Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Practical TBox Abduction Based on Justification Patterns

Jianfeng Du,[1,2] **Hai Wan,**[3,4*] **Huaguan Ma**[3]
[1]Collaborative Innovation Center for 21st-century Maritime Silk Road Studies,
Guangdong University of Foreign Studies, Guangzhou 510420, P.R.China
[2]Cisco School of Informatics, Guangdong University of Foreign Studies, Guangzhou 510006, P.R.China
jfdu@gdufs.edu.cn
[3]School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, P.R.China
[4]Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou 510006, P.R.China
wanhai@mail.sysu.edu.cn

Abstract

TBox abduction explains why an observation is not entailed by a TBox, by computing multiple sets of axioms, called *explanations*, such that each explanation does not entail the observation alone while appending an explanation to the TBox renders the observation entailed but does not introduce incoherence. Considering that practical explanations in TBox abduction are likely to mimic minimal explanations for TBox entailments, we introduce *admissible explanations* which are subsets of those justifications for the observation that are instantiated from a finite set of justification patterns. A justification pattern is obtained from a minimal set of axioms responsible for a certain atomic concept inclusion by replacing all concept (resp. role) names with concept (resp. role) variables. The number of admissible explanations is finite but can still be so large that computing all admissible explanations is impractical. Thus, we introduce a variant of subset-minimality, written $\subseteq_{\mathsf{ds}}$-minimality, which prefers fresh (concept or role) names than existing names. We propose efficient methods for computing all admissible $\subseteq_{\mathsf{ds}}$-minimal explanations and for computing all justification patterns, respectively. Experimental results demonstrate that combining the proposed methods is able to achieve a practical approach to TBox abduction.

Introduction

Ontologies have been used in many real-life applications, including e-Commerce, medical informatics, bio-informatics, and the Semantic Web. As a popular formalism for expressing ontologies, description logics (DLs) (Baader et al. 2003) underpin the standard Web Ontology Language (OWL) (Horrocks, Patel-Schneider, and van Harmelen 2003). A DL ontology is often expressed as a knowledge base consisting of both schema information in the *TBox* and data information in the *ABox*. TBox abduction, advocated in (Elsenbroich, Kutz, and Sattler 2006), is a pragmatic formalism for abductive reasoning in DLs. Given a coherent TBox and an observation which is a concept inclusion not entailed by the TBox, TBox abduction often explains the observation by computing multiple sets of axioms called *explanations* such that each explanation does not entail the observation alone while appending an explanation to the TBox renders the observation entailed but does not introduce incoherence.

TBox abduction is particularly useful in handling missing is-a relations (Lambrix, Dragisic, and Ivanova 2012; Lambrix and Liu 2013). An is-a relation is a concept inclusion between two concept names, called an *atomic concept inclusion* (*ACI*) in this paper. Simply adding the missing is-a relations to the TBox may not be a proper way to revise the TBox since some crucial axioms are still missing. A better way is to find essential causes of the missing is-a relations and add axioms in these causes to the TBox. Take a TBox $\mathcal{T}_1$ consisting of the following axioms $\alpha_1, \ldots, \alpha_6$ for example.

α_1: $\mathsf{GrandFather} \sqsubseteq \mathsf{Man}$
α_2: $\mathsf{GrandFather} \sqsubseteq \exists \mathsf{hasChild}.\exists \mathsf{hasChild}.\mathsf{Human}$
α_3: $\mathsf{GrandMother} \sqsubseteq \mathsf{Woman}$
α_4: $\mathsf{GrandMother} \sqsubseteq \exists \mathsf{hasChild}.\exists \mathsf{hasChild}.\mathsf{Human}$
α_5: $\mathsf{Man} \sqcap \exists \mathsf{hasChild}.\mathsf{Human} \sqsubseteq \mathsf{Father}$
α_6: $\mathsf{Woman} \sqcap \exists \mathsf{hasChild}.\mathsf{Human} \sqsubseteq \mathsf{Mother}$

Suppose there is a missing is-a relation $\mathsf{GrandFather} \sqsubseteq \mathsf{Father}$ and we want to make it entailed by $\mathcal{T}_1$. Although simply adding this relation to $\mathcal{T}_1$ will work, this addition cannot make another missing is-a relation $\mathsf{GrandMother} \sqsubseteq \mathsf{Mother}$ entailed by $\mathcal{T}_1$. This means that such an addition does not sufficiently improve the quality of $\mathcal{T}_1$. In contrast, adding a crucial axiom $\exists \mathsf{hasChild}.\mathsf{Human} \sqsubseteq \mathsf{Human}$ to $\mathcal{T}_1$ will make the above two missing is-a relations entailed by $\mathcal{T}_1$. TBox abduction is required to find such crucial axioms.

Most existing approaches to TBox abduction adopt simple representation languages for explanations, such as ACI (Lambrix, Dragisic, and Ivanova 2012; Lambrix and Liu 2013; Wei-Kleiner, Dragisic, and Lambrix 2014) or slightly extended ACI that allows atomic negation and unqualified existential or universal restriction (Halland, Britz, and Klarman 2014). These representation languages may discard essential causes of non-entailments. For the previous example, if we disallow qualified existential restrictions, we cannot discover the explanation $\{\exists \mathsf{hasChild}.\mathsf{Human} \sqsubseteq \mathsf{Human}\}$ for the missing is-a relation $\mathsf{GrandFather} \sqsubseteq \mathsf{Father}$ in $\mathcal{T}_1$.

A more practical way is to allow more DL constructors including qualified existential restriction to appear in explanations. However, a representation language supporting qualified existential restriction will yield infinitely many explanations since it allows infinitely many nested role names to appear in a concept. To guarantee a finite space for explanations without losing DL expressivity, we consider patterns.

*Corresponding author

All entailments have minimal explanations of limited forms, which are often referred to as *justifications*. A justification is a minimal set of axioms responsible for the entailment. It can be lifted to a *justification pattern* by replacing all different concept (resp. role) names with different concept (resp. role) variables. Consider a TBox $\mathcal{T}_2$ obtained from the aforementioned TBox $\mathcal{T}_1$ by adding $\alpha_7 : \exists\mathsf{hasChild.Human} \sqsubseteq \mathsf{Human}$. A justification for $\mathsf{GrandMother} \sqsubseteq \mathsf{Mother}$ in $\mathcal{T}_2$ is $\{\alpha_3, \alpha_4, \alpha_6, \alpha_7\}$, which can be lifted to a justification pattern $\mathcal{J}_p = \{X \sqsubseteq Z, X \sqsubseteq \exists U.\exists U.W, Z \sqcap \exists U.W \sqsubseteq Y, \exists U.W \sqsubseteq W\}$ for the ACI template $X \sqsubseteq Y$.

