Algorithm for Adapting Cases Represented in a Tractable Description Logic

Liang Chang[1,2], Uli Sattler[2], and Tianlong Gu[1]

[1] Guangxi Key Laboratory of Trusted Software,
Guilin University of Electronic Technology, Guilin, 541004, China
[2] School of Computer Science,
The University of Manchester, Manchester, M13 9PL, UK
changl.guet@gmail.com,sattler@cs.manchester.ac.uk,cctlgu@guet.edu.cn

Abstract. Case-based reasoning (CBR) based on description logics (DLs) has gained a lot of attention lately. Adaptation is a basic task in CBR that can be modeled as a knowledge base revision problem which has been solved in propositional logic. However, in DLs, adaptation is still a challenge problem since existing revision operators only work well for DLs of the *DL-Lite* family. It is difficult to design revision algorithms that are syntax-independent and fine-grained. In this paper, we propose a new method for adaptation based on the tractable DL $\mathcal{EL}_\bot$. Following the idea of adaptation as revision, we firstly extend the logical basis for describing cases from propositional logic to DL and present a formalism for adaptation based on $\mathcal{EL}_\bot$. Then we show that existing revision operators and algorithms in DLs can not be used for this formalism. Finally we present our adaptation algorithm. Our algorithm is syntax-independent and fine-grained, and satisfies the requirements on revision operators.

Keywords: description logic, case-based reasoning, adaptation, knowledge base revision, $\mathcal{EL}_\bot$.

1 Introduction

Description logic (DL) is a family of logics for representing and reasoning about knowledge of static application domains [2]. It is playing a central role in the Semantic Web, serving as the basis of the W3C-recommended Web ontology language OWL [11]. The main strength of DLs is that they offer considerable expressive power often going far beyond propositional logic, while reasoning is still decidable. Furthermore, DLs have well-defined semantics and are supported by many efficient reasoners.

In the last few years, there has been a growing interest in bringing the power and character of DLs into case-based reasoning (CBR) [9,10,18]. CBR is a type of analogical reasoning in which a new problem is solved by reusing past experiences called *source cases*. There are two basic tasks in the CBR inference: retrieval and adaptation. *Retrieval* aims at selecting a source case that is similar to the new problem according to some similarity criterion. *Adaptation* aims at generating a solution for the new problem by adapting the solution contained in the source case. At present, most research is concerned with the retrieval task when introducing DLs into CBR [9,18].

In comparison to retrieval, adaptation is often considered to be the more difficult task. One approach for this task is to model the adaptation process as the *revision* problem of a knowledge base (KB) [7,17]; it is hoped that an adaptation algorithm satisfies the AGM postulates on revision operators [1,13]. In propositional logic, there are many revision operators which satisfy the AGM postulates and can be applied to complete the adaptation task [17]. However, in DLs, it is difficult to design revision operators and algorithms that satisfy the AGM postulates [4]. Especially, it is a great challenge to design revision algorithms that are independent of the syntactical forms of KBs and are fine-grained for the minimal change principle.

According to the semantics adopted for defining "minimal change", existing revision operators and algorithms for DLs can be divided into two groups: *model-based approaches* (MBAs) [14] and *formula-based approaches* (FBAs) [4,16,20]. In MBAs, the semantics of minimal change is defined by measuring the distance between models. MBAs are syntax-independent and fine-grained, but at present they only work for DLs of the *DL-Lite* family [14]. In FBAs, the semantics of minimal change is reflected in the minimality of formulas removed by the revision process. There are two FBAs in the literature. One is based on the deductive closure of a KB [4,16]; it is syntax-independent and fine-grained, but again only works for *DL-Lite*. Another is based on justifications [20]; although it is applicable to DLs such as $\mathcal{SHOIN}$, it is syntax-dependent and not fine-grained.

DLs of the $\mathcal{EL}$ family are popular for building large-scale ontologies [3]. Some important medical ontologies and life science ontologies are built in $\mathcal{EL}$, such as the SNOMED CT [19] and the Gene Ontology [8]. A feature of this family of DLs is that they allow for reasoning in polynomial time, while being able to describe "relational structures". They are promising DLs for CBR since they are, on the one hand, of interesting expressive power and, on the other hand, restricted enough so that we can hope for a practical adaptation approach.

In the literature, some good results on combining DLs of the $\mathcal{EL}$ family with the retrieval task of CBR have been presented [18]; many algorithms for measuring the similarity of concepts in these DLs have also been proposed [15]. However, adaptation based on these DLs is still an open problem. One reason is that existing revision operators applicable to these DLs, to the best of our knowledge, are syntax-dependent and not fine-grained.

In this paper we present a new method for adaptation in the DL $\mathcal{EL}_\bot$ of the $\mathcal{EL}$ family. Our contributions regard three aspects. Firstly, we extend the logical basis for describing cases from propositional logic to the DL $\mathcal{EL}_\bot$, with a powerful way of describing cases as ABoxes in DL. Secondly, we extend the "adaptation as KB revision" view from [17] to the above setting and get a formalism for adaptation based on $\mathcal{EL}_\bot$. Finally, for the adaptation setting we provide an adaptation algorithm which is syntax-independent and fine-grained. The proofs of all of our technical results are given in the accompanying technical report [5].

2 The Description Logic $\mathcal{EL}_\bot$

The DL $\mathcal{EL}_\bot$ extends $\mathcal{EL}$ with bottom concept (and consequently disjointness statements) [3]. Let N_C, N_R and N_I be disjoint sets of *concept names*, *role names* and *individual names*, respectively. $\mathcal{EL}_\bot$*-concepts* are built according to the following syntax rule $C ::= \top \mid \bot \mid A \mid C \sqcap D \mid \exists r.C$, where $A \in N_C$, $r \in N_R$, and C, D range over $\mathcal{EL}_\bot$-concepts.

A *TBox* $\mathcal{T}$ is a finite set of *general concept inclusions* (GCIs) of the form $C \sqsubseteq D$, where C and D are concepts. An *ABox* $\mathcal{A}$ is a finite set of *concept assertions* of the form $C(a)$ and *role assertions* of the form $r(a, b)$, where $a, b \in N_I$, $r \in N_R$, and C is a concept. A *knowledge base* (KB) is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$.

Example 1. Consider the example on breast cancer treatment discussed in [17]. We add some background knowledge to it and describe the knowledge by the following GCIs in a TBox $\mathcal{T}$:

$$Tamoxifen \sqsubseteq \mathit{Anti\text{-}oestrogen}, \qquad \mathit{Anti\text{-}aromatases} \sqsubseteq \mathit{Anti\text{-}oestrogen},$$
$$Tamoxifen \sqsubseteq \exists \mathit{metabolizedTo}.(\mathit{Compounds} \sqcap \exists \mathit{bindto}.\mathit{OestrogenReceptor}),$$
$$(\exists \mathit{hasGene}.CYP2D6) \sqcap (\exists \mathit{TreatBy}.Tamoxifen) \sqsubseteq \bot.$$

These GCIs state that both tamoxifen and anti-aromatases are anti-oestrogen; tamoxifen can be metabolized into compounds which will bind to the oestrogen receptor; and tamoxifen is contraindicated for people with some gene of CYP2D6.

