Proc. 23rd Int. Workshop on Description Logics (DL2010), CEUR-WS 573, Waterloo, Canada, 2010.

Structure Preserving TBox Repair Using Defaults

Thomas Scharrenbach[1], Rolf Grütter[1], Bettina Waldvogel[1], and Abraham Bernstein[2]

[1] Swiss Federal Institute for Forest, Snow and Landscape Research WSL
Zürcherstrasse 111, 8903 Birmensdorf, Switzerland
{Thomas.Scharrenbach, Rolf.Gruetter, Bettina.Waldvogel}@wsl.ch
[2] University of Zurich, Department of Informatics Zurich, Switzerland
{bernstein}@ifi.uzh.ch

Abstract. Unsatisfiable concepts are a major cause for inconsistencies in Description Logics knowledge bases. Popular methods for repairing such concepts aim to remove or rewrite axioms to resolve the conflict by the original logics used. Under certain conditions, however, the structure and intention of the original axioms must be preserved in the knowledge base. This, in turn, requires changing the underlying logics for repair. In this paper, we show how Probabilistic Description Logics, a variant of Reiter's default logics with Lehmann's Lexicographical Entailment, can be used to resolve conflicts fully-automatically and receive a consistent knowledge base from which inferences can be drawn again.

Key words: default logics, unsatisfiability, justifications, TBox repair

1 Introduction

Ontologies have become standard for knowledge representation in the Semantic Web. While ontologies are usually expressed in Web Ontology Language (OWL) recommended by the W3C [1], one of the underlying formalisms for reasoning about data in the ontology is the Description Logic (DL) $\mathcal{SHOIN}(D)$, being a decidable subset of first-order logic [2].

Knowledge may evolve over time and might lead to contradictions in the knowledge base. Contradictions may as well occur when mapping two ontolgies on each other. In the case of terminological knowledge, this causes concepts to be inferred unsatisfiable. For example, in Figure 1, the concepts C, D and E are inferred unsatisfiable. Unsatisfiable concepts, in turn, cause the whole knowledge base to be inconsistent, if there exist assertions instantiating them.

Traditional approaches make the TBox satisfiable again by removing trouble-causing axioms and (possibly) adding new axioms modelling the unsatisfiability [3]. This will, anyway, lead to a loss of the information originally specified in the ontology. However, under certain conditions, all axiomatic information

[3] The second case can be seen as axiom rewriting.

should be preserved as much as possible in its original form as well as intuition. We propose to use *default logics* for relaxing the axioms that cause the incoherency.

Defaults, introduced by Reiter [3] and re-interpreted by Lehmann [4], facilitate the co-existence of default rules for typical cases together with exceptions from these rules. When querying the knowledge base, more specific knowledge, i.e. the exceptions, is preferred to more general knowledge, i.e. the defaults.

Transforming subclass inclusion axioms into defaults requires an extension of traditional DL reasoning that copes with the properties that come along with defaults. Probabilistic Description Logics (PDL) [5] is currently the only approach that is able to provide $\mathcal{SHOIN}(D)$ default reasoning, yet as a special case.

We introduce the Δ-transformation for transforming DL axioms and sets of these into defaults. We can show that, under certain conditions, transforming the axioms justifying the unsatisfiability of concepts [4] in the TBox results in a consistent P-$\mathcal{SHOIN}(D)$ knowledge base which re-enables us to draw conclusions.

This work is structured as follows: After introducing preliminaries and custom notions for methods used in Section 2, we present the proposed transformation scheme in Section 3. The actual approach along with supporting examples and formal framework is given in Sections 4 and 5. In Section 6 we give an overview about related work. We conclude this paper and give an outlook to future work in Section 7.

2 Unsatisfiable Concepts and Justifications

While we introduce the unsatisfiability of concept descriptions in Description Logics (DL) and how to justify these, we will not give an introduction to Description Logics in this paper. The interested reader os referred to [2]. For the rest of this paper, we will restrict ourselves to the DL $\mathcal{SHOIN}(D)$, because the methods presented in this paper build on Probabilistic Description Logics which is currently defined for $\mathcal{SHOIN}(D)$. An extension to the current W3C recommendation $\mathcal{SROIQ}(D)$ will remain for future work.