To mimic minimal explanations for entailments, it can be assumed that a practical explanation $\mathcal{E}$ for a non-entailment α in a TBox $\mathcal{T}$ is a subset of some justification for α in $\mathcal{T} \cup \mathcal{E}$. By considering that any justification can be instantiated from a justification pattern by a ground substitution which replaces all different concept (resp. role) variables with different concept (resp. role) names, we assume that only a finite set $\mathcal{P}$ of justification patterns occurring frequently in the domain of interest is used to yield such an explanation $\mathcal{E}$. We call $\mathcal{E}$ an *admissible explanation* w.r.t. $\mathcal{P}$. For example, given the aforementioned $\mathcal{T}_1$ and $\mathcal{J}_p$, $\{\exists\mathsf{hasChild.Human} \sqsubseteq \mathsf{Human}\}$ is an admissible explanation for both observations $\mathsf{GrandFather} \sqsubseteq \mathsf{Father}$ and $\mathsf{GrandMother} \sqsubseteq \mathsf{Mother}$ w.r.t. $\{\mathcal{J}_p\}$. It shows that finding admissible explanations is able to discover essential causes of non-entailments.

An admissible explanation is allowed to involve *fresh* concept or role names that do not appear in the TBox nor in the observation. These fresh concept (resp. role) names can be interpreted as arbitrary concept (resp. role) names in a domain whose vocabulary is larger than that of the given TBox. Although the set of different admissible explanations (up to renaming fresh concept or role names) is finite, the number of different admissible explanations can still be so large that computing all of them is impractical. To compute less admissible explanations, we introduce a novel notion of minimality, called $\subseteq_{\mathsf{ds}}$-*minimality*. By treating fresh (concept or role) names as variables, we call an explanation $\mathcal{E}$ $\subseteq_{\mathsf{ds}}$-*minimal* if there is no explanation $\mathcal{E}'$ such that $\mathcal{E}' \subseteq_{\mathsf{ds}} \mathcal{E}$ and $\mathcal{E} \not\subseteq_{\mathsf{ds}} \mathcal{E}'$, where we write $\mathcal{E}' \subseteq_{\mathsf{ds}} \mathcal{E}$ if there is a *distinguishable substitution* θ for $\mathcal{E}'$, which maps different fresh names to different fresh names or different existing names not occurring in $\mathcal{E}'$, such that $\mathcal{E}'\theta \subseteq \mathcal{E}$.

We simply call an admissible $\subseteq_{\mathsf{ds}}$-minimal explanation a *justification pattern based* or ***JP-based*** explanation. The primary challenge for computing JP-based explanations lies in discovering justification patterns. Computing all justifications in a TBox before discovering justification patterns is infeasible. We propose an efficient method that alternately computes justifications and justification patterns so that justification patterns can be generated as early as possible. The method calls a DL reasoner as black-box and is applied to any DL. Moreover, we propose an efficient method for computing all JP-based explanations from given justification patterns in a coherent TBox. To show that the proposed methods can constitute a practical approach to TBox abduction, we conduct experiments on ten coherent TBoxes with various complexity. Full proofs are available at http://www.dataminingcenter.net/tboxabd/AAAI17.pdf.

Preliminaries

We briefly introduce DLs by taking the DL $\mathcal{SROIQ}$ (Horrocks, Kutz, and Sattler 2006) for example. $\mathcal{SROIQ}$ is an expressive DL corresponding to OWL 2 DL, the most expressive while decidable species in the OWL family. A $\mathcal{SROIQ}$ ontology is composed of a TBox and an ABox, both of which are sets of axioms. TBox axioms include *concept inclusions* $C \sqsubseteq D$, *complex role inclusions* $r_1 \circ \ldots \circ r_n \sqsubseteq s$, *role disjointness assertions* $r \sqsubseteq \neg s$ and *role property assertions* that are expressible by other axioms: a role r being *symmetric* amounts to $r \sqsubseteq r^-$; r being *transitive* amounts to $r \circ r \sqsubseteq r$; r being *reflexive* amounts to $\top \sqsubseteq \exists r.\mathsf{Self}$; r being *irreflexive* amounts to $\exists r.\mathsf{Self} \sqsubseteq \bot$. The semantics for $\mathcal{SROIQ}$ is given by *interpretations* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ for $\Delta^{\mathcal{I}}$ the domain of $\mathcal{I}$ and $\cdot^{\mathcal{I}}$ a mapping function. A concept inclusion $C \sqsubseteq D$ (resp. a complex role inclusion $r_1 \circ \ldots \circ r_n \sqsubseteq s$, or a role disjointness assertion $r \sqsubseteq \neg s$) is said to be *satisfied by* an interpretation $\mathcal{I}$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (resp. $r_1^{\mathcal{I}} \circ \ldots \circ r_n^{\mathcal{I}} \subseteq s^{\mathcal{I}}$ or $r^{\mathcal{I}} \cap s^{\mathcal{I}} = \emptyset$). An interpretation $\mathcal{I}$ is called a *model* of a TBox $\mathcal{T}$ if it satisfies all axioms in $\mathcal{T}$. A concept C is said to be *unsatisfiable* in $\mathcal{T}$ if $C^{\mathcal{I}} = \emptyset$ for all models $\mathcal{I}$ of $\mathcal{T}$. A TBox $\mathcal{T}$ is said to be *coherent* if all concept names in it are satisfiable. An axiom α is said to be *entailed by* $\mathcal{T}$ or an *entailment* of $\mathcal{T}$, written $\mathcal{T} \models \alpha$, if α is satisfied by all models of $\mathcal{T}$. Given an entailment α of $\mathcal{T}$, a subset $\mathcal{J}$ of $\mathcal{T}$ is called a *justification* for α in $\mathcal{T}$ if $\mathcal{J} \models \alpha$ and $\mathcal{J}' \not\models \alpha$ for all proper subsets $\mathcal{J}'$ of $\mathcal{J}$. By $\mathsf{Jst}(\alpha, \mathcal{T})$ we denote the set of justifications for α in $\mathcal{T}$.

The Semantics for JP-Based Explanations

Given a coherent TBox and an observation which is a concept inclusion not entailed by the TBox, TBox abduction often explains the observation by computing *explanations* for it, formally defined below. The requirement that an explanation for a concept inclusion $C \sqsubseteq D$ is coherent prevents meaningless explanations $\{C \sqsubseteq \bot\}$ from being considered.

Definition 1 Given a coherent TBox $\mathcal{T}$ and a concept inclusion α such that $\mathcal{T} \not\models \alpha$ and $\mathcal{T} \cup \{\alpha\}$ is coherent, an *explanation* $\mathcal{E}$ for α in $\mathcal{T}$ is a set of axioms such that $\mathcal{T} \cup \mathcal{E} \models \alpha$, $\mathcal{E} \not\models \alpha$ (simply say $\mathcal{E}$ is *nontrivial*) and $\mathcal{T} \cup \mathcal{E}$ is coherent (simply say $\mathcal{E}$ is *coherent*).