Suppose *Mary* is a patient with a gene of the class CYP2D6 and with some symptom captured by a concept *Symp*. Then we can describe these information by an ABox $\mathcal{N}_{pb} = \{\mathit{Symp}(\mathit{Mary}), \exists \mathit{hasGene}.CYP2D6(\mathit{Mary})\}$. □

The semantics of $\mathcal{EL}_\bot$ is defined by an *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where the *interpretation domain* $\Delta^\mathcal{I}$ is a non-empty set composed of individuals, and $\cdot^\mathcal{I}$ is a function which maps each concept name $A \in N_C$ to a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, maps each role name $r \in N_R$ to a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and maps each individual name $a \in N_I$ to an individual $a^\mathcal{I} \in \Delta^\mathcal{I}$. The function $\cdot^\mathcal{I}$ is inductively extended to arbitrary concepts as follows: $\top^\mathcal{I} := \Delta^\mathcal{I}$, $\bot^\mathcal{I} := \emptyset$, $(C \sqcap D)^\mathcal{I} := C^\mathcal{I} \cap D^\mathcal{I}$, and $(\exists r.C)^\mathcal{I} := \{x \in \Delta^\mathcal{I} \mid \text{there exists } y \in \Delta^\mathcal{I} \text{ such that } (x, y) \in r^\mathcal{I} \text{ and } y \in C^\mathcal{I}\}$.

The *satisfaction relation* "$\models$" between any interpretation $\mathcal{I}$ and any GCI $C \sqsubseteq D$, concept assertion $C(a)$, role assertion $r(a, b)$, TBox $\mathcal{T}$ or ABox $\mathcal{A}$ is defined inductively as follows: $\mathcal{I} \models C \sqsubseteq D$ iff $C^\mathcal{I} \subseteq D^\mathcal{I}$; $\mathcal{I} \models C(a)$ iff $a^\mathcal{I} \in C^\mathcal{I}$; $\mathcal{I} \models r(a, b)$ iff $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$; $\mathcal{I} \models \mathcal{T}$ iff $\mathcal{I} \models X$ for every $X \in \mathcal{T}$; and $\mathcal{I} \models \mathcal{A}$ iff $\mathcal{I} \models X$ for every $X \in \mathcal{A}$.

$\mathcal{I}$ is a *model* of a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ if $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$. We use $\mathit{mod}(\mathcal{K})$ to denote the set of models of a KB $\mathcal{K}$. Two KBs $\mathcal{K}_1$ and $\mathcal{K}_2$ are *equivalent* (written $\mathcal{K}_1 \equiv \mathcal{K}_2$) iff $\mathit{mod}(\mathcal{K}_1) = \mathit{mod}(\mathcal{K}_2)$.

There are many inference problems on DLs. Here we only introduce *consistency* and *entailment*. A KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is *consistent* (or $\mathcal{A}$ is *consistent* w.r.t. $\mathcal{T}$) if $mod(\mathcal{K}) \neq \emptyset$. $\mathcal{K}$ *entails* a GCI, assertion or ABox X (written $\mathcal{K} \models X$) if $\mathcal{I} \models X$ for every $\mathcal{I} \in mod(\mathcal{K})$. For these two inference problems the following results hold: a KB $\mathcal{K}$ is inconsistent iff $\mathcal{K} \models \top \sqsubseteq \bot$ iff $\mathcal{K} \models \bot(a)$ for some individual name a occurring in $\mathcal{K}$. If $\mathcal{K}$ is inconsistent, then we say that $\mathcal{K}$ *entails a clash.*

Example 2. Consider the TBox $\mathcal{T}$ and ABox $\mathcal{N}_{pb}$ presented in Example 1. Suppose there is an ABox $\mathcal{A}_{sol} = \{TreatBy(Mary, y), Tamoxifen(y)\}$ and a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}_{sol} \cup \mathcal{N}_{pb} \rangle$. It is obvious that $\mathcal{K}$ is inconsistent and entails a clash. More precisely, we have that $\mathcal{K} \models \top \sqsubseteq \bot$ and $\mathcal{K} \models \bot(Mary)$. □

For any TBox, ABox or KB X, we use N_C^X (resp., N_R^X, N_I^X) to denote the set of concept names (resp., role names, individual names) occurring in X, and define the *signature* of X as $sig(X) = N_C^X \cup N_R^X \cup N_I^X$.

For any concept C, the *role depth* $rd(C)$ is the maximal nesting depth of "$\exists$" in C. For any TBox or ABox X, let $sub(X) = \bigcup_{C \sqsubseteq D \in X} \{C, D\}$ if X is a TBox, and $sub(X) = \{C \mid C(a) \in X\}$ if X is an ABox, then the *depth* of X is defined as $depth(X) = \max\{rd(C) \mid C \in sub(X)\}$.

3 Formalization of Adaptation Based on $\mathcal{EL}_\bot$

In this section we present a formalism for adaptation based on $\mathcal{EL}_\bot$. There are many different approaches for the formalization of adaptation in CBR. Here we follow the approach presented in [17] to formulate adaptation as knowledge base revision, with the difference that our formalism is based on the DL $\mathcal{EL}_\bot$ instead of propositional logic.

The basic idea of CBR is to solve similar problems with similar solutions. The new problem that needs to be solved is called *target problem.* The problems which have been solved and stored are called *source problems.* Each source problem *pb* has a *solution sol*, and the pair (pb, sol) is called a *source case.* A finite set of source cases forms a *case base.* Given a target problem *tgt*, the retrieval step of CBR will pick out a source case (pb, sol) according to the similarity between target problem and source problems, then the adaptation step will generate a solution sol_{tgt} for *tgt* by adapting *sol*.

In [17], the adaptation process is modeled as KB revision in propositional logic. More precisely, let two formulas $kb_1 = pb \wedge sol$ and $kb_2 = tgt$, then the solution sol_{tgt} is generated by calculating $(dk \wedge kb_1) \circ (dk \wedge kb_2)$, where dk is a formula describing the domain knowledge, and $\circ$ is a revision operator that satisfies the AGM postulates in propositional logic.

In our paper, after introducing the DL $\mathcal{EL}_\bot$ into CBR, knowledge in a CBR system is composed of three parts:

- the *domain or background knowledge* which is represented as a TBox $\mathcal{T}$;
- the knowledge about *case base* in which each *source case* is described by an ABox $\mathcal{A} = \mathcal{A}_{pb} \cup \mathcal{A}_{sol}$, where $\mathcal{A}_{pb}$ describes the *source problem* and $\mathcal{A}_{sol}$ describes the *solution*;

– the knowledge about *target problem* described by an ABox $\mathcal{N}_{pb}$.

