess $\sqsubseteq$ NonNormalProcess, TraumaticProcess $\sqsubseteq$ PathologicalProcess, SoftTissueTraumaProcess $\sqsubseteq$ TraumaticProcess, SoftTissueTraumaProcess $\sqsubseteq$ NonNormalProcess, SoftTissueTraumaProcess $\sqsubseteq$ PathologicalProcess, BurningProcess $\sqsubseteq$ NonNormalProcess, BurningProcess $\sqsubseteq$ PathologicalProcess, BurningProcess $\sqsubseteq$ SoftTissueTraumaProcess, BurningProcess $\sqsubseteq$ TraumaticProcess.

Let $\mathcal{P}$ = GTAP(T, C, Or, M).

Figure 2: Small $\mathcal{EL}^{++}$ example.

Process, Wound $\sqsubseteq$ Fracture, Wound $\sqsubseteq$ CardioVascularDisease provide logical solutions for repairing the missing is-a relation. Out of these, only Wound $\sqsubseteq$ PathologicalPhenomenon and SoftTissueTraumaProcess $\sqsubseteq$ PathologicalProcess are correct according to the oracle, and *RepairSingleIsA* therefore returns {Wound $\sqsubseteq$ PathologicalPhenomenon, SoftTissueTraumaProcess $\sqsubseteq$ PathologicalProcess}. For the remaining missing is-a relations BurningProcess $\sqsubseteq$ SoftTissueTraumaProcess and BurningProcess $\sqsubseteq$ TraumaticProcess the procedure *RepairSingleIsA* returns {BurningProcess $\sqsubseteq$ SoftTissueTraumaProcess} and {BurningProcess $\sqsubseteq$ TraumaticProcess} respectively. The solutions for the single is-a relations are then combined to form a solution for the set of missing is-a relations. In our case, Wound $\sqsubseteq$ PathologicalPhenomenon is redundant and therefore *RepairMultipleIsA* returns {InflammationProcess $\sqsubseteq$ PathologicalProcess, Carditis $\sqsubseteq$ CardioVascularDisease, GranulomaProcess $\sqsubseteq$ PathologicalProcess, BurningProcess $\sqsubseteq$ TraumaticProcess, BurningProcess $\sqsubseteq$ SoftTissueTraumaProcess, SoftTissueTraumaProcess $\sqsubseteq$ PathologicalProcess}. We note that this is a skyline optimal solution. In *Repair* the system tries to improve the acquired solution. This time the oracle is presented with a total of 25 relations for validation out of which only two are validated to be correct, i.e. GranulomaProcess $\sqsubseteq$ InflammationProcess and SoftTissueTraumaProcess $\sqsubseteq$ TraumaticProcess. These are added to the solution. Given these new is-a relations, GranulomaProcess $\sqsubseteq$ PathologicalProcess and BurningProcess $\sqsubseteq$ TraumaticProcess are removed from the solution as they are redundant. The new solution is {InflammationProcess $\sqsubseteq$ PathologicalProcess, Carditis $\sqsubseteq$ CardioVascularDisease, GranulomaProcess $\sqsubseteq$ InflammationProcess, SoftTissueTraumaProcess $\sqsubseteq$ TraumaticProcess, BurningProcess $\sqsubseteq$ SoftTissueTraumaProcess, SoftTissueTraumaProcess $\sqsubseteq$ PathologicalProcess}. This is again a skyline optimal solution and it is more informative than the previous solution.