W.l.o.g. we assume that observations in TBox abduction are ACIs. This assumption does not impair applicability of TBox abduction because the set of explanations for $C \sqsubseteq D$ in $\mathcal{T}$ for C and D general concepts coincides with the set of explanations for $P_C \sqsubseteq P_D$ in $\mathcal{T} \cup \{C \sqsubseteq P_C, P_D \sqsubseteq D\}$, where P_C and P_D are globally unique fresh concept names.

To guarantee a finite space for explanations without losing DL expressivity, we consider patterns for explanations. We notice that justifications for an entailment are irreducible sets of axioms that are responsible for the entailment. Hence we assume that an explanation for an ACI in TBox abduction has the same pattern as a justification for the ACI. To define patterns for justifications, we introduce *variational axioms* in which variables may appear. A *variational axiom* is obtained from an axiom by replacing some concept (resp. role) names with concept (resp. role) variables. It is called a *fully variational axiom* if no concept or role names appear in it. A fully variational axiom does not alter the top (bottom) concept $\top$ ($\bot$) and individuals in its original axiom.

To make a pattern generalize as many justifications for ACIs as possible, we use a set of fully variational axioms to denote a pattern for justifications. This pattern should have some correspondence to a justification. Thus, we introduce *substitutions* for variational axioms, which map concept variables to concept names or other concept variables and map role variables to role names or other role variables. A substitution is said to be *ground* if it maps concept (resp. role) variables to concept (resp. role) names only.

One may only confine that a pattern for justifications generalizes a justification by applying some ground substitution. However, this restriction does not reject arbitrarily large patterns. For example, consider the justification $\mathcal{J} = \{A \sqsubseteq B\}$ for the ACI $A \sqsubseteq B$. It can be seen that $\mathcal{J}_p = \{X_1 \sqsubseteq Y_1, X_2 \sqsubseteq Y_2, \ldots, X_n \sqsubseteq Y_n\}$ for n an arbitrarily large integer is a pattern generalizing $\mathcal{J}$, because $\theta = \{X_i \mapsto A, Y_i \mapsto B \mid 1 \leq i \leq n\}$ is a ground substitution for $\mathcal{J}_p$ such that $\mathcal{J}_p\theta = \mathcal{J}$. To reject arbitrarily large patterns, we introduce *differentiated substitutions* which map different concept (resp. role) variables to different concept (resp. role) names. Intuitively, a differentiated substitution is a ground substitution that keeps a one-to-one correspondence between variables and names. We confine that a pattern generalizes at least one justification by applying some differentiated substitution. Moreover, an ideal pattern should generalize justifications only, hence we also confine that a pattern for the ACI template $X \sqsubseteq Y$ keeps entailment of $X\sigma \sqsubseteq Y\sigma$ for all ground substitutions σ. We call a pattern satisfying the above restrictions a *justification pattern*.

Definition 2 A set $\mathcal{J}_p$ of fully variational axioms is called a *justification pattern* for $X \sqsubseteq Y$ in $\mathcal{T}$ for X and Y different concept variables, if (1) there is a differentiated substitution θ for $\mathcal{J}_p$ such that $\mathcal{J}_p\theta \in \mathsf{Jst}(X\theta \sqsubseteq Y\theta, \mathcal{T})$, and (2) $\mathcal{J}_p\sigma \models X\sigma \sqsubseteq Y\sigma$ for all ground substitutions σ for $\mathcal{J}_p$.

Consider the pattern $\mathcal{J}_p = \{X_1 \sqsubseteq Y_1, X_2 \sqsubseteq Y_2, \ldots, X_n \sqsubseteq Y_n\}$ for $n \geq 2$ again. As required, Definition 2 prevents $\mathcal{J}_p$ from being a justification pattern for any $X_i \sqsubseteq Y_j$ (where $1 \leq i, j \leq n$) in any TBox $\mathcal{T}$.

By $\mathsf{lift}(S, A, B)$ we denote the set of fully variational axioms obtained from a set S of axioms by replacing the concept name A with X, the concept name B with Y, other concept names C with concept variables Z_C and role names r with role variables U_r. The following proposition provides a method for generating justification patterns for the ACI template $X \sqsubseteq Y$ from justifications for ACIs.

Proposition 1 *Let $\mathcal{T}$ be a TBox, $A \sqsubseteq B$ an ACI entailed by $\mathcal{T}$, and $\mathcal{J}$ a justification for $A \sqsubseteq B$ in $\mathcal{T}$. Then $\mathsf{lift}(\mathcal{J}, A, B)$ is a justification pattern for $X \sqsubseteq Y$ in $\mathcal{T}$.*

To make explanations $\mathcal{E}$ for an ACI α in a coherent TBox $\mathcal{T}$ mimic the way justification patterns are generated, we restrict $\mathcal{E}$ to be a subset of a certain justification for α in $\mathcal{T} \cup \mathcal{E}$, where this justification is obtained from a given justification pattern by applying some differentiated substitution.

Definition 3 Given a coherent TBox $\mathcal{T}$, a set $\mathcal{P}$ of justification patterns for $X \sqsubseteq Y$ and an ACI observation α such that $\mathcal{T} \not\models \alpha$ and $\mathcal{T} \cup \{\alpha\}$ is coherent, an explanation $\mathcal{E}$ for α in $\mathcal{T}$ is said to be *admissible* w.r.t. $\mathcal{P}$ if there is a justification pattern $\mathcal{J}_p \in \mathcal{P}$ and a differentiated substitution θ for $\mathcal{J}_p$ such that $\alpha = X\theta \sqsubseteq Y\theta$, $\mathcal{E} \subseteq \mathcal{J}_p\theta$ and $\mathcal{J}_p\theta \in \mathsf{Jst}(\alpha, \mathcal{T} \cup \mathcal{E})$.

Example 1 Consider the justification pattern $\mathcal{J}_p = \{X \sqsubseteq Z, X \sqsubseteq \exists U.\exists U.W, Z \sqcap \exists U.W \sqsubseteq Y, \exists U.W \sqsubseteq W\}$ for $X \sqsubseteq Y$. Suppose $n \geq 2$, $\mathcal{T} = \{A \sqsubseteq \exists r.\exists r.C, \exists r.C \sqsubseteq C, D_1 \sqsubseteq B\} \cup \{D_i \sqsubseteq C \mid 2 \leq i \leq n\}$, $\mathcal{P} = \{\mathcal{J}_p\}$ and the observation is $A \sqsubseteq B$. Then $\mathcal{E} = \{A \sqsubseteq D_1, D_1 \sqcap \exists r.C \sqsubseteq B\}$ is an admissible explanation for $A \sqsubseteq B$ in $\mathcal{T}$ w.r.t. $\mathcal{P}$, because $\mathcal{T} \cup \mathcal{E} \models A \sqsubseteq B$, $\mathcal{E} \not\models A \sqsubseteq B$, $\mathcal{T} \cup \mathcal{E}$ is coherent and $\theta = \{X \mapsto A, Y \mapsto B, Z \mapsto D_1, W \mapsto C, U \mapsto r\}$ is a differentiated substitution for $\mathcal{J}_p$ such that $A \sqsubseteq B = X\theta \sqsubseteq Y\theta$, $\mathcal{E} \subseteq \mathcal{J}_p\theta$ and $\mathcal{J}_p\theta \in \mathsf{Jst}(A \sqsubseteq B, \mathcal{T} \cup \mathcal{E})$.