With such a framework, given a target problem $\mathcal{N}_{pb}$, we can make use of similarity-measuring algorithms presented in the literature [18,15] to select a source case $\mathcal{A} = \mathcal{A}_{pb} \cup \mathcal{A}_{sol}$ such that, by treating individual names occurring in *srce* as variables, there exists a substitution σ such that $\sigma(\mathcal{A}_{pb})$ and $\mathcal{N}_{pb}$ has the maximum similarity. The retrieval algorithm will applies σ on $\mathcal{A}_{sol}$ and return $\sigma(\mathcal{A}_{sol})$ as a *possible solution* for the target problem.

Since retrieval algorithm is not the topic of this paper, we do not discuss it in detail here. In the rest of this paper, we will omit the notation σ and use $\mathcal{A}_{sol}$ directly to denote the possible solution returned by the retrieval algorithm.

Now suppose a possible solution has been returned by the retrieval algorithm, we define adaptation setting as follows.

Definition 1. *An* adaptation setting *based on $\mathcal{EL}_{\bot}$ is a triple $\mathcal{AS} = (\mathcal{T}, \mathcal{A}_{sol}, \mathcal{N}_{pb})$, where $\mathcal{T}$ is a TBox describing the domain knowledge of the CBR system, $\mathcal{N}_{pb}$ is an ABox describing the target problem, and $\mathcal{A}_{sol}$ is an ABox describing the possible solution returned by the retrieval algorithm.*

An ABox $\mathcal{A}'$ is a solution case *for an adaptation setting $\mathcal{AS} = (\mathcal{T}, \mathcal{A}_{sol}, \mathcal{N}_{pb})$ if $sig(\mathcal{A}') \subseteq sig(\mathcal{T}) \cup sig(\mathcal{A}_{sol}) \cup sig(\mathcal{N}_{pb})$ and the following statements hold:*

(R1) $\langle \mathcal{T}, \mathcal{A}' \rangle \models \mathcal{N}_{pb}$;
(R2) $\mathcal{A}' = \mathcal{A}_{sol} \cup \mathcal{N}_{pb}$ *if* $\mathcal{A}_{sol} \cup \mathcal{N}_{pb}$ *is consistent w.r.t.* $\mathcal{T}$*; and*
(R3) *if* $\mathcal{N}_{pb}$ *is consistent w.r.t.* $\mathcal{T}$ *then* $\mathcal{A}'$ *is also consistent w.r.t.* $\mathcal{T}$.

There may be more than one solution cases for an adaptation setting. From these solution cases, the user will select the best one and get a solution for the target problem.

The adaptation setting defined above is similar to the *instance-level revision* based on DLs [4]; **R1-R3** are just the basic requirements specified by the AGM postulates on revision operators [1,13]. More precisely, **R1** specifies that a revision result must entail the new information $\mathcal{N}_{pb}$; **R2** states that the revision operator should not change the KB $\langle \mathcal{T}, \mathcal{A}_{sol} \cup \mathcal{N}_{pb} \rangle$ if there is no conflict; **R3** states that the revision operator must preserve the consistency of KBs.

From the point of view of adaptation, these requirements on solutions are explained as follows [17]. If **R1** is violated, then it means that the adaptation process failed to solve the target problem. **R2** states that if the possible solution does not contradict the target problem w.r.t. the background knowledge, then it can be applied directly to the target problem. **R3** states that whenever the description of the target problem is consistent w.r.t. the domain knowledge, the adaptation process provides satisfiable result.

In the literature, there exist many revision operators and algorithms that can generate revision results satisfying the above requirements [4,14,16,20]. However, in practice, besides the necessary requirements specified by the definition, we hope that the adaptation algorithm satisfies two more requirements.

Firstly, the adaptation algorithm should be syntax-independent. Especially, if two target problems are logically equivalent w.r.t. the domain knowledge, then they should have the same solution. This requirement is formalized as follows:

(R4) Let $\mathcal{AS}_1 = (\mathcal{T}, \mathcal{A}_{sol_1}, \mathcal{N}_{pb_1})$ and $\mathcal{AS}_2 = (\mathcal{T}, \mathcal{A}_{sol_2}, \mathcal{N}_{pb_2})$ be two adaptation settings with $\langle \mathcal{T}, \mathcal{A}_{sol_1} \rangle \equiv \langle \mathcal{T}, \mathcal{A}_{sol_2} \rangle$ and $\langle \mathcal{T}, \mathcal{N}_{pb_1} \rangle \equiv \langle \mathcal{T}, \mathcal{N}_{pb_2} \rangle$. If $\mathcal{A}'_1$ is a solution case for $\mathcal{AS}_1$, then there must be a solution case $\mathcal{A}'_2$ for $\mathcal{AS}_2$ such that $\langle \mathcal{T}, \mathcal{A}'_1 \rangle \equiv \langle \mathcal{T}, \mathcal{A}'_2 \rangle$.

Secondly, the adaptation algorithm should guarantee a minimal change so that the experience contained in the solution of source cases is preserved as much as possible. Without such a requirement, given an adaptation setting $\mathcal{AS} = (\mathcal{T}, \mathcal{A}_{sol}, \mathcal{N}_{pb})$, if $\mathcal{A}_{sol} \cup \mathcal{N}_{pb}$ is inconsistent w.r.t. $\mathcal{T}$, then the ABox $\mathcal{N}_{pb}$ itself is a solution case by the definition. However, it is obvious that $\mathcal{N}_{pb}$ does not contain any information on solution, and all the experiences contained in the solution $\mathcal{A}_{sol}$ of the source case are completely lost.

We hope to specify the requirement on minimal change formally. However, it is non-trivial to do it in a framework based on DLs. Furthermore, it is well-accepted that there is no general notion of minimality that will "do the right thing" under all circumstances [4]. Therefore, under the framework of adaptation setting, we only specify this requirement informally as follows:

(R5) If $\mathcal{A}'$ is a solution case for the adaptation setting $\mathcal{AS} = (\mathcal{T}, \mathcal{A}_{sol}, \mathcal{N}_{pb})$, then the change from the KB $\langle \mathcal{T}, \mathcal{A}_{sol} \rangle$ to the KB $\langle \mathcal{T}, \mathcal{A}' \rangle$ is minimal.

To sum up, given an adaptation setting, we hope to generate solution cases which not only satisfy **R1-R3** specified by Definition 1, but also satisfy **R4** and "some reading" of **R5**.

Before the end of this section, we look an example of adaptation setting.

Example 3. Consider the TBox $\mathcal{T}$ and ABox $\mathcal{N}_{pb}$ presented in Example 1. Suppose $\mathcal{N}_{pb}$ is a description of the target problem. Suppose many successful treatment cases have been recorded in the case base, and from them a possible solution $\mathcal{A}_{sol} = \{TreatBy(Mary, y), Tamoxifen(y)\}$ is returned by the retrieval algorithm. Then we get an adaptation setting $\mathcal{AS} = (\mathcal{T}, \mathcal{A}_{sol}, \mathcal{N}_{pb})$. □

4 Existing Approaches to Instance-Level Revision

As we mentioned in Section 1, there are two groups of revision operators and algorithms for DLs in the literature. In this section, we show that they either do not support the DL $\mathcal{EL}_\perp$ or do not satisfy **R4** and **R5**.