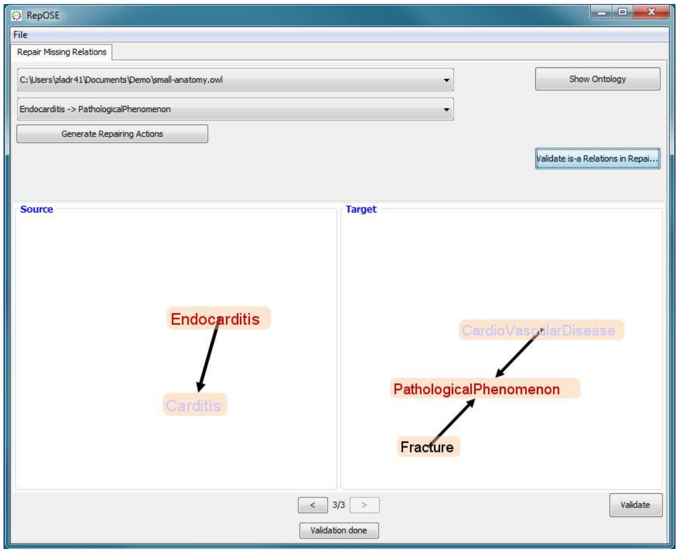
As new missing is-a relations were detected, the repairing is run for the third time. In this iteration 5 relations required validation and only relation TraumaticProcess $\sqsubseteq$ PathologicalProcess is validated as correct according to the domain. The new solution is {InflammationProcess $\sqsubseteq$ PathologicalProcess, Carditis $\sqsubseteq$ CardioVascularDisease, GranulomaProcess $\sqsubseteq$ InflammationProcess, SoftTissueTraumaProcess $\sqsubseteq$ TraumaticProcess, BurningProcess $\sqsubseteq$ SoftTissueTraumaProcess, TraumaticProcess $\sqsubseteq$ PathologicalProcess}. The relation SoftTissueTraumaProcess $\sqsubseteq$ PathologicalProcess was removed from the solution as it is redundant.

The algorithm is run again and in this iteration no new is-a relations were validated to be correct so the solution from the previous iteration is returned as the final solution.

System

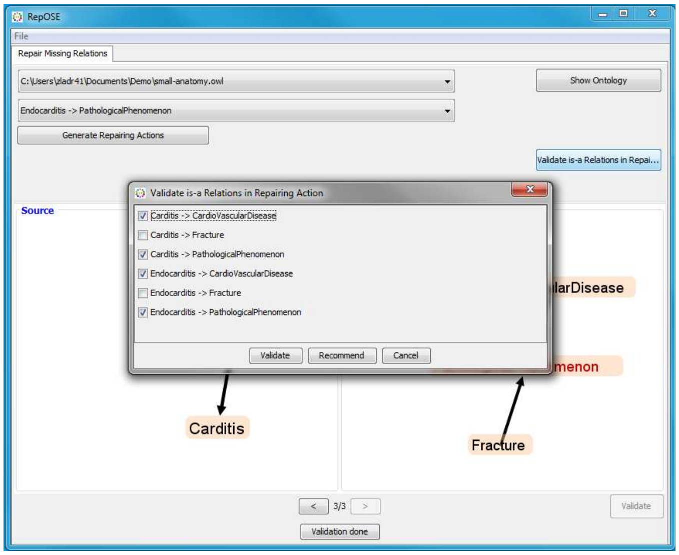
We have implemented a system for repairing missing is-a relations. The input to the system is a an ontology and a set of validated missing is-a relations. The output is a solution to GTAP (called a *repairing action*). The system was implemented in Java and uses the ELK reasoner (version 0.4.1) (Kazakov, Krötzsch, and Simančík 2011) to detect implicit entailments in the ontology. The system is semi-automatic and requires interaction with a user which is a domain expert serving as an oracle and who decides whether an is-a relation is correct according to the domain.

Once the ontology and the set of missing is-a relations are loaded, the user starts the debugging process by pressing the button `Generate Repairing Actions`. The system then removes redundant is-a relations and the non-redundant missing is-a relations are shown in a drop-down list allowing


C:\Users\zladr41\Documents\Demo\small-anatomy.owl

Endocarditis -> PathologicalPhenomenon

Figure 3: Screenshot - Repairing using Source and Target sets.