Users of abduction often prefer explanations $\mathcal{E}$ that are *subset-minimal*, i.e., any proper subset of $\mathcal{E}$ is not an explanation. Admissible explanations are not necessarily subset-minimal. For example, the admissible explanation $\mathcal{E}$ in Example 1 is not subset-minimal because $\mathcal{E}' = \{A \sqsubseteq D_1\}$ is an explanation for $A \sqsubseteq B$ in $\mathcal{T}$ such that $\mathcal{E}' \subset \mathcal{E}$. We call a subset-minimal explanation an *admissible subset-minimal explanation* if it is also admissible.

Since admissible subset-minimal explanations do not contain axioms in the given TBox, it is natural to allow fresh names to appear in these explanations. We call a concept or role name a *fresh* name if it does not appear in the given TBox nor in the given observation, or an *existing* name otherwise. Consider Example 1. $\mathcal{E}_1 = \{A \sqsubseteq C_Z, C_Z \sqcap \exists r.C \sqsubseteq B\}$ is an admissible subset-minimal explanation for $A \sqsubseteq B$ in $\mathcal{T}$ w.r.t. $\mathcal{P}$, where C_Z is a fresh concept name.

The introduction of fresh names enables a compact representation for all admissible subset-minimal explanations. We treat fresh names as variables and define a *substitution* for explanations as a mapping from fresh names to existing or fresh names. The set of admissible subset-minimal explanations can be compacted using substitutions. Consider Example 1 again. $\mathcal{E}_i = \{A \sqsubseteq D_i, D_i \sqcap \exists r.C \sqsubseteq B\}$ for $i \in \{2, \ldots, n\}$ are also admissible subset-minimal explanations for $A \sqsubseteq B$ in $\mathcal{T}$ w.r.t. $\mathcal{P}$. The set of admissible subset-minimal explanations $\{\mathcal{E}_i \mid 1 \leq i \leq n\}$ for $A \sqsubseteq B$ in $\mathcal{T}$ w.r.t. $\mathcal{P}$ can be compacted as $\{\mathcal{E}_1\}$ and recovered from $\{\mathcal{E}_1\}$ by applying certain substitutions.

To reduce the number of explanations that need to be computed, one may consider computing only those explanations that cannot be obtained from others by applying substitutions. However, the verification of such explanations needs to consider longer explanations (i.e. those explanations having larger cardinalities) and thus is inefficient. For example, the explanation $\{A \sqsubseteq \exists r.A\}$ can be obtained from a longer explanation $\{A \sqsubseteq \exists r.C_{X_1}, C_{X_1} \sqsubseteq \exists r.C_{X_2}, \ldots, C_{X_{n-1}} \sqsubseteq \exists r.C_{X_n}\}$ by applying $\theta = \{C_{X_i} \mapsto A \mid 1 \leq i \leq n\}$, where $n \geq 2$ and C_{X_i} are fresh. To improve efficiency, we introduce the notion of *distinguishable* substitutions. A *distinguishable substitution* θ for $\mathcal{E}$ is a substitution for $\mathcal{E}$ that maps different fresh names to different fresh names or different existing names not occurring in $\mathcal{E}$. Intuitively, the set of concept (resp. role) names in $\mathcal{E}\theta$ has a one-to-one correspondence to the set of concept (resp. role) names in $\mathcal{E}$. For the above example, θ is not a distinguishable substitution for $\mathcal{E}$ since it maps C_{X_1} to A which appears in $\mathcal{E}$.

For two explanations $\mathcal{E}$ and $\mathcal{E}'$, we write $\mathcal{E}' \subseteq_{\mathsf{ds}} \mathcal{E}$ if there is a distinguishable substitution θ for $\mathcal{E}'$ such that $\mathcal{E}'\theta \subseteq \mathcal{E}$, or write $\mathcal{E}' \not\subseteq_{\mathsf{ds}} \mathcal{E}$ otherwise. We call an explanation $\mathcal{E}$ for an observation α in a TBox $\mathcal{T}$ $\subseteq_{\mathsf{ds}}$-*minimal* if there is no explanation $\mathcal{E}'$ for α in $\mathcal{T}$ such that $\mathcal{E}' \subseteq_{\mathsf{ds}} \mathcal{E}$ and $\mathcal{E} \not\subseteq_{\mathsf{ds}} \mathcal{E}'$. The following proposition shows that $\subseteq_{\mathsf{ds}}$-minimality implies subset-minimality.

Proposition 2 *A $\subseteq_{\mathsf{ds}}$-minimal explanation for an axiom α in a TBox $\mathcal{T}$ is a subset-minimal explanation for α in $\mathcal{T}$.*

The checking of $\subseteq_{\mathsf{ds}}$-minimality does not need to consider longer explanations. In fact, it only needs to consider n explanations that are not longer than $\mathcal{E}$, where n is the sum of the number of axioms in $\mathcal{E}$ and the number of existing names in $\mathcal{E}$. To show this checking method, we introduce two variant sets for a set S of axioms. Firstly, we define a *closest lift set* of S as a set of axioms obtained from S by replacing an existing name with a fresh name not occurring in S. By $\mathbf{lifts_1}(S)$ we denote the set of closest lift sets of S. For example, there are two closest lift sets of $\{A \sqsubseteq \exists r.A\}$, namely $\{A \sqsubseteq \exists r_U.A\}$ and $\{C_X \sqsubseteq \exists r.C_X\}$, where r_U is a fresh role name and C_X is a fresh concept name. Secondly, we define a *closest proper subset* of S as a proper subset of S that consists of all but one axioms in S. By $\mathbf{subs_1}(S)$ we denote the set of closest proper subsets of S. The following proposition shows that checking $\subseteq_{\mathsf{ds}}$-minimality can be reduced to linearly many entailment tests w.r.t. the size of $\mathcal{E}$.