4.1 Model-based Approaches

MBAs define revision operators over the distance between interpretations [14]. Under the framework of adaptation setting, suppose $\mathcal{A}'_i$ $(1 \leq i \leq n)$ are all the solution cases for $\mathcal{AS} = (\mathcal{T}, \mathcal{A}_{sol}, \mathcal{N}_{pb})$, and let $\mathcal{M} = \bigcup_{1 \leq i \leq n} mod(\langle \mathcal{T}, \mathcal{A}'_i \rangle)$, then, with MBAs, $\mathcal{M}$ should satisfy the following equation:

$$\mathcal{M} = \{\mathcal{J} \in mod(\langle \mathcal{T}, \mathcal{N}_{pb} \rangle) \mid there\ exists\ \mathcal{I} \in mod(\langle \mathcal{T}, \mathcal{A}_{sol} \rangle)\ with\ dist(\mathcal{I}, \mathcal{J}) = \min\{dist(\mathcal{I}', \mathcal{J}') \mid \mathcal{I}' \in mod(\langle \mathcal{T}, \mathcal{A}_{sol} \rangle), \mathcal{J}' \in mod(\langle \mathcal{T}, \mathcal{N}_{pb} \rangle)\}\}.$$

With MBAs, we can firstly calculate the set $\mathcal{M}$ by the above equation, and then construct the ABoxes $\mathcal{A}'_i$ $(1 \leq i \leq n)$ correspondingly.

In propositional logic, since each interpretation is only a truth assignment on propositional symbols, it is not difficult to measure the distance between interpretations and to calculate the set of models according to the distance [17]. However, it becomes very complex for DL, since basic symbols of DL are concept names, role names and individual names which are interpreted with set theory.

Let Σ be the set of concept names and role names occurring in $\mathcal{AS}$. There are four different approaches for measuring the distance $dist(\mathcal{I}, \mathcal{J})$:

– $dist^s_\sharp(\mathcal{I}, \mathcal{J}) = \sharp\{X \in \Sigma \mid X^\mathcal{I} \neq X^\mathcal{J}\}$,
– $dist^s_\subseteq(\mathcal{I}, \mathcal{J}) = \{X \in \Sigma \mid X^\mathcal{I} \neq X^\mathcal{J}\}$,
– $dist^a_\sharp(\mathcal{I}, \mathcal{J}) = \underset{X \in \Sigma}{sum}\ \sharp(X^\mathcal{I} \ominus X^\mathcal{J})$,
– $dist^a_\subseteq(\mathcal{I}, \mathcal{J}, X) = X^\mathcal{I} \ominus X^\mathcal{J}$ for every $X \in \Sigma$,

where $X^\mathcal{I} \ominus X^\mathcal{J} = (X^\mathcal{I} \setminus X^\mathcal{J}) \cup (X^\mathcal{J} \setminus X^\mathcal{I})$. Distances under $dist^s_\sharp$ and $dist^a_\sharp$ are natural numbers and are compared in the standard way. Distances under $dist^s_\subseteq$ are sets and are compared by set inclusion. Distances under $dist^a_\subseteq$ are compared as follows: $dist^a_\subseteq(\mathcal{I}_1, \mathcal{J}_1) \leq dist^a_\subseteq(\mathcal{I}_2, \mathcal{J}_2)$ iff $dist^a_\subseteq(\mathcal{I}_1, \mathcal{J}_1, X) \subseteq dist^a_\subseteq(\mathcal{I}_2, \mathcal{J}_2, X)$ for every $X \in \Sigma$. It is assumed that all models have the same interpretation domain and the same interpretation on individual names. In [14], the above four different semantics for MBAs are denoted as $\mathcal{G}^s_\sharp$, $\mathcal{G}^s_\subseteq$, $\mathcal{G}^a_\sharp$, and $\mathcal{G}^a_\subseteq$ respectively.

The limitation of applying these MBAs in our adaptation setting is shown by the following example.

Example 4. Consider an adaptation setting $\mathcal{AS}_1 = (\mathcal{T}_1, \mathcal{A}_1, \mathcal{N}_1)$, where

$$\mathcal{T}_1 = \{A \sqsubseteq \exists R.A,\ A \sqsubseteq C,\ E \sqcap \exists R.A \sqsubseteq \bot\},\ \ \mathcal{A}_1 = \{A(a)\},\ \ \mathcal{N}_1 = \{E(a)\}.$$

Firstly, by applying MBAs with the semantics $\mathcal{G}^s_\subseteq$ and $\mathcal{G}^s_\sharp$, we can calculate the set $\mathcal{M}$ and get the following result:

$$\mathcal{M} = \{\mathcal{J} \in mod(\langle \mathcal{T}_1, \mathcal{N}_1 \rangle) \mid \textit{there exists } \mathcal{I} \in mod(\langle \mathcal{T}_1, \mathcal{A}_1 \rangle) \textit{ such that } C^\mathcal{I} = C^\mathcal{J} \textit{ and } R^\mathcal{I} = R^\mathcal{J}\}.$$

We can show that there does not exist a finite number of ABoxes $\mathcal{A}'_i$ $(1 \leq i \leq n)$ such that $sig(\mathcal{A}'_i) \subseteq sig(\mathcal{T}_1) \cup sig(\mathcal{A}_1) \cup sig(\mathcal{N}_1)$ and $\bigcup_{1 \leq i \leq n} mod(\langle \mathcal{T}_1, \mathcal{A}'_i \rangle)$ $= \mathcal{M}$. Interested readers can refer to [6] for further details. Therefore, MBAs under the semantics $\mathcal{G}^s_\subseteq$ and $\mathcal{G}^s_\sharp$ suffer from inexpressibility.

Secondly, by applying MBAs with the semantics $\mathcal{G}^a_\subseteq$ and $\mathcal{G}^a_\sharp$, we will get

$$\mathcal{M} = \{\mathcal{J} \in mod(\langle \mathcal{T}_1, \mathcal{N}_1 \rangle) \mid \textit{there exists } \mathcal{I} \in mod(\langle \mathcal{T}_1, \mathcal{A}_1 \rangle) \textit{ such that } A^\mathcal{I} \ominus A^\mathcal{J} = E^\mathcal{I} \ominus E^\mathcal{J} = \{a^\mathcal{I}\}, C^\mathcal{I} = C^\mathcal{J} \textit{ and } R^\mathcal{I} = R^\mathcal{J}\}.$$

From $\mathcal{M}$ we can construct an ABox $\mathcal{A}'_1 = \{E(a), C(a), R(a, a)\}$ which satisfies $mod(\langle \mathcal{T}_1, \mathcal{A}'_1 \rangle) = \mathcal{M}$. Therefore, under the semantics $\mathcal{G}^a_\subseteq$ and $\mathcal{G}^a_\sharp$, $\mathcal{A}'_1$ is the only solution case. This result is very strange, since during the adaptation process there seems to be no "good" reason to enforce the assertion $R(a, a)$ to hold, and to exclude other possible assertions such as $\exists R.A(a)$. □

To sum up, there are four notions of computing models in existing MBAs. For the adaptation based on $\mathcal{EL}_\perp$, two notions suffer from inexpressibility and the other two notions only generate solution cases which are counterintuitive.