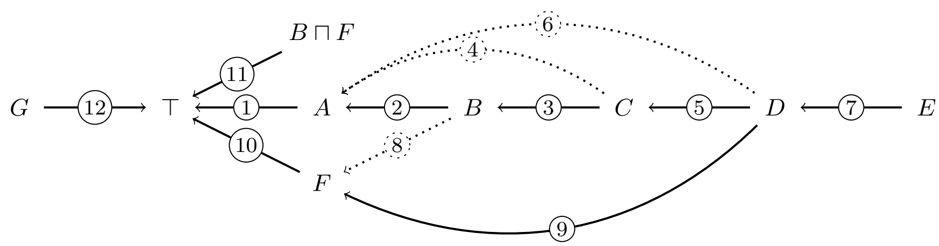
A concept description U is called unsatisfiable w.r.t. a TBox $\mathcal{T}$, iff $\mathcal{T} \models U \sqsubseteq \bot$. A justification for an entailment $\mathcal{T} \models \eta$ is the minimal set of axioms from $\mathcal{T}$ such that the entailment still holds. It is possible to compute the set of all justifications for an entailment [6] using an adapted version of Reiter's Minimum Hitting Set Tree (HST) [7] that originates from the area of Model Based Diagnosis.

Definition 1 (Justifications). *Let $\mathcal{T}$ be an TBox. $J_\eta \subseteq \mathcal{T}$ is a justification for $\mathcal{T} \models \eta$, iff $J_\eta \models \eta$ and $J'_\eta \not\models \eta$ for any $J'_\eta \subset J_\eta$.*

It turns out that the unsatisfiability of a concept description U_1 may depend on the unsatisfiability of another concept description U_2, i.e. $J_{U_1 \sqsubseteq \bot} \supseteq J_{U_2 \sqsubseteq \bot}$. In

[4] By concepts we consider atomic as well as concept descriptions, if not stated otherwise

Fig. 1. Example TBox. Nodes correspond to concepts and arcs correspond to subclass inclusions. Dotted arcs represent disjoints. The axioms are numbered for referring to them in the text. Below, the root justifications for the non-purely derived unsatisfiable concepts are shown.



$J^{root}_{1C\sqsubseteq\bot} = \{2,3,4\}$, $J^{root}_{2D\sqsubseteq\bot} = \{2,3,5,6\}$, $J^{root}_{3D\sqsubseteq\bot} = \{3,5,8,9\}$, $J^{root}_{4B\sqcap F\sqsubseteq\bot} = \{8,11\}$

this case we say that J_{U_2} is more general. The most general justifications for the unsatisfiable concepts of an ontology are called *root unsatisfiable.* [5]
It should be noted that root justifications are sets of axioms and should not be mixed up with the notion of *root unsatisfiable*, *partially derived* and *purely derived* concepts as denoted in [6]. However, there is a correspondence, i.e. every root unsatisfiable concept description has only root justifications, every partially dervied unsatisfiable concept description has at least one root justification and every purely derived unsatisfiable concept description has no root justification.

Definition 2 (Root Justifications). *Let $\mathcal{J}$ be the set of all justifications for all unsatisfiable concept description of a TBox $\mathcal{T}$. Then $J^{root}_{U\sqsubseteq\bot} \in \mathcal{J}$ is a* root justification *for some unsatisfiable concept description U, iff for any concept description U' there is no $J_{U'\sqsubseteq\bot} \in \mathcal{J}$ such that $J_{U'\sqsubseteq\bot} \subset J^{root}_{U\sqsubseteq\bot}$.*

Root justifications allow us to resolve only the most general causes for unsatisfiablility in a Tbox, which in turn result in the satisfiability of all unsatisfiable concepts. For example in Figure 1, the partially derived unsatisfiable concept D will be inferred unsatisfiable for the same reason as is the root unsatisfiable concept C. The unsatisfiability of concept E is purely derived, since it depends on the unsatisfiability of D. We do not have to distinguish between the unsatisfiability of these concepts as long as we remove the most general causes. Please also note, that the (root) justification for the concept description $B \sqcap F$ contains its declaration. If it did not, then $J^{root}_{3D\sqsubseteq\bot} \supset J^{root}_{4B\sqcap F\sqsubseteq\bot}$ and hence the unsatisfiability of the atomic concept D would depend on the unsatisfiability of the concept description $B \sqcap F$. This is indeed not the case, and hence this dependency has to

[5] Please note that axioms of the form $A \sqsubseteq \top$ are only included in a justification, if A is a complex concept description, but not, if A is an atomic concept.

be seen as an artifact. Therefore the declaration $B \sqcap F$ is included in the (root) justification.