Proposition 3 *An explanation $\mathcal{E}$ for an axiom α in a TBox $\mathcal{T}$ is a $\subseteq_{\mathsf{ds}}$-minimal explanation for α in $\mathcal{T}$ if and only if $\mathcal{T} \cup \mathcal{E}' \not\models \alpha$ for all $\mathcal{E}' \in \mathbf{subs_1}(\mathcal{E}) \cup \mathbf{lifts_1}(\mathcal{E})$.*

By combining $\subseteq_{\mathsf{ds}}$-minimality with admissibility, we obtain a new class of explanations named *justification pattern based* (simply *JP-based*) explanations, defined below.

Definition 4 A $\subseteq_{\mathsf{ds}}$-minimal explanation for α in $\mathcal{T}$ is said to be a *justification pattern based* (simply *JP-based*) explanation for α in $\mathcal{T}$ w.r.t. $\mathcal{P}$ if it is admissible w.r.t. $\mathcal{P}$.

We call two explanations *different* if they cannot be converted to each other by renaming fresh names. We propose to compute all different JP-based explanations rather than all different admissible subset-minimal explanations, because (1) by Proposition 2, a JP-based explanation must be admissible and subset-minimal, and (2) the number of different JP-based explanations can be much smaller than the number of different admissible subset-minimal explanations.

How to Compute JP-based Explanations

Proposition 1 has indicated that we are able to compute justification patterns from a given TBox by first computing all justifications for ACIs entailed by the TBox and then lifting computed justifications to justification patterns. However, this method is infeasible because computing all justifications for a single ACI is already time consuming (Kalyanpur et al. 2007; Du, Qi, and Fu 2014) even if it is optimized by modularization (Suntisrivaraporn et al. 2008). A practical way should compute justification patterns as early as possible.

Algorithm 1 ComputeJustificationPatterns($\mathcal{T}$)

1: $\mathcal{P} \leftarrow \emptyset$ *// $\mathcal{P}$ stores justification patterns for $X \sqsubseteq Y$*
2: **for all** ACIs $A \sqsubseteq B$ such that $A \neq B$ and $\mathcal{T} \models A \sqsubseteq B$ **do**
3: $\quad \mathcal{S} \leftarrow \{\mathcal{J}_p\theta \mid \mathcal{J}_p \in \mathcal{P}$, θ is a differentiated substitution for $\mathcal{J}_p$ such that $X\theta = A$, $Y\theta = B$, $\mathcal{J}_p\theta \subseteq \mathcal{T}$ and $\mathcal{J} \not\models A \sqsubseteq B$ for all $\mathcal{J} \in \mathbf{subs_1}(\mathcal{J}_p\theta)\}$
4: $\quad$ **while** $\mathcal{S} = \emptyset$ (then H is set as $\emptyset$) or there is a minimal hitting set H for $\mathcal{S}$ such that $\mathcal{T} \setminus H \models A \sqsubseteq B$ **do**
5: $\quad\quad \mathcal{J} \leftarrow$ FindJustification($\emptyset$, $\mathcal{T} \setminus H$, $A \sqsubseteq B$)
6: $\quad\quad \mathcal{J}_p \leftarrow \mathsf{lift}(\mathcal{J}, A, B)$
7: $\quad\quad \mathcal{P} \leftarrow \mathcal{P} \cup \{\mathcal{J}_p\}$
8: $\quad\quad \mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{J}_p\theta \mid \theta$ is a differentiated substitution for $\mathcal{J}_p$ such that $X\theta = A$, $Y\theta = B$, $\mathcal{J}_p\theta \subseteq \mathcal{T}$ and $\mathcal{J} \not\models A \sqsubseteq B$ for all $\mathcal{J} \in \mathbf{subs_1}(\mathcal{J}_p\theta)\}$
9: $\quad$ **end while**
10: **end for**
11: **return** $\mathcal{P}$

Function FindJustification(S_u, S_c, $A \sqsubseteq B$) *// Return a minimal subset S of S_c such that $S_u \cup S \models A \sqsubseteq B$.*

1: **if** $S_c = \emptyset$ or $S_u \models A \sqsubseteq B$ **then**
2: $\quad$ **return** $\emptyset$
3: **else if** $|S_c| = 1$ **then**
4: $\quad$ **return** S_c
5: **else**
6: $\quad$ Divide S_c into two disjoint subsets S_1 and S_2 such that $S_1 \cup S_2 = S_c$, $S_1 \cap S_2 = \emptyset$ and $0 \leq |S_1| - |S_2| \leq 1$
7: $\quad \Delta_2 \leftarrow$ FindJustification($S_u \cup S_1$, S_2, $A \sqsubseteq B$)
8: $\quad \Delta_1 \leftarrow$ FindJustification($S_u \cup \Delta_2$, S_1, $A \sqsubseteq B$)
9: $\quad$ **return** $\Delta_1 \cup \Delta_2$
10: **end if**

Thus, we propose a method that alternately computes justifications and justification patterns, formally shown in Algorithm 1. This method is extended from the method proposed in (Du, Qi, and Fu 2014) by adapting it to an incremental one and by embedding the computation of justification patterns.

The main function ComputeJustificationPatterns($\mathcal{T}$) in Algorithm 1 handles every entailed ACI one by one. For an ACI $A \sqsubseteq B$ entailed by $\mathcal{T}$, it computes all justifications for $A \sqsubseteq B$ that can be retrieved from computed justification patterns for $X \sqsubseteq Y$ before computing new justifications (line 3). It then incrementally computes a new justification for $A \sqsubseteq B$ using the same search strategy given by (Du, Qi, and Fu 2014). A new justification for $A \sqsubseteq B$ exists if and only if there is a *minimal hitting set* (*MHS*) H for the set of computed justifications such that $\mathcal{T} \setminus H \models A \sqsubseteq B$, where an MHS H for a set $\mathcal{S}$ of sets of axioms is a minimal set of axioms such that $H \cap S \neq \emptyset$ for all $S \in \mathcal{S}$. When there is an MHS H such that $\mathcal{T} \setminus H \models A \sqsubseteq B$, a new justification for $A \sqsubseteq B$ in $\mathcal{T}$ is computed from $\mathcal{T} \setminus H$ by a applying divide-and-conquer method developed in (Junker 2004; Du, Qi, and Fu 2014) (line 5). After a new justification $\mathcal{J}$ is computed, it is lifted to a justification pattern $\mathcal{J}_p$ for $X \sqsubseteq Y$ in $\mathcal{T}$ (line 6). Afterwards the main function retrieves more justifications for $A \sqsubseteq B$ from $\mathcal{J}_p$ (line 8) and tries to find a new justification (line 4). Algorithm 1 eventually computes a set of justification patterns for $X \sqsubseteq Y$ that generalizes all justifications for all ACI entailments, as shown below.