C:\Users\zladr41\Documents\Demo\small-anatomy.owl

Endocarditis -> PathologicalPhenomenon

Validate is-a Relations in Repairing Action

☑ Carditis -> CardioVascularDisease

□ Carditis -> Fracture

☑ Carditis -> PathologicalPhenomenon

☑ Endocarditis -> CardioVascularDisease

□ Endocarditis -> Fracture

☑ Endocarditis -> PathologicalPhenomenon

Figure 4: Screenshot - Validating is-a relations in a repairing action.

the user to switch between missing is-a relations. Additional relations acquired using $\exists$ expressions are also included in the drop-down list. It is also possible to scroll between relations using the arrow buttons in the bottom part of the screen.

After selecting an is-a relation from the list, the user is presented with the Source and the Target set for that is-a relation. The user then needs to choose relations which are correct according to the domain for that is-a relation. Missing is-a relations are automatically validated to be correct according to the domain while the relations that were acquired using $\exists$ expressions have to be explicitly validated by the user.

In Figure 3 the user is presented with the Source and the Target set for the missing is-a relation Endocarditis $\sqsubseteq$ PathologicalPhenomenon (concepts in the missing is-a relation are marked in red). In this case the user has selected $\{$Carditis $\sqsubseteq$ CardioVascularDisease$\}$ as a repairing action for the missing is-a relation (concepts marked in purple) and needs to confirm this by clicking the `Validate` button.

The user also has the option to check which relations have been validated so far and which relations can be validated, by clicking the `Validate Is-a Relations` button. In the pop-up window that appears the user can validate new relations, remove validations from already validated relations as well as ask for a recommendation by clicking the `Recommend` button (Figure 4). Recommendations are acquired by querying external sources (currently, WordNet, UMLS Methathesaurus and Uberon).

The validation phase is ended by clicking on the `Validation Done` button. The system then calculates the consequences of the chosen repairing actions and presents the user with a new set of is-a relations that need to be repaired. The validation phase and consequent computations represent one iteration of the Repair procedure in Algorithm 1. If the repairing did not change between two iterations the system outputs the repairing.

At any point the user can save validated relations from the ”File” menu which makes it possible to do debugging accross multiple sessions.

Experiments

We have run several debugging experiments on an Intel Core i7-2620M Processor at 3.07 GHz with 4 GB RAM under Windows 7 Professional and Java 1.7 compiler. In all experiments the validation phase took the most time while the computations between iterations took less than 10 seconds.

The results are summarized in Tables 3 - 7. The ’It’ columns represent the different iterations of Repair in Algoritm 1. The ’Missing’ rows give the number of missing is-a relations in each iteration. Such a missing is-a relation can be repaired by adding itself (’Repaired by itself’), by adding other is-a relations that were not derivable in the ontology and thus represent new knowledge added to the ontology (’Repaired using new knowledge’). The ’New relations’ row shows how many new is-a relations were added to the ontology. When such relations were found using $\exists$ (lines 13 and 14 in the algorithm), then the number of such relations is shown in parentheses. We note that for iteration $i + 1$ the number of missing is-a relations is the number of new relations from iteration i plus the number of missing is-a relations repaired by themselves from iteration i if there are no redundant relations. We also note that in the *last* iteration all missing is-a relations from that iteration are always repaired by themselves and these represent the final repairing action.

For the example in Figure 1 the system behaves as explained in the algorithm section and the results are summarized in Table 3. The results for the example in Figure 2 are given in Table 4. We also experimented with repairing an ontology for which we randomly removed is-a relations and

	It1	It2	It3
Missing	2	3	3
Repaired by itself	0	2	3
Repaired using new knowledge	2	1	0
New relations	3(1)	1	0

Table 3: Results for small ontology in Figure 1.

	It1	It2	It3
Missing	5	6	6
Repaired by itself	2	4	6
Repaired using new knowledge	3	2	0
New relations	4(2)	2	0

Table 4: Results for ontology in Figure 2.

then repaired the ontology. Further, we debugged the two ontologies from the Anatomy track at the 2013 Ontology Alignment Evaluation Initiative.

BioTop Experiment

In this experiment we used the Biotop ontology from the 2013 OWL Reasoner Evaluation Workshop dataset containing 280 concepts and 42 object properties. For the set of missing is-a relations we randomly selected 47 is-a relations. Then the ontology was modified by removing is-a relations which would make the selected is-a relations derivable. The unmodified ontology was used as domain knowledge in the experiment. The results for debugging Biotop ontology are presented in Table 5.