3 Resolving Unsatisfiable Concepts

$\mathcal{SHOIN}(D)$ fulfils the monotonicity assumption, i.e. adding new axioms does not invalidate existing entailments or introduce unsatisfiability. Hence, unsatisfiability cannot be resolved by just adding new axioms. Repair has to involve the removal of axioms and is therefore always a non-monotone operation.

The currently most convenient way of resolving unsatisfiability in a TBox is to remove axioms that are responsible for it. This task is often referred to as OWL-Debugging. The interested reader may find a more detailed survey of approaches to OWL-Debugging in [6].

In addition to that, attempts for semi-automatic axiom rewriting have been made [6], referred to as repair plans. Common modelling errors have been identified empirically, and according to the kind of axioms that caused unsatisfiability, repair plans are generated and proposed to an end-user that decides how to repair the unsatisfiability.

Instead of doing the repair in $\mathcal{SHOIN}(D)$, it is also possible to change the formalism for knowledge representation and/or inference. We propose that it is desirable to keep as much of the original intention as well as structure of the stated axioms as possible. Keeping the axioms' structure requires some mechanism of how to prefer some of the contradicting axioms over the others to keep possible models of the ontology consistent.

Default logic is a way of generating a model of preferece for axioms of a first-order logic knowledge base that soley relies upon the structure of the knowledge base.

3.1 Probabilistic Description Logics

Recently, a method called *probabilistic description logics* (PDL) has been proposed [5] that extends a $\mathcal{SHOIN}(D)$ knowledge base with probabilistic constraints. Such a constraint $(A|B)[l,u]$ can be viewed as assigning the $\mathcal{SHOIN}(D)$ TBox axiom $B \sqsubseteq A$ the belief interval $[l,u]$ with $0 \leq l \leq u \leq 1$. The special case $l = u = 1$, however, corresponds to to Reiter's normal defaults [3]. For sake of readability, we omit the interval and write defaults as $(A|B)$.

As a consequence, PDL can be used as a way of modelling OWL-axioms $B \sqsubseteq A$ as a set of defaults $(A|B)$. The resulting logic is called P-$\mathcal{SHOIN}(D)$. PDL extends a classical $\mathcal{SHOIN}(D)$ TBox by a set of constraints called PBox $\mathcal{P}$. Together, both of these form a so-called PTBox PT $= (\mathcal{T}, \mathcal{P})$.

In case we restrict these constraints to defaults like above, inferences can be drawn according to Lehmann's lexicographical entailment [4]. The PTBox is partitioned into sets of defaults $P_0, \ldots, P_N$ where P_0 contains the most general defaults and P_N the most specific ones. Models are defined as in classical knowledge bases. A default $(A|B)$ falls into partition P_n iff there is a model for the

TBox and the remaining defaults [6] that satisfies $A(i)$ as well as $B(i)$ for a new individual i. We say that such a model *verifies* this default. A PTBox is consistent iff there exists a partition for the PBox.
Inferences are drawn according to the *lexicographical minimal model* for a PTBox, where models are ordered lexicographically w.r.t. the number and level of generality of defaults they violate. Models violating as few of the least specific defaults as possible have higher preference when ordering the models.

3.2 From DL to PDL: The Δ-Transformation

Using default logic for resolving an unsatisfiable concept of a TBox $\mathcal{T}$, we must transform a TBox into a PTBox and hence a subset of $\mathcal{T}$ into a set of defaults. We introduce the Δ-transformation for this transformation which changes the logics from $\mathcal{SHOIN}(D)$ to P-$\mathcal{SHOIN}(D)$.

Definition 3 (Δ-Transformation). *Let $\alpha = B \sqsubseteq A$ be a subclass inclusion axiom in a TBox $\mathcal{T}$, and $\mathcal{U}_n$ be a set of subclass inclusion axioms being a subset of $\mathcal{T}$. The Δ-transformation for $\mathcal{T}$ maps axioms from $\mathcal{T}$ to defaults and sets of axioms to a (partitioned) PBox.*

$$\begin{array}{llcl}
(i) & \Delta_{\mathcal{T}}(\alpha) & = & (A|B) \\
(ii) & \Delta_{\mathcal{T}}(\mathcal{U}_n) & = & \{\Delta_{\mathcal{T}}(\alpha) | \alpha \in \mathcal{U}_n\} \\
(iii) & \Delta_{\mathcal{T}}\ \underbrace{(\mathcal{U}_0, \ldots, \mathcal{U}_N)}_{pairwise\ disjoint} & = & (\underbrace{(\mathcal{T} \setminus \mathcal{U}_0 \cup \ldots \cup \mathcal{U}_N)}_{new\ TBox}, \underbrace{(\Delta(\mathcal{U}_0), \ldots, \Delta(\mathcal{U}_N))}_{partitioned\ PBox})
\end{array}$$

Please note that the Δ-transformation is bijective, i.e. we can easily define $\Delta_{\mathcal{T}}^{-1}(A|B) = B \sqsubseteq A$.