Theorem 1 *ComputeJustificationPatterns($\mathcal{T}$) returns a set $\mathcal{P}$ of justification patterns for $X \sqsubseteq Y$ in $\mathcal{T}$ such that for* *all ACIs $A \sqsubseteq B$ entailed by $\mathcal{T}$ and all justifications $\mathcal{J}$ for $A \sqsubseteq B$ in $\mathcal{T}$, there exists a justification pattern $\mathcal{J}_p \in \mathcal{P}$ and a differentiated substitution θ for $\mathcal{J}_p$ such that $\mathcal{J}_p\theta = \mathcal{J}$.*

After a set $\mathcal{P}$ of justification patterns for $X \sqsubseteq Y$ in $\mathcal{T}$ is obtained, we are able to compute all JP-based explanations for $A \sqsubseteq B$ in $\mathcal{T}$ w.r.t. $\mathcal{P}$ by bipartition of every justification pattern in $\mathcal{P}$ and by treating one part of the pattern as a subset of $\mathcal{T}$ and the other part as a JP-based explanation. Given a justification pattern $\mathcal{J}_p$, we define a *bipartition* of $\mathcal{J}_p$ as a pair $(\mathcal{J}_1, \mathcal{J}_2)$ such that $\mathcal{J}_1 \cap \mathcal{J}_2 = \emptyset$ and $\mathcal{J}_1 \cup \mathcal{J}_2 = \mathcal{J}_p$. By $\mathsf{bipart}(\mathcal{J}_p)$ we denote the set of bipartitions of $\mathcal{J}_p$. For a set S of variational axioms, by $\mathsf{inst}(S)$ we denote the set of axioms obtained from S by replacing all different concept (resp. role) variables with different fresh concept (resp. role) names. The following theorem provides a method for computing all different JP-based explanations.

Theorem 2 *Given a coherent TBox $\mathcal{T}$, a set $\mathcal{P}$ of justification patterns for $X \sqsubseteq Y$ in $\mathcal{T}$, and an observation $A \sqsubseteq B$ such that $\mathcal{T} \not\models A \sqsubseteq B$ and $\mathcal{T} \cup \{A \sqsubseteq B\}$ is coherent, the set of different JP-based explanations for $A \sqsubseteq B$ in $\mathcal{T}$ w.r.t. $\mathcal{P}$ amounts to $\mathcal{S} = \{\mathsf{inst}(\mathcal{J}_1\theta) \mid \mathcal{J}_p \in \mathcal{P}, (\mathcal{J}_1, \mathcal{J}_2) \in \mathsf{bipart}(\mathcal{J}_p)$ and θ is a differentiated substitution for $\mathcal{J}_2 \cup \{X \sqsubseteq Y\}$ such that $X\theta = A$, $Y\theta = B$, $\mathcal{J}_2\theta \subseteq \mathcal{T}$, $\mathsf{inst}(\mathcal{J}_1\theta) \not\models A \sqsubseteq B$, $\mathcal{T} \cup \mathsf{inst}(\mathcal{J}_1\theta)$ is coherent, $\mathcal{J}_2\theta \cup \mathsf{inst}(\mathcal{J}_1\theta) \in \mathsf{Jst}(A \sqsubseteq B, \mathcal{T} \cup \mathsf{inst}(\mathcal{J}_1\theta))$, and no explanation $\mathcal{E}'$ for $A \sqsubseteq B$ in $\mathcal{T}$ fulfills $\mathcal{E}' \subseteq_{\mathsf{ds}} \mathsf{inst}(\mathcal{J}_1\theta)$ and $\mathsf{inst}(\mathcal{J}_1\theta) \not\subseteq_{\mathsf{ds}} \mathcal{E}'\}$ up to renaming fresh names.*

For efficiency, the above method can be optimized by simplifying the tests for non-triviality, $\subseteq_{\mathsf{ds}}$-minimality and admissibility, as shown in the following corollary.

Corollary 1 *The set of different JP-based explanations for $A \sqsubseteq B$ in $\mathcal{T}$ w.r.t. $\mathcal{P}$ amounts to $\{\mathsf{inst}(\mathcal{J}_1\theta) \mid \mathcal{J}_p \in \mathcal{P}, (\mathcal{J}_1, \mathcal{J}_2) \in \mathsf{bipart}(\mathcal{J}_p)$ and θ is a differentiated substitution for $\mathcal{J}_2 \cup \{X \sqsubseteq Y\}$ such that $\emptyset \subset \mathcal{J}_2 \subset \mathcal{J}_p$, $X\theta = A$, $Y\theta = B$, $\mathcal{J}_2\theta \subseteq \mathcal{T}$, $\mathcal{T} \cup \mathsf{inst}(\mathcal{J}_1\theta)$ is coherent, $\mathcal{J} \cup \mathsf{inst}(\mathcal{J}_1\theta) \not\models A \sqsubseteq B$ for all $\mathcal{J} \in \mathbf{subs_1}(\mathcal{J}_2\theta)$, and $\mathcal{T} \cup \mathcal{E}' \not\models A \sqsubseteq B$ for all $\mathcal{E}' \in \mathbf{lifts_1}(\mathsf{inst}(\mathcal{J}_1\theta)) \cup \mathbf{subs_1}(\mathsf{inst}(\mathcal{J}_1\theta))\}$ up to renaming fresh names.*

Experimental Evaluation

We implemented both the method for computing all justification patterns and the method for computing all JP-based explanations in Java, using the Pellet (Sirin et al. 2007) API (version 2.3.1) to discover and check entailments, and using MySQL to manage justification patterns. The implementation of the former method computes justifications in the syntactic locality-based module (Grau et al. 2007) for the left-hand side of the given entailment rather than in the whole TBox. This optimization has been proved to be sound and complete (Suntisrivaraporn et al. 2008). The implementation of the latter method computes JP-based explanations for an observation from lower to higher levels, where at level k it computes all JP-based explanations whose cardinalities are k. This way enables us to stop computing JP-based explanations at a certain level when time resource is limited.

We collected ten coherent TBoxes. Some of their statistics are reported in Table 1. The first TBox is the well-known LUBM (Guo, Pan, and Heflin 2005) TBox. The next two are the UOBM (Ma et al. 2006) TBoxes. They extend LUBM by adding OWL Lite features and OWL DL features, respectively. The next four were collected from http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library. They describe family relationships, the philosophical domain, wines and an early prototype GALEN model, respectively. The last three TBoxes are large TBoxes about medical extensions and were collected from http://www.opengalen.org/download/opengalen8-owl-sources.zip.