The debugging process took 4 iterations. In the first iteration 28 relations were repaired by adding new relations. In total 26 new relations were added in the first iteration out of which 3 are of the form $N \sqsubseteq O$ where for some missing is-a relation $A \sqsubseteq B$ the ontology contains axioms $A \sqsubseteq \exists r.N$ and $\exists r.O \sqsubseteq B$. For example, for missing is-a relation GreatApe $\sqsubseteq$ Primate we have a repairing action {FamilyHominidaeQuality $\sqsubseteq$ OrderPrimatesQuality} given that the ontology contains axioms GreatApe $\sqsubseteq$ $\exists$hasInherence.FamilyHominidaeQuality and $\exists$hasInherence.OrderPrimatesQuality $\sqsubseteq$ Primate.

The input to the second iteration contained 41 non-redundant is-a relations (4 redundant is-a relations were removed from the solution in iteration 1). In total 10 is-a relations were repaired by adding new is-a relations. Out of these 10 repaired is-a relations, 5 are relations from the initial set of missing is-a relations while the other 5 are relations which were added in the first iteration. For example, is-a relation Atom $\sqsubseteq$ Entity from the initial set of missing relations can be repaired with {Atom $\sqsubseteq$ MaterialEntity} given that MaterialEntity $\sqsubseteq$ Entity was added in the previous iteration.

In the third iteration, the input contained 42 is-a relations. In total 4 is-a relations (3 from the initial set of missing is-a relations and 1 from iteration 1) were repaired by adding 3 new relations. Out of the 3 new relations 1 is acquired using axioms containing $\exists$ expressions.

	It1	It2	It3	It4
Missing	47	41	42	41
Repaired by itself	19	31	38	41
Repaired using new knowledge	28	10	4	0
New relations	26(3)	11	3(1)	0

Table 5: Results for debugging the Biotop ontology.

Finally, in the fourth iteration no new relations were added and the system outputs the solution.

OAEI Anatomy Experiment

We debugged the two ontologies from the Anatomy track at the 2013 Ontology Alignment Evaluation Initiative, i.e. Mouse Anatomy ontology (AMA) containing 2744 concepts and a fragment of NCI human anatomy ontology (NCI-A) containing 3304 concepts. The input missing is-a relations for these two experiments were a set of 94 and 58 missing is-a relations, respectively, for AMA and NCI-A. These missing is-a relations were obtained by using a logic-based approach using an alignment between AMA and NCI-A (Lambrix and Liu 2013) to generate candidate missing is-a relations which were then validated by a domain expert to obtain actual missing is-a relations.

Mouse Anatomy The results for debugging AMA are given in Table 6. Three iterations were required to reach the final solution. Out of 94 initial missing is-a relations 37 were repaired by repairing actions which add new knowledge to the ontology while 57 were repaired using only the missing is-a relation itself. In total 44 new and non-redundant relations were added to the ontology in the first iteration. Out of 37 relations which were repaired by adding new relations, 22 had more than 1 non-redundant relation in the repairing action. For example, the missing is-a relation wrist joint $\sqsubseteq$ joint is repaired by a repairing action {limb joint $\sqsubseteq$ joint, wrist joint $\sqsubseteq$ synovial joint}.

The set of missing is-a relations in the second iteration contains 101 relations, i.e. 57 relations which were repaired by adding the missing is-a relation itself and 44 newly added relations. In this iteration, 3 is-a relations were repaired by adding new knowledge to the ontology. All 3 of these is-a relations are is-a relations which were added in the previous iteration. For example, is-a relation wrist joint $\sqsubseteq$ synovial joint is repaired by a repairing action {wrist joint $\sqsubseteq$ hand joint} which is possible given that the is-a relation metacarpo-phalangeal joint $\sqsubseteq$ joint from the initial set of missing is-a relations was repaired by a repairing action {hand joint $\sqsubseteq$ synovial joint, limb joint $\sqsubseteq$ joint} in the first iteration.