4 Constraints on the TBox

The most obvious method for resolving unsatisfiable concepts of a TBox is to remove all the axioms from the justifications proving the unsatisfiability. However, it clearly suffices to remove only all the axioms of the *root justifications* from the TBox, since any (purely) derived unsatisfiable concept will then also become satisfiable.
Removing axioms from the TBox results in a loss of knowledge. We therefore propose not to fully remove the axioms but to keep them in a different form, i.e. as defaults. The Δ-transformtion will be applied to all axioms of the root justifications for the unsatisfiable concepts of a TBox. In this section, it is shown that this transformation results in a consistent PTBox, if the TBox fulfils certain constraints which are explained in the remainder of this section. Conflict resolving using default logic works only if the axioms justifying the conflict are on different levels of preference, like it is implied by PDL. Situations where conflicting axioms are on the same level of preference must be excluded.

[6] $\mathcal{T} \cup (P \setminus P_0 \cup \ldots \cup P_{n-1})$

This means that all kinds of cycles and situtations where a concept is explicitely stated to be subclass of one concept and its negation must not be allowed, since the Δ-transformation will result in an inconsistent PTBox.

4.1 Disallow Cycles, Logical and Direct Contradictions

If we allowed for cycles in the TBox, then a justification may also contain this cycle. In turn, all axioms involved in the conflict are on the same level of preference and there cannot be a verifying model for any of these axioms.
In the following, we assume every TBox and hence all justifications, to be free of cycles. Logical contradictions, i.e. concepts of the form $A \sqcap \neg A$ cannot be resolved by applying the Δ-transformation. There is no valid world w.r.t. [5] for the concept $A \sqcap \neg A$.

Corollary 1. *If one of the axioms of a TBox $\mathcal{T}$ contains a logical contradiction on the right hand side, then the Δ-transformation of $\mathcal{T}$ is inconsistent.*

Since logical contradictions do not provide any useful information, we can safely remove axioms containing $A \sqcap \neg A$ from the TBox without changing the intended semantics. In the following, we assume every TBox not to contain any logical contradiction. Default logics require contradictive information to be on different level of preference in order to provide a consistent way for inference. This mechanism is doomed to fail in cases where a contradiction is stated explicitly, i.e. some concept C is explicitely stated to be a subclass of the concepts A_1 and A_2 where $A_1 \sqcap A_2$ are a logical contradiction.

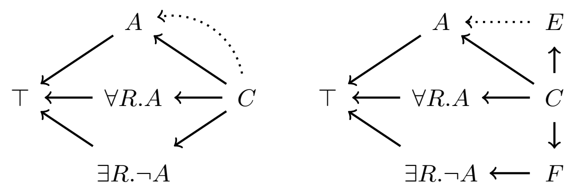
Definition 4 (Direct Contradictions).
A set of two axioms $\mathcal{DC} = \{C \sqsubseteq A_1, C \sqsubseteq A_2\}$ from a TBox $\mathcal{T}$ is called a direct contradiction *$\mathcal{DC}$ for a concept $C \in \mathcal{T}$, iff $A_1 \sqcap A_2$ is a logical contradiction.*

There exists some justification for $\mathcal{T} \models C \sqsubseteq \bot$ that soley consists of the axioms of the direct contradiction. Since there cannot exist a model that satisfies $A_1(i)$, $A_2(i)$ and $C(i)$ at the same time for a new individual i, the Δ-transformation of the example TBox will lead to an inconsistent PTBox. Hence default logics cannot resolve the unsatisfiability of C.

Corollary 2. *The Δ-transformation of a TBox that contains a direct contradiction results in an inconsistent PTBox.*

While logical contradictions can simply be removed from the TBox without loss of relevant information, the situation for a direct contradiction $\mathcal{DC}$ is slightly more difficult. Removing the axioms of the $\mathcal{DC}$ might lead to a loss of information. Yet there is an option how to resolve a $\mathcal{DC}$.
Considering PDL, we can simply add some new "intermediate" concept to at least one of the axioms of the direct contradiction. In particular, we replace, e.g., the axiom $C \sqsubseteq A_1 \in \mathcal{DC}$ with $C \sqsubseteq B$ and $B \sqsubseteq A_1$ where B is a new concept that does not yet occur in the TBox. The concept C is still unsatisfiable, but there is no direct contradiction anymore. We can therefore safely assume the TBox not

Fig. 2. Example TBox where C is inferred unsatisfiable due to the two direct contradictions $C \sqsubseteq A, \neg A$ and $C \sqsubseteq \forall R.A, \exists R.\neg A$ in the left figure (a).
One possibility for resolving the direct contradictions by adding a new "intermediate" concept is shown in the right figure (b).