Table 1: The characteristics of every test TBox

TBox	DL Expressivity	#C	#R	#A	#E
LUBM	$\mathcal{ALEHI}_+(\mathbf{D})$	43	32	93	76
UOBM-Lite	$\mathcal{ALEHIN}_+(\mathbf{D})$	52	43	145	86
UOBM-DL	$\mathcal{SHOIN}(\mathbf{D})$	69	44	206	113
generations	$\mathcal{ALCOIF}$	18	4	38	45
wine	$\mathcal{SHOIN}(\mathbf{D})$	77	14	657	322
philosurfical	$\mathcal{AL}$	377	313	1465	2981
not-galen	$\mathcal{ALEHF}_+$	3097	413	5771	32475
physiology	$\mathcal{ALEI}$	2129	134	2256	2146
drugs	$\mathcal{ALEHIF}(\mathbf{D})$	4258	89	5042	17103
pathology	$\mathcal{ALEI}(\mathbf{D})$	7378	258	7159	14235

Note: #C is the number of concept names; #R is the number of role names; #A is the number of axioms in the TBox; #E is the number of ACIs that are entailed by the TBox; The DL expressivity was detected by Pellet (version 2.3.1).

To show whether the proposed methods can constitute a practical approach to TBox abduction, we carried out two experiments. One computes up to the initial 1000 justification patterns in every test TBox. These justification patterns are ranked in descending order of their supports, where the *support* of a justification pattern $\mathcal{J}_p$ in a TBox $\mathcal{T}$ is defined as the number of justifications that can be instantiated from $\mathcal{J}_p$ in $\mathcal{T}$. The other experiment computes JP-based explanations whose cardinalities are respectively 1 and 2 for 50 randomly generated observations from the top 100 justification patterns computed in the former experiment. Each generated observation is an ACI that is not entailed by the given TBox while adding it to the TBox does not introduce incoherence. The top justification patterns have the largest supports in a given TBox, thus the JP-based explanations derived from them are more likely to mimic the way the TBox is modeled than other explanations. We set a time limit of 10000 seconds for computing JP-based explanations whose cardinalities are k (where $k \in \{1, 2\}$) for all 50 generated observations in a test TBox. When timeout incurs we can also see how many top justification patterns are completely handled for all 50 generated observations. All experiments were conducted on a laptop with Intel Dual-Core 2.60GHz CPU and 8GB RAM, running Windows 7 (64 bit) with the maximum Java heap size set to 8GB. The implemented system and test TBoxes are available at http://www.dataminingcenter.net/tboxabd/.

The "JP-Comp" part in Table 2 shows the statistics about every test TBox in computing up to the initial 1000 justification patterns. If the number of justification patterns is less than 1000, the actual number is shown. The reported execution time includes the time for loading the TBox and com-

Table 2: The running statistics for every test TBox

TBox	JP-Comp		JPBE1-Comp			JPBE2-Comp		
	#P	T(s)	#TP	T1(s)	#E	#TP	T2(s)	#E
LUBM	17	*1.0	17	3.9	253	17	178.8	18898
UOBM-Lite	23	*1.3	23	5.5	342	23	250.5	39870
UOBM-DL	44	*2.3	44	6.1	255	44	290.7	40659
generations	31	*0.9	31	2.7	182	–	–	–
wine	894	*86.6	100	66.7	0	100	188.9	10948
philosurfical	12	*7.6	12	89.7	1834	7	7111.4	724961
not-galen	1000	1340.9	100	72.0	1384	3	5637.8	144626
physiology	274	*14.8	100	108.7	477	98	9996.8	398521
drugs	289	*69.6	100	193.3	617	4	2749.2	49650
pathology	1000	122.8	100	112.6	935	7	4161.8	29759

Note: #P is the number of computed justification patterns; T is the execution time in seconds for computing up to the initial 1000 justification patterns (leading with * if all justification patterns are computed); #TP is the number of top justification patterns that are completely handled (for all 50 generated observations) in 10000 seconds; T1 (T2) is the execution time in seconds for completely handling the top #TP justification patterns; #E is the total number of JP-based explanations computed for all 50 generated observations after completely handling the top #TP justification patterns.

puting syntactic locality-based modules. The results show that justification patterns can efficiently be computed in expressive DLs. Except for `not-galen` (resp. `pathology`) that costs 1340.9 (resp. 122.8) seconds to compute the initial 1000 justification patterns, other test TBoxes only cost less than two minutes to compute all justification patterns.

The "JPBEk-Comp" part of Table 2 shows the statistics about every test TBox in computing all JP-based explanations whose cardinalities are k for top 100 justification patterns and all 50 generated observations. When $k = 1$, up to the top 100 justification patterns are completely handled in at most a few minutes. But the setting of $k = 1$ cannot guarantee that at least one JP-based explanation is generated (see 0 for `wine`), hence it may need to compute JP-based explanations with cardinality 2. The case where $k = 2$ is inapplicable to `generations` because all JP-based explanations in `generations` have cardinality 1. For each of other TBoxes, when $k = 2$, there are at least several top justification patterns that can be completed handled in 10000 seconds with a number of JP-based explanations generated. These results show that a practical use of the proposed methods is to compute JP-based explanations with cardinality 1 or from several top justification patterns. The results also indicate that we may need to postprocess the computed JP-based explanations by choosing several best ones from them. This can be done, either by ranking the computed JP-based explanations, or by exploiting an interactive debugging approach such as (Shchekotykhin and Friedrich 2010).

Related Work

For abductive reasoning in DLs, there are three different general problems advocated in (Elsenbroich, Kutz, and Sattler 2006). They are concept abduction (Colucci et al. 2004; Bienvenu 2008; Noia, Sciascio, and Donini 2009), ABox abduction (Klarman, Endriss, and Schlobach 2011; Du et al. 2011; Calvanese et al. 2013; Du, Wang, and Shen 2014) as well as TBox abduction targeted in this work. There are several slightly different problem settings for TBox abduction. All of them typically target computing explanations $\mathcal{E}$ for an observation α in a TBox $\mathcal{T}$ such that $\mathcal{T} \cup \mathcal{E} \models \alpha$. In (Elsenbroich, Kutz, and Sattler 2006) $\mathcal{T}$ and $\mathcal{E}$ are allowed to be expressed in different DLs, but no method is provided with this general setting. In (Hubauer, Lamparter, and Pirker 2010) $\mathcal{E}$ is defined as a set of concept inclusions having specified patterns. An automata-based method for this setting is proposed but no evaluation is provided. In (Lambrix, Dragisic, and Ivanova 2012; Lambrix and Liu 2013; Wei-Kleiner, Dragisic, and Lambrix 2014) $\mathcal{E}$ is restricted to be a set of ACIs. Some methods are provided with this setting for concept taxonomy (Lambrix and Liu 2013), acyclic $\mathcal{ALC}$ TBox (Lambrix, Dragisic, and Ivanova 2012) and $\mathcal{EL}^{++}$ TBox (Wei-Kleiner, Dragisic, and Lambrix 2014), respectively. In (Halland, Britz, and Klarman 2014) $\mathcal{E}$ is defined as a set of concept inclusions allowing only concept name, atomic negation and unqualified existential or universal restriction. A tableau-based method for this setting is proposed but no evaluation is provided. In (Koopmann and Schmidt 2015) $\mathcal{E}$ is restricted to be of the form $\{A \sqsubseteq \forall r.(B_1 \sqcup \ldots \sqcup B_n)\}$, computed by the uniform interpolant of the negated observation $\neg\alpha$ and a given acyclic $\mathcal{ALCH}$ TBox. Compared to (Hubauer, Lamparter, and Pirker 2010), our problem setting restricts the pattern of an explanation as a whole rather than patterns of individual axioms, thus making the computation of explanations more practical. Compared to other existing problem settings, our setting allows arbitrary DL constructors to be used in explanations. More importantly, we have provided evaluation results to show that our setting enables a practical approach to TBox abduction that works well with complex TBoxes.