Finally, the set of missing is-a relations containing 101 is-a relations in the third iteration is also the solution for the initial set of missing is-a relations given that no new relations were added in the third iteration.

NCI - Human Anatomy The initial set of missing is-a relations contained 58 relations for the NCI-A ontology. Out of these 58 relations in the first iteration 9 were repaired by

	It1	It2	It3
Missing	94	101	101
Repaired by itself	57	98	101
Repaired using new knowledge	37	3	0
New relations	44	3	0

Table 6: Results for debugging AMA - Mouse Anatomy ontology.

	It1	It2	It3
Missing	58	55	54
Repaired by itself	49	50	54
Repaired using new knowledge	9	5	0
New relations	6	4	0

Table 7: Results for debugging NCI-A - Human Anatomy ontology.

adding relations which introduce new knowledge to the ontology. In total 6 new is-a relations were added.

In the second iteration, 5 out of 55 is-a relations were repaired by adding new relations while repairing actions for the 50 other is-a relations were unchanged. All 5 is-a relations which were repaired by adding new relations to the ontology are is-a relations which were repaired by repairing actions containing only the missing is-a relation from the first iteration. This exemplifies why it is beneficial to consider already repaired is-a relations in subsequent iterations as Source and Target sets for some missing is-a relations can change and more informative solutions might be identified.

The input to the third iteration is a set of 54 is-a relations and given that no changes were made, these relations are the final solution.

Lessons Learned

The experiments have shown that the iterative approach to repairing missing is-a relations is beneficial as in all our experiments additional relations were added to the ontology in subsequent iterations. Running the system on already repaired is-a relations gives the opportunity to identify new repairing actions which introduce new knowledge to the ontology. An example of this is found in the BioTop experiment where is-a relations from the initial set of missing is-a relations were repaired by more informative solutions in the third iteration.

Currently, the system removes redundant is-a relations from a solution after every iteration. This step is crucial for producing skyline optimal solutions. However, in situations where an is-a relation is repaired by a relation acquired from the axioms containing $\exists$ expressions it might be advantageous to keep also the missing is-a relation in subsequent iterations even though it is redundant. The reason for this is that the Source set and the Target set for the missing is-a relation might get updated in later iterations and therefore new repairing actions might be identified. One way to solve this is to make it possible in the system to show these missing is-a relations with their Source and Target sets but not to include them in the solution unless they are repaired using new knowledge. For example, let us assume that the missing is-a relation Human $\sqsubseteq$ Primate was repaired in one iteration by a repairing action $\{$Human $\sqsubseteq$ Primate, SpeciesHomoSapiensQuality $\sqsubseteq$ OrderPrimatesQuality$\}$ in which case the second relation was found using $\exists$. In the next iteration the relation GreatApe $\sqsubseteq$ Primate was added to the ontology. If the system removed redundant relation Human $\sqsubseteq$ Primate then relation Human $\sqsubseteq$ GreatApe would not be detected as a possible repairing action for Human $\sqsubseteq$ Primate.

In cases where missing is-a relations are repaired using new knowledge, new is-a relations are added to the ontology which were not derivable before. These new is-a relations can be considered as missing is-a relations as they were not detected by the detection algorithm. Given this, the system can also be used for completing the is-a structure of ontologies, even when no missing is-a relations are available. This can be achieved by using a set of is-a relations which are already derivable from the ontology as input. As in the BioTop experiment, by doing this, the system may identify additional is-a relations which represent new knowledge which can be added to the ontology. This methodology also allows a domain expert to deal with existing is-a relations which the domain expert has identified as relations which need to be revised or investigated further.

Related Work

The abduction framework has been applied to the database and knowledge representation problems. In the early years it was used in database update problems (Kakas and Mancarella 1990). Database provenance (Cheney, Chiticariu, and Tan 2009) is a variant of an abduction process. (Eiter and Gottlob 1995) is the most related article regarding the proof techniques in the current paper. Calvanese et al. (Calvanese et al. 2011) presented the complexity results on ABox abduction regarding conjunctive query answering over DL-Lite ontologies, that is, to explain why a given tuple is missing in the answer set.