4.2 Formula-Based Approaches

In the literature there are two typical formula-based approaches for instance-level revision in DLs.

The first one is based on deductive closures [4,16]. With this approach, given an adaptation setting $\mathcal{AS} = (\mathcal{T}, \mathcal{A}, \mathcal{N})$, we will firstly calculate the deductive closure of $\mathcal{A}$ w.r.t. $\mathcal{T}$ (denoted $cl_\mathcal{T}(\mathcal{A})$), then find a maximal subset $\mathcal{A}_m$ of $cl_\mathcal{T}(\mathcal{A})$ that does not conflict with $\mathcal{N}$ and $\mathcal{T}$, and finally return $\mathcal{A}_m \cup \mathcal{N}$ as a solution case. This approach works for restricted forms of *DL-Lite*, by assuming that $\mathcal{A}$ only contains assertions of the form $A(a)$, $\exists R(a)$ and $R(a, b)$, with A and R concept names or role names, and therefore $cl_\mathcal{T}(\mathcal{A})$ is finite and can be calculated effectively. However, it does not work for $\mathcal{EL}_\perp$, since in our adaptation setting any $\mathcal{EL}_\perp$ concept can be used for describing assertions in the ABox $\mathcal{A}$.

The second FBA is based on justifications (also known as MinAs or kernel) [20]. With this approach, given an adaptation setting $\mathcal{AS} = (\mathcal{T}, \mathcal{A}, \mathcal{N})$, we will firstly construct a KB $\mathcal{K}_0 = \langle \mathcal{T}, \mathcal{A} \cup \mathcal{N} \rangle$ and find all the minimal subsets of $\mathcal{K}_0$ that entail a clash (i.e., all justifications for clashes); then we will compute a minimal set $\mathcal{R} \subseteq \mathcal{A}$ that contains at least one element from each justification (such a set is also called a *repair*); finally we will return $(\mathcal{A} \setminus \mathcal{R}) \cup \mathcal{N}$ as a solution case. This approach is applicable to DLs such as $\mathcal{SHOIN}$ and therefore can deal with $\mathcal{EL}_\perp$. However, as shown by the following examples, it is syntax-dependent and not fine-grained, and therefore does not satisfy **R4** and **R5**.

Example 5. Consider the adaptation setting $\mathcal{AS}_1 = (\mathcal{T}_1, \mathcal{A}_1, \mathcal{N}_1)$ described in the previous example. It is obvious that $\langle \mathcal{T}_1, \mathcal{A}_1 \cup \mathcal{N}_1 \rangle \models \perp(a)$ and for which there is only one justification $\mathcal{J} = \{A \sqsubseteq \exists R.A, E \sqcap \exists R.A \sqsubseteq \perp, A(a), E(a)\}$. Therefore the only repair is $\mathcal{R} = \{A(a)\}$ and the only solution case is $\mathcal{A}_1' = (\mathcal{A}_1 \setminus \mathcal{R}) \cup \mathcal{N}_1 = \mathcal{N}_1 = \{E(a)\}$.

This result is not good, since $\mathcal{A}_1'$ only contains the description of target problem and does not contain any information on solution. All the experiences provided by the solution $\mathcal{A}_1$ of source case are completely lost, therefore this approach is not fine-grained and does not satisfy **R5**. □

Example 6. Consider another adaptation setting $\mathcal{AS}_2 = (\mathcal{T}_2, \mathcal{A}_2, \mathcal{N}_2)$, where $\mathcal{T}_2 = \mathcal{T}_1$, $\mathcal{N}_2 = \mathcal{N}_1$ and $\mathcal{A}_2 = \{A(a), C(a), \exists R.C(a)\}$. Apply the FBA based on justifications again, we will get a solution case $\mathcal{A}_2' = \{E(a), C(a), \exists R.C(a)\}$.

It is obvious that $\langle \mathcal{T}_2, \mathcal{A}_2 \rangle \equiv \langle \mathcal{T}_1, \mathcal{A}_1 \rangle$ but $\langle \mathcal{T}_2, \mathcal{A}_2' \rangle \not\equiv \langle \mathcal{T}_1, \mathcal{A}_1' \rangle$. In other words, for the same target problem, we get two totally different solutions from two descriptions of experiences that are syntactically different but logically equivalent. The reason is that this approach does not satisfy **R4**. □

To sum up, for the adaptation based on $\mathcal{EL}_\perp$, existing FBAs either can not be applied directly, or can be applied but is syntax-dependent and not fine-grained.

5 Our Approach for Adaptation Based on $\mathcal{EL}_\bot$

In this section we present an algorithm for adaptation based on $\mathcal{EL}_\bot$. Given an adaptation setting $\mathcal{AS} = (\mathcal{T}, \mathcal{A}, \mathcal{N})$, our algorithm will firstly construct a non-redundant depth-bounded model for the KB $\langle \mathcal{T}, \mathcal{A} \rangle$; then a revision process based on justifications will be carried out on this model by treating a model as a set of assertions; finally the resulting model will be mapped back to an ABox which will be returned as a solution case.

Our algorithm is based on a structure named revision graph, which is close to the completion graph used in classical tableau decision algorithms of DLs [12]. We firstly introduce some notions and operations on this structure and then present the algorithm.

5.1 Notions and Operations on Revision Graph

First, we consider a set N_V of *variables*, and extend interpretations $\mathcal{I}$ of $\mathcal{EL}_\bot$ to interpret these variables just like individual names.

A *revision graph* for $\mathcal{EL}_\bot$ is a directed graph $\mathcal{G} = (V, E, \mathcal{L})$, where

- V is a finite set of nodes composed of individual names and variables;
- $E \subseteq V \times V$ is a set of edges satisfying:
  - there is no edge from variables to individual names, and
  - for each variable $y \in V$, there is at most one node x with $\langle x, y \rangle \in E$;
- each node $x \in V$ is labelled with a set of concepts $\mathcal{L}(x)$; and
- each edge $\langle x, y \rangle \in E$ is labelled with a set of role names $\mathcal{L}(\langle x, y \rangle)$; furthermore, if y is a variable then $\sharp\mathcal{L}(\langle x, y \rangle) = 1$.

For each edge $\langle x, y \rangle \in E$, we call y a *successor* of x and x a *predecessor* of y. *Descendant* is the transitive closure of successor.

For any node $x \in V$, we use $level(x)$ to denote the level of x in the graph, and define it inductively as follows: $level(x) = 0$ if x is an individual name, $level(x) = level(y) + 1$ if x is a variable with a predecessor y, and $level(x) = +\infty$ if x is a variable without predecessor.