(a) (b)

to contain any direct contradictions.
In the example in Figure 2(a) there exist the direct contradictions $\mathcal{DC}_1 = \{C \sqsubseteq A, C \sqsubseteq \neg A\}$ and $\mathcal{DC}_2 = \{C \sqsubseteq \forall R.A, C \sqsubseteq \exists R.\neg A\}$. These can be resolved, for example, by introducing the new concepts E and F in between the subconcept hierarchy of $C \sqsubseteq \neg A$ and $C \sqsubseteq \exists R.\neg A$, respectively.

5 Consistency of the Δ-Transformed TBox

After having excluded logical as well as direct contradictions and cycles, we have to show that the PTBox that results from Δ-transforming all axioms from all (root) justifications is consistent. According to [5], we have to show that the resulting PBox is a valid z-partiton. We do so soley using the structure of the justifications for the unsatisfiable concepts.
For each unsatisfiable concept U, we split the union of its justifications into two parts: one that contains unsatisfiable concepts in the axioms Γ_U and one that does not, Θ_U. The idea is to iteratively first transform axioms for a new partition that occur only in some Θ, but not in some Γ, since these are not in conflict with any other axiom.
Every axiom we transformed and hence removed from some Θ_U has its conflicting axioms its correponding Γ_U set. The conflict for Γ_U set is solved, if its Θ_U set is empty. The next partition is hence formed by all axioms in some $\Gamma_{U'}$ for which $\Theta_{U'}$ set is empty. We can now proceed with step one and, since the number of axioms is finite, the procedure will terminate eventually.

5.1 Splitting the Root Justifications

Every justification for an unsatisfiable concept U contains at least one axiom with U on the left-hand side of the subclass inclusion. As such, every justification can be split up into two sets of axioms: one that contains axioms with U on the left-hand-side and one that contains the rest. We call the first one the Γ-set of

Table 1. Procedure for Δ-transforming the unsatisfiability splitting for the root justifications for the non-purely derived unsatisfiable concepts in Figure 1. The last column shows the axioms that are chosen to be Δ-transformed to the partitions P_0, P_1, P_2 of the resulting PBox during the corresponding Θ- or Γ step (indicated by a bold symbol), whereas the $\mathcal{J}$-columns show the current contents of the Θ and Γ sets.

Step		$\mathcal{J}(C \sqsubseteq \bot)$	$\mathcal{J}(D \sqsubseteq \bot)$	$\mathcal{J}(B \sqcap F \sqsubseteq \bot)$	
1	$\boldsymbol{\Theta}$	2	$2, 3, 8$	8	$P_0 = \{\Delta_{\mathcal{T}}(\{2, 8\})\}$
	Γ	$3, 4$	$5, 6, 9$	11	- - - - - - - - - - - - - -
	Θ	$\emptyset$	3	$\emptyset$	- - - - - - - - - - - - - -
2	$\boldsymbol{\Gamma}$	$3, 4$	$5, 6, 9$	11	$P_1 = \{\Delta_{\mathcal{T}}(\{3, 4, 11\})\}$
3	$\boldsymbol{\Theta}$	$\emptyset$	$\emptyset$	$\emptyset$	- - - - - - - - - - - - - -
	Γ	$\emptyset$	$5, 6, 9$	$\emptyset$	- - - - - - - - - - - - - -
	Θ	$\emptyset$	$\emptyset$	$\emptyset$	- - - - - - - - - - - - - -
4	$\boldsymbol{\Gamma}$	$\emptyset$	$5, 6, 9$	$\emptyset$	$P_2 = \{\Delta_{\mathcal{T}}(\{5, 6, 9\})\}$
5	$\boldsymbol{\Theta}$	$\emptyset$	$\emptyset$	$\emptyset$	- - - - - - - - - - - - - -
	Γ	$\emptyset$	$\emptyset$	$\emptyset$	- - - - - - - - - - - - - -

U and the latter one its Θ-set. The splitting for the example of Figure 1 can be obtained from the first row of Table 1.