There is not much work on the *repairing of missing is-a structure*. In (Lambrix, Liu, and Tan 2009; Lambrix and Liu 2013) this was addressed in the setting of taxonomies where the problem as well as some preference criteria were defined. Further, an algorithm was given and an implemented system was proposed. A later version of that system (Lambrix and Ivanova 2013), also dealing with semantic defects, was then used for debugging ontologies related to a project for the Swedish National Food Agency (Ivanova et al. 2012). An extension dealing with both ontology debugging and ontology alignment is described in (Ivanova and Lambrix 2013). In (Lambrix, Dragisic, and Ivanova 2012) the problem was formalized as an abduction problem and an algorithm was given for finding solutions for $\mathcal{ALC}$ acyclic terminologies. In this paper we extend the previous formalization by formalizing the role of the domain expert as well as by introducing preference criteria for the solutions to the problem. Further, we present complexity results for different decision problems and provide an algorithm for $\mathcal{EL}$ and $\mathcal{EL}^{++}$ ontologies. Also, the algorithms in this paper can be restricted to taxonomies and in that case finds more informative solutions than (Lambrix, Liu, and Tan 2009). Except for (Lambrix, Dragisic, and Ivanova 2012) in which GTAP without abduciles was defined, there is no other work yet on *GTAP*. There is some work on TBox abduction. (Hubauer, Lamparter, and Pirker 2010) proposes an automata-based approach to TBox abduction in $\mathcal{EL}$. It is based on a reduction to the axiom pinpointing problem which is then solved with automata-based methods.

Further, there is work that addresses *related topics* but not directly the problem that is addressed in this paper. There is much work on the *detection of missing (is-a) relations* in e.g. ontology learning (Cimiano, Buitelaar, and Magnini 2005), using linguistic (Hearst 1992) and logical (Corcho et al. 2009) patterns, or by using knowledge inherent in an ontology network (Lambrix, Liu, and Tan 2009; Ivanova et al. 2012). As mentioned before, these approaches, in general, do not detect all missing is-a relations. There is also much work on a dual problem to the one addressed in this paper, i.e. the *debugging of semantic defects.* Most of the work on debugging semantic defects aims at identifying and removing logical contradictions from an ontology (Haase and Stojanovic 2005; Schlobach 2005; Kalyanpur et al. 2006b; 2006a; Flouris et al. 2008), from mappings between ontologies (Meilicke, Stuckenschmidt, and Tamilin 2007; Wang and Xu 2008; Ji et al. 2009; Qi, Ji, and Haase 2009) or ontologies in a network (Jimenez-Ruiz et al. 2009; Ivanova et al. 2012).

Finally, there is also work on other *abductive reasoning problems in (simple) description logics* including concept abduction (Colucci et al. 2004; Bienvenu 2008; Donini et al. 2009) and ABox abduction (Du et al. 2011; Klarman, Endriss, and Schlobach 2011; Calvanese et al. 2012) as defined in (Elsenbroich, Kutz, and Sattler 2006).

Conclusions and Future Work

We have studied the GTAP in the context of ontology repairing. We first defined a model of GTAP and extended it with various preferences. Then we presented complexity results on the existence, relevance and necessity decision problems for ontologies that can be represented as TBoxes using a member of the $\mathcal{EL}$ family. Unless the polynomial hierarchy collapses, GTAP is much harder than the classical deduction problem, which is tractable for $\mathcal{EL}^{++}$. Further, we provided algorithms and a system for finding skyline optimal solutions to the GTAP and showed its usefulness through experiments.

In the future, we are interested in studying the GTAP for other knowledge representation languages. Further, we will investigate variants of the GTAP with different preference relations and restrictions of the signature. Another interesting topic is to study the GTAP in the context of modular ontologies where it may not be possible to introduce changes in the imported ontologies. Further, we will look into the integration of different abduction frameworks to deal with both modeling and semantic defects.