A graph $\mathcal{B} = (V', E', \mathcal{L}')$ is a *branch* of $\mathcal{G}$ if $\mathcal{B}$ is a tree and a subgraph of $\mathcal{G}$.

A branch $\mathcal{B}_1 = (V_1, E_1, \mathcal{L}_1)$ is *subsumed* by another branch $\mathcal{B}_2 = (V_2, E_2, \mathcal{L}_2)$ if $\mathcal{B}_1$ and $\mathcal{B}_2$ have the same root node, $\sharp(V_1 \cap V_2) = 1$, and there is a function $f : V_1 \to V_2$ such that: $f(x) = x$ if x is the root node, $\mathcal{L}_1(x) \subseteq \mathcal{L}_2(f(x))$ for every node $x \in V_1$, $\langle f(x), f(y) \rangle \in E_2$ for every edge $\langle x, y \rangle \in E_1$, and $\mathcal{L}_1(\langle x, y \rangle) \subseteq \mathcal{L}_2(\langle f(x), f(y) \rangle)$ for every edge $\langle x, y \rangle \in E_1$.

A branch $\mathcal{B}$ is *redundant* in $\mathcal{G}$ if $\mathcal{B}$ is subsumed by another branch in $\mathcal{G}$, and every node in $\mathcal{B}$ except the root is a variable.

Revision graphs can be seen as ABoxes with variables. Given a revision graph $\mathcal{G} = (V, E, \mathcal{L})$, we call $\mathcal{A}_\mathcal{G} = \bigcup_{x \in V} \{C(x) \mid C \in \mathcal{L}(x)\} \cup \bigcup_{\langle x, y \rangle \in E} \{R(x, y) \mid R \in \mathcal{L}(\langle x, y \rangle)\}$ as the *ABox representation* of $\mathcal{G}$, and call $\mathcal{G}$ as the *revision-graph representation* of $\mathcal{A}_\mathcal{G}$.

Procedure B-MW($\mathcal{K}$, k)

```
  Input: a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and a non-negative integer $k$.
  Output: a revision graph $\mathcal{G} = (V, E, \mathcal{L})$.
1 Initialize the revision graph $\mathcal{G} = (V, E, \mathcal{L})$ as
    – $V := N_I^{\mathcal{K}}$,
    – $\mathcal{L}(a) := \{C \mid C(a) \in \mathcal{A}\}$ for each node $a \in V$,
    – $E := \{\langle a, b \rangle \mid$ there is some $R$ with $R(a, b) \in \mathcal{A}\}$,
    – $\mathcal{L}(\langle a, b \rangle) := \{R \mid R(a, b) \in \mathcal{A}\}$ for each edge $\langle a, b \rangle \in E$.
2 while there exists an expansion rule in Fig. 1 that is applicable to $\mathcal{G}$ do
    expand $\mathcal{G}$ by applying this rule.
3 for each node $x \in V$ do
    $\mathcal{L}(x) := \{C \mid C \in \mathcal{L}(x)$ and $C$ is a concept name $\}$.
4 while there exists a redundant branch $\mathcal{B} = (V_{\mathcal{B}}, E_{\mathcal{B}}, \mathcal{L}_{\mathcal{B}})$ in $\mathcal{G}$ do
    $E := E \setminus E_{\mathcal{B}}$;
    $V := V \setminus (V_{\mathcal{B}} \setminus \{x_{\mathcal{B}}\})$, where $x_{\mathcal{B}}$ is the root of $\mathcal{B}$.
5 Return $\mathcal{G} = (V, E, \mathcal{L})$.
```

Given a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and a non-negative integer k, we use the procedure B-MW($\mathcal{K}$, k) to construct a revision graph for them, and call this revision graph a *k-role-depth bounded minimal witness* for $\mathcal{K}$.

Example 7. Consider the adaptation setting $\mathcal{AS}_1$ described in Example 4 and call the procedure B-MW($\langle \mathcal{T}_1, \mathcal{A}_1 \rangle, 1$). During the execution of this procedure, two variables x_1 and x_2 will be introduced by the $\exists$-rule. Let $\mathcal{G}_1$ be the revision graph returned by this procedure, then its ABox representation is $\mathcal{A}_{\mathcal{G}_1} = \{A(a), C(a), R(a, x_1), A(x_1), C(x_1), R(x_1, x_2), A(x_2), C(x_2)\}$.

Consider the adaptation setting $\mathcal{AS}_2$ described in Example 6 and call the procedure B-MW($\langle \mathcal{T}_2, \mathcal{A}_2 \rangle, 1$). During the execution of step **2**, three variables y_1, z_1 and z_2 will be introduced by the $\exists$-rule. Let $\mathcal{G}_2'$ be the revision graph after the execution of step **3**, then its ABox representation is $\mathcal{A}_{\mathcal{G}_2'} = \{A(a), C(a), R(a, y_1), C(y_1), R(a, z_1), A(z_1), C(z_1), R(z_1, z_2), A(z_2), C(z_2)\}$. In $\mathcal{G}_2'$ there exists a redundant branch which will be removed by step **4**. Finally, let $\mathcal{G}_2$ be the revision graph returned by the procedure, then its ABox representation is $\mathcal{A}_{\mathcal{G}_2} = \{A(a), C(a), R(a, z_1), A(z_1), C(z_1), R(z_1, z_2), A(z_2), C(z_2)\}$. □

GCI$_I$-rule:	if $x \in N_I^{\mathcal{K}}$, $C \sqsubseteq D \in \mathcal{T}$, $D \notin \mathcal{L}(x)$, and $\langle \mathcal{T}, \mathcal{A} \rangle \models C(x)$, then set $\mathcal{L}(x) := \mathcal{L}(x) \cup \{D\}$.
GCI$_V$-rule:	if $x \notin N_I^{\mathcal{K}}$, $C \sqsubseteq D \in \mathcal{T}$, $D \notin \mathcal{L}(x)$, and $\langle \mathcal{T}, \{E(x) \mid E \in \mathcal{L}(x)\} \rangle \models C(x)$, then set $\mathcal{L}(x) := \mathcal{L}(x) \cup \{D\}$.
$\sqcap$-rule:	if $C_1 \sqcap C_2 \in \mathcal{L}(x)$, and $\{C_1, C_2\} \nsubseteq \mathcal{L}(x)$, then set $\mathcal{L}(x) := \mathcal{L}(x) \cup \{C_1, C_2\}$.
$\exists$-rule:	if $\exists R.C \in \mathcal{L}(x)$, $level(x) \leq k$, and x has no successor z with $C \in \mathcal{L}(z)$, then introduce a new variable z, set $V := V \cup \{z\}$, $E := E \cup \{\langle x, z \rangle\}$, $\mathcal{L}(z) := \{C\}$, and $\mathcal{L}(\langle x, z \rangle) := \{R\}$.