Definition 5 (Unsatisfiability Splitting).
Let $U_0, \ldots, U_N$ be the unsatisfiable concepts of a TBox $\mathcal{T}$. Let $\mathcal{J}^{root}_{U_i \sqsubseteq \bot}$ be the union of the root justifications for the unsatisfiability of the concept U_i. The unsatisfiability splitting *for $\mathcal{T}$ is defined as:*[7]
$\mathcal{J}^{root}_{U_i \sqsubseteq \bot} = \Theta_{U_i} \oplus \Gamma_{U_i}$ *where* $\Gamma_{U_i} = \{X \sqsubseteq Y \in \mathcal{J}^{root}_{U_i \sqsubseteq \bot} | X = U_i\}$

5.2 Obtaining the Partition by Δ-Transforming the Splitting

For an axiom of the root justifications, there exist three different possibilities where it my reside:

1. In some Θ_{U_j} but not in any Γ_{U_k}. We denote these axioms with ϑ.
2. In some Γ_{U_k} but not in any Θ_{U_j}. These axioms are denoted with γ.
3. In both some Θ_{U_j} as well as some Γ_{U_k}. In this case the axiom is denoted with η.

In our example, processing step one, axioms 2 and 8 are of the ϑ type whereas axioms $4, 5, 6$ and 9 are of type γ. Axiom 3 is of type η, since it is contained in both, Γ_C and Θ_D. For preparing the proof of the induction, we first proof some auxiliary lemma stating the important properties of ϑ, γ and η axioms.

Lemma 1 (Satisfiability of θ, γ and η axioms).

[7] The operator $\oplus$ denotes the union of pairwise disjoint sets.

1. *If some axiom ϑ is contained only in some Θ_{U_i} but not in some Γ_{U_j}, then there exists verifying model for $\Delta(\vartheta)$.*
2. *If some axiom γ is contained in some Γ_{U_i} for which the corresponding Θ_{U_i} is empty, then there exists verifying model for $\Delta(\gamma)$.*
3. *If some axiom η is contained in both, some Θ_{U_i} and some Γ_{U_j}, then there cannot exist a verifying model for $\Delta(\eta)$ w.r.t. these two sets.*

We explain the procedure using the example from Figure 1. We alternatively Δ-transform axioms according to 1, the so-called Θ-step, followed by the Γ-step where axioms are Δ-transformed according to 2. The single steps are visualized in Table 1.
In our example, step one, we can find a model for each ϑ axiom 2 and 8. In particular we can obviously find a model in which $A \sqcap B$ is satisfied and some model in which $B \sqcap \neg F$ is satisfied. On the other hand, all remaining axioms contain by definition some unsatisfiable concept, which denies the existence of a model for each of the remaining axioms. So even though axiom 3 is part of Θ_D we are not able to find a verifying model for it [8]. Hence, the first partition is $P_0 = \{2, 8\}$.
We proceed with the next step and have a look at the Γ sets for which the Θ set is empty. This is the case for Γ_C. We remember that Θ_{U_i} contains at least one element from each justification for $\mathcal{T} \models U_i \sqsubseteq \bot$. Hence, for each justification for $\mathcal{T} \models C \sqsubseteq \bot$ we Δ-transformed at least one axiom which means that $\mathcal{T} \setminus \Delta^{-1}(P_0) \not\models C \sqsubseteq \bot$. As a consequence, we can find a verifying model for each axiom in Γ_C. On the other hand, $D \sqsubseteq \bot$ still holds, such that we cannot find a verifying model for any of the remaining axioms $5, 6$ and 9. Hence, the second partition is $P_1 = \{3, 4, 11\}$.
For the next Θ-step we find that all of the Θ sets are empty, so we proceed with the next Γ-step and find that Γ_D is the only Γ-set left. Since all conflicting axioms have already been Δ-transformed, we can for each axiom in Γ_D trivially find a verifying model which results in the next partition $P_2 = \{5, 6, 9\}$.
In step nine, there are no more axioms left that we could process. The resulting PTBox is:

$$\mathrm{PT} = (\underbrace{(\{1, 10, 12\})}_{\mathcal{T}\setminus\Delta^{-1}(\mathcal{P})}, \underbrace{(\overbrace{\Delta(\{2, 8\})}^{P_0}, \overbrace{\Delta(\{3, 4, 11\})}^{P_1}, \overbrace{\Delta(\{5, 6, 9\})}^{P_2})}_{\mathcal{P}})$$

Since we found a valid partition w.r.t. PDL, PT is consistent.
We now proof the parts of Lemma 1.