Acknowledgements

We thank the Swedish Research Council (Vetenskapsrådet), the Swedish e-Science Research Centre (SeRC) and the Swedish National Graduate School in Computer Science (CUGS) for financial support.

References

Ashburner, M.; Ball, C.; Blake, J.; Botstein, D.; Butler, H.; Cherry, J.; Davis, A.; Dolinski, K.; Dwight, S.; Eppig, J.; Harris, M.; Hill, D.; Issel-Tarver, L.; Kasarskis, A.; Lewis, S.; Matese, J.; Richardson, J.; Ringwald, M.; Rubin, G.; and Sherlock, G. 2000. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics* 25(1):25–29.

Baader, F.; Brandt, S.; and Lutz, C. 2005. Pushing the $\mathcal{EL}$ envelope. In *19th International Joint Conference on Artificial Intelligence*, 364–369.

Bienvenu, M. 2008. Complexity of abduction in the $\mathcal{EL}$ family of lightweight description logics. In *11th International Conference on Principles of Knowledge Representation and Reasoning*, 220–230.

Calvanese, D.; Ortiz, M.; Simkus, M.; and Stefanoni, G. 2011. The complexity of conjunctive query abduction in DL-lite. In *International Workshop on Description Logics*, 81–91.

Calvanese, D.; Ortiz, M.; Simkus, M.; and Stefanoni, G. 2012. The complexity of explaining negative query answers in DL-Lite. In *13th International Conference on Principles of Knowledge Representation and Reasoning*, 583–587.

Cheney, J.; Chiticariu, L.; and Tan, W.-C. 2009. Provenance in databases: Why, how, and where. *Foundations and Trends in Databases* 1(4):379–474.

Cimiano, P.; Buitelaar, P.; and Magnini, B. 2005. *Ontology Learning from Text: Methods, Evaluation and Applications*. IOS Press.

Colucci, S.; Di Noia, T.; Di Sciascio, E.; Donini, F.; and Mongiello, M. 2004. A uniform tableaux-based approach to concept abduction and contraction in $\mathcal{ALN}$. In *International Workshop on Description Logics*, 158–167.

Corcho, O.; Roussey, C.; Vilches, L. M.; and Pérez, I. 2009. Pattern-based OWL ontology debugging guidelines. In *Workshop on Ontology Patterns*, 68–82.

Donini, F.; Colucci, S.; Di Noia, T.; and Di Sciasco, E. 2009. A tableaux-based method for computing least common subsumers for expressive description logics. In *21st International Joint Conference on Artificial Intelligence*, 739–745.

Du, J.; Qi, G.; Shen, Y.-D.; and Pan, J. 2011. Towards practical Abox abduction in large OWL DL ontologies. In *25th AAAI Conference on Artificial Intelligence*, 1160–1165.

Eiter, T., and Gottlob, G. 1995. The complexity of logic-based abduction. *Journal of the ACM* 42(1):3–42.

Elsenbroich, C.; Kutz, O.; and Sattler, U. 2006. A case for abductive reasoning over ontologies. In *OWL: Experiences and Directions*.

Flouris, G.; Manakanatas, D.; Kondylakis, H.; Plexousakis, D.; and Antoniou, G. 2008. Ontology Change: Classification and Survey. *Knowledge Engineering Review* 23(2):117–152.

Friedrich, G.; Gottlob, G.; and Nejdl, W. 1990. Hypothesis classification, abductive diagnosis and therapy. In *Proceedings of the International Workshop on Expert Systems in Engineering : Principles and Applications: Principles and Applications*, 69–78. Springer-Verlag New York, Inc.

Garey, M., and Johnson, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co.

Haase, P., and Stojanovic, L. 2005. Consistent Evolution of OWL Ontologies. In *2nd European Semantic Web Conference*, 182–197.

Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. In *14th International Conference on Computational Linguistics*, 539–545.

Hubauer, T.; Lamparter, S.; and Pirker, M. 2010. Automata-based abduction for tractable diagnosis. In *International Workshop on Description Logics*, 360–371.