Fig. 1. Expansion rules used by the procedure B-MW($\mathcal{K}$, k)

Given a TBox $\mathcal{T}$ and an ABox representation $\mathcal{A}_\mathcal{G}$ of some revision graph $\mathcal{G}$, we use the procedure Rolling($\mathcal{A}_\mathcal{G}$, $\mathcal{T}$) to roll up variables contained in $\mathcal{A}_\mathcal{G}$.

```
Procedure Rolling($\mathcal{A}_\mathcal{G}$, $\mathcal{T}$)
   Input: an ABox $\mathcal{A}_\mathcal{G}$ that may contain variables, and a TBox $\mathcal{T}$.
   Output: an ABox $\mathcal{A}$ without variables.
1  Transform $\mathcal{A}_\mathcal{G}$ into its revision-graph representation $\mathcal{G} = (V, E, \mathcal{L})$.
2  Delete from $V$ the variables which are not descendants of any individual name.
3  while there exists variable in $V$ do
       select a variable $y \in V$ that has no successor;
       $x$ := the predecessor of $y$;
       if $\mathcal{L}(y) \neq \emptyset$ then $C_y := \bigsqcap_{C \in \mathcal{L}(y)} C$ else $C_y := \top$;
       $R$ := the role name contained in $\mathcal{L}(\langle x, y \rangle)$;
       if $\langle \mathcal{T}, \{D(x) \mid D \in \mathcal{L}(x)\} \rangle \not\models (\exists R.C_y)(x)$ then $\mathcal{L}(x) := \mathcal{L}(x) \cup \{\exists R.C_y\}$;
       $E := E \setminus \{\langle x, y \rangle\}$;
       $V := V \setminus \{y\}$.
4  Return $\mathcal{A} := \bigcup_{x \in V} \{C(x) \mid C \in \mathcal{L}(x)\} \cup \bigcup_{\langle x,y \rangle \in E} \{R(x, y) \mid R \in \mathcal{L}(\langle x, y \rangle)\}$.
```

Example 8. Let us continue Example 7. Both the procedure Rolling($\mathcal{A}_{\mathcal{G}_1}$, $\mathcal{T}_1$) and the procedure Rolling($\mathcal{A}_{\mathcal{G}_2}$, $\mathcal{T}_2$) return the same ABox $\{A(a), C(a)\}$.

Let $\mathcal{A}'_{\mathcal{G}_1} = \mathcal{A}_{\mathcal{G}_1} \setminus \{A(a), A(x_1)\} = \{C(a), R(a, x_1), C(x_1), R(x_1, x_2), A(x_2), C(x_2)\}$, and let $\mathcal{A}''_{\mathcal{G}_1} = \mathcal{A}_{\mathcal{G}_1} \setminus \{A(a), R(a, x_1)\} = \{C(a), A(x_1), C(x_1), R(x_1, x_2), A(x_2), C(x_2)\}$. Then the procedure Rolling($\mathcal{A}'_{\mathcal{G}_1}$, $\mathcal{T}_1$) returns the ABox $\{C(a), \exists R.(C \sqcap \exists R.(A \sqcap C))(a)\}$, and Rolling($\mathcal{A}''_{\mathcal{G}_1}$, $\mathcal{T}_1$) returns the ABox $\{C(a)\}$. □

5.2 The Adaptation Algorithm

Let $\mathcal{T}$ be a TBox, and let $\mathcal{A}$, $\mathcal{N}$ be two ABoxes. If $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{N} \rangle \models \top \sqsubseteq \bot$, then:

– a set $\mathcal{J} \subseteq \mathcal{A}$ is a $(\mathcal{A}, \mathcal{N})$-*justification* for a clash w.r.t. $\mathcal{T}$ if $\langle \mathcal{T}, \mathcal{J} \cup \mathcal{N} \rangle \models \top \sqsubseteq \bot$ and $\langle \mathcal{T}, \mathcal{J}' \cup \mathcal{N} \rangle \not\models \top \sqsubseteq \bot$ for every $\mathcal{J}' \subset \mathcal{J}$;

– a set $\mathcal{R} \subseteq \mathcal{A}$ is a $(\mathcal{A},\mathcal{N})$-*repair* for clashes w.r.t. $\mathcal{T}$ if $\mathcal{R} \cap \mathcal{J} \neq \emptyset$ for every $(\mathcal{A},\mathcal{N})$-justification $\mathcal{J}$, and for every $\mathcal{R}_i \subset \mathcal{R}$ there must be some $(\mathcal{A},\mathcal{N})$-justification $\mathcal{J}_i$ such that $\mathcal{R}_i \cap \mathcal{J}_i = \emptyset$.

Now we are ready to present our algorithm. Given an adaptation setting $\mathcal{AS} = (\mathcal{T},\mathcal{A},\mathcal{N})$ and any integer k, where k is greater than the role depths of all the concepts occurring in $\mathcal{AS}$, the algorithm Adaptation($\mathcal{AS}$, k) operates as follows. Firstly, a revision graph $\mathcal{G}$ will be constructed for the KB $\langle\mathcal{T},\mathcal{A}\rangle$ and the integer k. Secondly, a revision process based on justifications will be carried out on the ABox representation $\mathcal{A}_\mathcal{G}$ of $\mathcal{G}$. Thirdly, for each maximal subset $\mathcal{A}_i$ of $\mathcal{A}_\mathcal{G}$ that does not conflict with $\mathcal{N}$ and $\mathcal{T}$, the procedure Rolling($\mathcal{A}_i$, $\mathcal{T}$) will be used to roll up variables and get an ABox $\mathcal{A}'_i$. Finally, the ABox $\mathcal{A}'_i \cup \mathcal{N}$ will be returned as a solution case. Together with the solution case, an ABox $\mathcal{R}_i$ will also be returned; the function of $\mathcal{R}_i$ is to record the information which is entailed by $\mathcal{A}$ but removed in $\mathcal{A}'_i$, so that the user can select the best solution according to it.

Algorithm 1. Adaptation($\mathcal{AS}$, k)

```
Input: an adaptation setting $\mathcal{AS} = (\mathcal{T},\mathcal{A},\mathcal{N})$, and a non-negative integer $k$.
Output: a finite number of pairs $(\mathcal{A}'_1 \cup \mathcal{N}, \mathcal{R}_1)$, ..., $(\mathcal{A}'_n \cup \mathcal{N}, \mathcal{R}_n)$, where $\mathcal{A}'_i \cup \mathcal{N}$
        is a solution case and $\mathcal{R}_i$ records the information been removed.
if $\mathcal{A} \cup \mathcal{N}$ is consistent w.r.t. $\mathcal{T}$ then
    return $(\mathcal{A} \cup \mathcal{N}, \emptyset)$;
else
    $\mathcal{G}$ := B-MW$(\langle\mathcal{T},\mathcal{A}\rangle, k)$;
    $\mathcal{A}_\mathcal{G}$ := the ABox representation of $\mathcal{G}$;
    $S_\mathcal{R}$ := $\{\mathcal{R}_1, ..., \mathcal{R}_n\}$ all the $(\mathcal{A}_\mathcal{G},\mathcal{N})$-repairs for clashes w.r.t. $\mathcal{T}$;
    for $i \leftarrow 1$ to $n$ do
        $\mathcal{A}_i$ := $\mathcal{A}_\mathcal{G} \setminus \mathcal{R}_i$;
        $\mathcal{A}'_i$ := Rolling($\mathcal{A}_i$, $\mathcal{T}$);
    return $(\mathcal{A}'_1 \cup \mathcal{N}, \mathcal{R}_1)$, ..., $(\mathcal{A}'_n \cup \mathcal{N}, \mathcal{R}_n)$.
```