Proof. 1 If some axiom ϑ is contained only in some Θ_U but not in some $\Gamma_{U'}$, then ϑ has no unsatisfiable concept on the left-hand side. It also cannot have an unsatisfiable concept on the right-hand side, because then it would be purely derived. As such, we can find a model in which both the subconcept and the superconcept are satisfied.

[8] Indeed, axiom 3 is of type η.

Proof. 2 If some axiom $\gamma = U \sqsubseteq A$ is contained in some $\mathit{\Gamma}_U$ for which the corresponding $\mathit{\Theta}_U$ is empty, there has been one axiom removed from every root justification for $\mathcal{T} \models U \sqsubseteq \bot$, i.e. the elements that had been in the now empty $\mathit{\Theta}_U$ and were $\mathit{\Delta}$-transformed before. Hence U is not unsatisfiable anymore and A must be satisfiable for the same reasons as in the proof for 1 which proofs the existence of a model.
It should be noted that in this case, γ has been root unsatisfiable and the $\mathit{\Delta}$-transformed axioms from the $\mathit{\Theta}_U$ were part of the root justifications.

Proof. 3 Some axiom η is contained in both, some $\mathit{\Theta}_U$ and some $\mathit{\Gamma}_{U'}$. Because $\eta \in \mathit{\Theta}_U$, there still exists some $\mathit{\Gamma}_U$ correspoding to $\mathit{\Theta}_U$, which means that there still exists a justification for $U \sqsubseteq \bot$. Hence, we cannot find a model for an axiom that contains an unsatisfiable concept.
It should be noted that in this case, $\eta = U \sqsubseteq A$ is part of a justification for some partially derived unsatisfiable concept.

5.3 Consistency of the $\mathit{\Delta}$-Transformation of the Splitting

It remains to show that we can always find some axioms that fulfil the conditions of the $\mathit{\Theta}$-step followed by a $\mathit{\Gamma}$ step. We do this by induction. As stated before, every justification can be split into non-empty $\mathit{\Theta}_U$ and $\mathit{\Gamma}_U$. Since the number of sets of the splitting of Definition 5 is finite, there has to exist some $\mathit{\Theta}_{U_0}$ such that for all axioms $\vartheta \in \mathit{\Theta}_{U_0}$ follows $\vartheta \notin \mathit{\Gamma}_U$. We $\mathit{\Delta}$-transform all of these axioms into the starting partition P_0 and proceed with all axioms γ of the sets $\mathit{\Gamma}_{U_0}$ that correspond to $\mathit{\Theta}_{U_0}$. By Lemma 1, part 2, these form the next partition P_1.
In the induction step we have to show that having transformed the $\mathit{\Gamma}$-axioms

1. either there is some $\mathit{\Theta}_{U_0}$ such that $\mathit{\Theta}_{U_0}$is disjoint to all remaining $\mathit{\Gamma}_U$,
2. or there is some $\mathit{\Gamma}_{U_0}$ for which all $\mathit{\Theta}$-axioms have already been $\mathit{\Delta}$-transformed
3. or there are no more axioms left to transform.

Since every $\mathit{\Gamma}_{U_i}$ refers to a $\mathit{\Theta}_{U_i}$, and since the number of sets is finite, and justifications cannot be circular, for at least one $\mathit{\Gamma}_{U_i}$ there has to exist some $\mathit{\Theta}_{U_i}$ that contains neither γ nor η axioms. Please note that we allow the case $\mathit{\Theta}_{U_i} = \emptyset$. In case $\mathit{\Theta}_{U_i}$ is non-empty we proceed with the $\mathit{\Theta}$-step, if it is empty, we proceed with the $\mathit{\Gamma}$-step. The procedure terminates, if also the $\mathit{\Gamma}$ sets are empty.

Theorem 1. *Let $\mathcal{T}$ be a TBox and $\mathcal{P} = (P_0, \ldots, P_N)$ be the partition resulting from the $\mathit{\Delta}$-transformation of the unsatisfiability splitting of all root justifications for all unsatisfiable concepts in $\mathcal{T}$.*
Then the PTBox $PT = (\mathcal{T} \setminus \mathit{\Delta}^{-1}(P_0, \ldots, P_N), \mathcal{P})$ is consistent.

5.4 Complexity of the $\mathit{\Delta}$-Transformation of the Splitting

The complexity of the presented procedure is dominated by the complexity for finding justifications. This in turn depends on the complexity for consistency checking in the tableaux calculus which is - in the case of $\mathcal{SHOIN}(D)$ -

NEXPTIME-complete.
It should be noted that the presented approach does not involve any satisfiability checks in addition to checking and tracing unsatisability, which have to be performed anyway.