Conclusion and Future Work

For practical TBox abduction in complex TBoxes, we made the following contributions in this paper. First of all, we proposed a novel class of preferred explanations in TBox abduction, namely JP-based explanations. The restriction of being derivable from justification patterns guarantees a finite space for explanations. The $\subseteq_{\mathsf{ds}}$-minimality further reduces the space for preferred explanations. Secondly, we proposed an efficient method for computing all justification patterns from a TBox expressible in any DL. Thirdly, we proposed an efficient method for computing all JP-based explanations from a given set of justification patterns. Finally, we empirically showed that the proposed methods are efficient for complex TBoxes.

For future work, we plan to explore real-life applications of TBox abduction based on justification patterns besides repairing missing is-a relations. One potential application is recognizing textual entailment, in which some promising results achieved by applying abduction have already been shown (Raina, Ng, and Manning 2005). We also plan to study practical criteria for ranking JP-based explanations in TBox abduction and to investigate methods for computing high-ranked JP-based explanations as early as possible.

Acknowledgements

This work was partly supported by National Natural Science Foundation of China under grants 61375056 and 61573386, Natural Science Foundation of Guangdong Province under grant 2016A030313292, Guangdong Province Science and Technology Plan projects under grant 2016B030305007, and Sun Yat-sen University Cultivation Project (16lgpy40).

References

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

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

Calvanese, D.; Ortiz, M.; Simkus, M.; and Stefanoni, G. 2013. Reasoning about explanations for negative query answers in DL-Lite. *Journal of Artificial Intelligence Research* 48:635–669.

Colucci, S.; Noia, T. D.; Sciascio, E. D.; Donini, F. M.; and Mongiello, M. 2004. A uniform tableaux-based approach to concept abduction and contraction in $\mathcal{ALN}$. In *Proceedings of the 17th International Workshop on Description Logics*.

Du, J.; Qi, G.; Shen, Y.; and Pan, J. Z. 2011. Towards practical ABox abduction in large OWL DL ontologies. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI)*, 1160–1165.

Du, J.; Qi, G.; and Fu, X. 2014. A practical fine-grained approach to resolving incoherent OWL 2 DL terminologies. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM)*, 919–928.

Du, J.; Wang, K.; and Shen, Y. 2014. A tractable approach to ABox abduction over description logic ontologies. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, 1034–1040.

Elsenbroich, C.; Kutz, O.; and Sattler, U. 2006. A case for abductive reasoning over ontologies. In *Proceedings of the 3rd OWLED Workshop on OWL: Experiences and Directions*.

Grau, B. C.; Horrocks, I.; Kazakov, Y.; and Sattler, U. 2007. Just the right amount: extracting modules from ontologies. In *Proceedings of the 16th International Conference on World Wide Web (WWW)*, 717–726.

Guo, Y.; Pan, Z.; and Heflin, J. 2005. LUBM: A benchmark for OWL knowledge base systems. *Journal of Web Semantics* 3(2-3):158–182.

Halland, K.; Britz, A.; and Klarman, S. 2014. TBox abduction in ALC using a DL tableau. In *Proceedings of the 27th International Workshop on Description Logics*, 556–566.

Horrocks, I.; Kutz, O.; and Sattler, U. 2006. The even more irresistible $\mathcal{SROIQ}$. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 57–67.

Horrocks, I.; Patel-Schneider, P. F.; and van Harmelen, F. 2003. From $\mathcal{SHIQ}$ and RDF to OWL: the making of a web ontology language. *Journal of Web Semantics* 1(1):7–26.

Hubauer, T.; Lamparter, S.; and Pirker, M. 2010. Automata-based abduction for tractable diagnosis. In *Proceedings of the 23rd International Workshop on Description Logics*.

Junker, U. 2004. QUICKXPLAIN: Preferred explanations and relaxations for over-constrained problems. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI)*, 167–172.

Kalyanpur, A.; Parsia, B.; Horridge, M.; and Sirin, E. 2007. Finding all justifications of OWL DL entailments. In *Proceedings of the 6th International Semantic Web Conference (ISWC)*, 267–280.

Klarman, S.; Endriss, U.; and Schlobach, S. 2011. ABox abduction in the description logic $\mathcal{ALC}$. *Journal of Automated Reasoning* 46(1):43–80.

Koopmann, P., and Schmidt, R. A. 2015. LETHE: saturation-based reasoning for non-standard reasoning tasks. In *Informal Proceedings of the 4th International Workshop on OWL Reasoner Evaluation (ORE)*, 23–30.

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

Ma, L.; Yang, Y.; Qiu, Z.; Xie, G.; Pan, Y.; and Liu, S. 2006. Towards a complete OWL ontology benchmark. In Sure, Y., and Domingue, J., eds., *Proceedings of the 3rd European Semantic Web Conference (ESWC)*, 125–139.

Noia, T. D.; Sciascio, E. D.; and Donini, F. M. 2009. A tableaux-based calculus for abduction in expressive description logics: Preliminary results. In *Proceedings of the 22nd International Workshop on Description Logics*.

Raina, R.; Ng, A. Y.; and Manning, C. D. 2005. Robust textual inference via learning and abductive reasoning. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI)*, 1099–1105.

Shchekotykhin, K. M., and Friedrich, G. 2010. Query strategy for sequential ontology debugging. In *Proceedings of the 9th International Semantic Web Conference (ISWC)*, 696–712.

Sirin, E.; Parsia, B.; Grau, B. C.; Kalyanpur, A.; and Katz, Y. 2007. Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics* 5(2):51–53.

Suntisrivaraporn, B.; Qi, G.; Ji, Q.; and Haase, P. 2008. A modularization-based approach to finding all justifications for OWL DL entailments. In *Proceedings of the 3rd Asian Semantic Web Conference (ASWC)*, 1–15.

Wei-Kleiner, F.; Dragisic, Z.; and Lambrix, P. 2014. Abduction framework for repairing incomplete EL ontologies: Complexity results and algorithms. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, 1120–1127.