Ivanova, V., and Lambrix, P. 2013. A unified approach for aligning taxonomies and debugging taxonomies and their alignments. In *10th Extended Semantic Web Conference*, 1–15.

Ivanova, V.; Laurila Bergman, J.; Hammerling, U.; and Lambrix, P. 2012. Debugging taxonomies and their alignments: the ToxOntology - MeSH use case. In *1st International Workshop on Debugging Ontologies and Ontology Mappings*, 25–36.

Ji, Q.; Haase, P.; Qi, G.; Hitzler, P.; and Stadtmuller, S. 2009. RaDON - repair and diagnosis in ontology networks. In *6th European Semantic Web Conference*, 863–867.

Jimenez-Ruiz, E.; Grau, B. C.; Horrocks, I.; and Berlanga, R. 2009. Ontology Integration Using Mappings: Towards Getting the Right Logical Consequences. In *6th European Semantic Web Conference*, 173–187.

Kakas, A. C., and Mancarella, P. 1990. Database updates through abduction. In *16th International Conference on Very Large Data Bases*, 650–661.

Kalyanpur, A.; Parsia, B.; Sirin, E.; and Cuenca-Grau, B. 2006a. Repairing Unsatisfiable Concepts in OWL Ontologies. In *3rd European Semantic Web Conference*, 170–184.

Kalyanpur, A.; Parsia, B.; Sirin, E.; and Hendler, J. 2006b. Debugging Unsatisfiable Classes in OWL Ontologies. *Journal of Web Semantics* 3(4):268–293.

Kazakov, Y.; Krötzsch, M.; and Simančík, F. 2011. Concurrent classification of $\mathcal{EL}$ ontologies. In *10th International Semantic Web Conference*, 305–320.

Klarman, S.; Endriss, U.; and Schlobach, S. 2011. Abox abduction in the description logic $\mathcal{ALC}$. *Journal of Automated Reasoning* 46:43–80.

Lambrix, P., and Ivanova, V. 2013. A unified approach for debugging is-a structure and mappings in networked taxonomies. *Journal of Biomedical Semantics* 4:10.

Lambrix, P., and Liu, Q. 2013. Debugging the missing is-a structure within taxonomies networked by partial reference alignments. *Data & Knowledge Engineering* 86:179–205.

Lambrix, P.; Wei-Kleiner, F.; Dragisic, Z.; and Ivanova, V. 2013. Repairing missing is-a structure in ontologies is an abductive reasoning problem. In *2nd International Workshop on Debugging Ontologies and Ontology Mappings*, 33–44.

Lambrix, P.; Dragisic, Z.; and Ivanova, V. 2012. Get my pizza right: Repairing missing is-a relations in $\mathcal{ALC}$ ontologies. In *2nd Joint International Semantic Technology Conference*, 17–32.

Lambrix, P.; Liu, Q.; and Tan, H. 2009. Repairing the Missing is-a Structure of Ontologies. In *4th Asian Semantic Web Conference*, 76–90.

Meilicke, C.; Stuckenschmidt, H.; and Tamilin, A. 2007. Repairing Ontology Mappings. In *22th National Conference on Artificial Intelligence*, 1408–1413.

Papadimitriou, C. M. 1994. *Computational complexity*. Reading, Massachusetts: Addison-Wesley.

Qi, G.; Ji, Q.; and Haase, P. 2009. A Conflict-Based Operator for Mapping Revision. In *8th International Semantic Web Conference*, 521–536.

Schlobach, S. 2005. Debugging and Semantic Clarification by Pinpointing. In *2nd European Semantic Web Conference*, 226–240.

Wang, P., and Xu, B. 2008. Debugging ontology mappings: a static approach. *Computing and Informatics* 27:21–36.

Wei-Kleiner, F.; Dragisic, Z.; and Lambrix, P. 2014. Abduction framework for repairing incomplete $\mathcal{EL}$ ontologies: Complexity results and algorithms. In *28th AAAI Conference on Artificial Intelligence*.