6 Related Work

In recent years, much progess has been made in the task to explain why a conclusion can be drawn from a DL knowledge base by soley using axioms from the knowledge base itself. Schlobach and Cornet [8] came up with minimal unsatisfiable preserving sub-TBoxes (MUPS) which can explain the reason for unsatisfiability of concepts. Kalyanpur et al. [6] introduced justification as a form of minimal explanation for any arbitrary entailment. It could be shown that computing all justifications for an entailment is feasible in the tableaux calculus [6].
In the area of ontology evolution, the main focus usually lies on resolving inconsistencies and hence changes mainly occur on instance level or rather restricted TBoxes [9]. Repair can also be done using higher-order logics like in the Ontology Repair System [10]. This, however makes changes to the ontology and cannot be applied easily to OWL ontologies.
Alternatives to do reasoning with incoherent DL knowledge bases are, for example, paraconsistent logics [11]. However, these change the notion of inference and hence their semantics much more than default logic does.
There have been made propositions of how to incorporate default knowledge in OWL-DL knowledge bases in [12] [13], and [14]. While the first two deal with applcations of Reiter's interpretation of defaults, to our knowledge, P-$\mathcal{SHOIN}(D)$ [5] is currently the only formalism providing default reasoning services w.r.t. Lehmann's lexicographical entailment for OWL DL knowledge bases for which an implementation is available [15].

7 Conclusion

We showed that default logics as introduced in [5] provide a way of re-enabling coherency for incoherent DL knowledge bases. This way, structure as well as semantics of the original axioms is kept as much as possible. The proposed approach makes use of justifications, a standard technique for computing reasons for conflicts in DL knowledge bases. Since these have to be computed anyhow for repairing the knowledge base, the presented approach does not need to perform any additional satisfiability checks.
While this paper proofs the correctness of the approach, an implementation and evaluation on real-world data has to be performed showing whether the approach is feasible. Comparisons to alternative approaches, for example, what can still be inferred from the knowledge base after the repair and what not, will also remain for future work.

References

1. McGuinness, D.L., van Harmelen, F.: OWL web ontology language overview. W3C recommendation, W3C (February 2004) http://www.w3.org/TR/2004/REC-owl-features-20040210/.
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. 2nd edn. Cambridge University Press, Cambridge, MA, USA (August 2007)
3. Reiter, R.: A logic for default reasoning. Artificial Intelligence **13**(1-2) (1980) 81–132
4. Lehmann, D.: Another perspective on default reasoning. Ann. Math. Artif. Intell. **15** (1995) 61–82
5. Lukasiewicz, T.: Expressive probabilistic description logics. Artif. Intell. **172**(6-7) (April 2008) 852–883
6. Kalyanpur, A.: Debugging and Repair of OWL Ontologies. PhD thesis, University of Maryland, Department of Computer Science (2006)
7. Reiter, R.: A theory of diagnosis from first principles. Artif. Intell. **32**(1) (1987) 57–95
8. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: IJCAI, Morgan Kaufmann (2003) 355–362
9. Haase, P., Völker, J.: Ontology learning and reasoning - dealing with uncertainty and inconsistency. In: URSW Pt. 1. Volume 5327 of LNCS. (January 2009) 45–55
10. Bundy, A.: Where's my stuff? an ontology repair plan. In: Workshop on DISPROVING - Non-Theorems, Non-Validity, Non-Provability. Volume 4., CADE Inc (July 2007) 2–11
11. Ma, Y., Lin, Z., Lin, Z.: Inferring with inconsistent owl dl ontology: A multi-valued logic approach. In: EDBT Workshops. Volume 4254 of LNCS. (March 2006) 535–553
12. Baader, F., Hollunder, B.: Embedding defaults into terminological knowledge representation formalisms. In: J. Autom. Reas., Morgan Kaufmann (1995) 306–317
13. Dao-Tran, M., Eiter, T., Krennwallner, T.: Realizing default logic over description logic knowledge bases. In: ECSQARU 2009. 602–613
14. Navarro, J.L., Sanchez, J.M., Zurita, J.M.: Default reasoning in web ontology language. In: Proc. Intell. Systems and Agents (IADIS2007). (July 2007) 35–42
15. Klinov, P.: Pronto: A non-monotonic probabilistic description logic reasoner. In: ESWC. Volume 5021 of LNCS., Springer (2008) 822–826