Example 9. Consider the adaptation setting $\mathcal{AS}_1 = (\mathcal{T}_1,\mathcal{A}_1,\mathcal{N}_1)$ described in Example 4. Since $\max\{depth(\mathcal{T}_1), depth(\mathcal{A}_1), depth(\mathcal{N}_1)\} = 1$, we select k=1 and execute the algorithm Adaptation$(\mathcal{AS}_1, 1)$.

Firstly, by calling the procedure B-MW$(\langle\mathcal{T}_1,\mathcal{A}_1\rangle, 1)$, we get a revision graph $\mathcal{G}_1$ for which the ABox representation is $\mathcal{A}_{\mathcal{G}_1} = \{A(a),\ C(a),\ R(a,x_1),\ A(x_1),\ C(x_1),\ R(x_1,\ x_2),\ A(x_2),\ C(x_2)\}$.

Secondly, for the clash $\langle\mathcal{T}_1, \mathcal{A}_{\mathcal{G}_1} \cup \mathcal{N}_1\rangle \models \top \sqsubseteq \bot$, there are two $(\mathcal{A}_{\mathcal{G}_1},\mathcal{N}_1)$-justifications $\mathcal{J}_1 = \{A \sqsubseteq \exists R.A,\ E \sqcap \exists R.A \sqsubseteq \bot,\ E(a),\ A(a)\}$ and $\mathcal{J}_2 = \{E \sqcap \exists R.A \sqsubseteq \bot, E(a), A(a), R(a,x_1), A(x_1)\}$. From them we get two $(\mathcal{A}_{\mathcal{G}_1},\mathcal{N}_1)$-repairs $\mathcal{R}_1 = \{A(a), A(x_1)\}$ and $\mathcal{R}_2 = \{A(a), R(a,x_1)\}$.

Thirdly, from $\mathcal{R}_1$ we get $\mathcal{A}'_1$ = Rolling($\mathcal{A}_{\mathcal{G}_1} \setminus \mathcal{R}_1$, $\mathcal{T}_1$) = $\{C(a), \exists R.(C \sqcap \exists R.(A \sqcap C))(a)\}$. From $\mathcal{R}_2$ we get $\mathcal{A}'_2$ = Rolling($\mathcal{A}_{\mathcal{G}_1} \setminus \mathcal{R}_2$, $\mathcal{T}_1$) = $\{C(a)\}$.

Finally, the algorithm returns $(\mathcal{A}'_1 \cup \mathcal{N}_1, \mathcal{R}_1)$ and $(\mathcal{A}'_2 \cup \mathcal{N}_1, \mathcal{R}_2)$, from them the user can select the best solution case with the help of $\mathcal{R}_1$ and $\mathcal{R}_2$. For example, since $\mathcal{R}_2$ contains a role assertion $R(a, x_1)$ which indicates that all the information related to x_1 is lost in $\mathcal{A}'_2$, the first choice for the user is $\mathcal{A}'_1 \cup \mathcal{N}_1$. $\square$

Example 10. Consider the adaptation setting $\mathcal{AS} = (\mathcal{T}, \mathcal{A}_{sol}, \mathcal{N}_{pb})$ described in Example 3. Since $\max\{depth(\mathcal{T}), depth(\mathcal{A}_{sol}), depth(\mathcal{N}_{pb})\} = 2$, we select $k = 2$ and execute the algorithm Adaptation$(\mathcal{AS}, 2)$.

The algorithm will return two results $(\mathcal{A}'_1 \cup \mathcal{N}_{pb}, \mathcal{R}_1)$ and $(\mathcal{A}'_2 \cup \mathcal{N}_{pb}, \mathcal{R}_2)$, where $\mathcal{A}'_1 = \{\exists TreatBy.\ (Anti\text{-}oestrogen \sqcap \exists metabolizedTo.(Compounds \sqcap \exists bindto.OestrogenReceptor))\ (Mary)\}$, $\mathcal{R}_1 = \{Tamoxifen(y)\}$, $\mathcal{A}''_2 = \emptyset$, and $\mathcal{R}_2 = \{TreatBy(Mary, y)\}$.

Since $\mathcal{R}_2$ contains a role assertion $TreatBy(Mary, y)$ which indicates that all the information related to y is lost in $\mathcal{A}'_2$, the first choice for the user is the solution case $\mathcal{A}'_1 \cup \mathcal{N}_{pb}$. This solution case states that, for the target problem described by $\mathcal{N}_{pb}$, a solution is to treat $Mary$ by something that is not only anti-oestrogen but also can be metabolized into some compounds which can be bound to oestrogen receptors. $\square$

The following theorems state that our algorithm satisfies **R1-R4**.

Theorem 1. *Let* $(\mathcal{A}''_i, \mathcal{R}_i)$ $(1 \leq i \leq n)$ *be the pairs returned by Adaptation*$(\mathcal{AS}, k)$ *for* $\mathcal{AS} = (\mathcal{T}, \mathcal{A}, \mathcal{N})$. *Then the following statements hold for every* $1 \leq i \leq n$: (1) $\langle \mathcal{T}, \mathcal{A}''_i \rangle \models \mathcal{N}$; (2) $\mathcal{A}''_i = \mathcal{A} \cup \mathcal{N}$ *if* $\mathcal{A} \cup \mathcal{N}$ *is consistent w.r.t.* $\mathcal{T}$; *and* (3) *if* $\mathcal{N}$ *is consistent w.r.t.* $\mathcal{T}$ *then* $\mathcal{A}''_i$ *is also consistent w.r.t.* $\mathcal{T}$.

Theorem 2. *Given two adaptation settings* $\mathcal{AS}_i = (\mathcal{T}, \mathcal{A}_i, \mathcal{N}_i)$ $(i = 1, 2)$ *and an integer* k, *where* $\langle \mathcal{T}, \mathcal{A}_1 \rangle \equiv \langle \mathcal{T}, \mathcal{A}_2 \rangle$, $\langle \mathcal{T}, \mathcal{N}_1 \rangle \equiv \langle \mathcal{T}, \mathcal{N}_2 \rangle$, $k \geq \max\{depth(\mathcal{T}), depth(\mathcal{A}_1), depth(\mathcal{A}_2), depth(\mathcal{N}_1), depth(\mathcal{N}_2)\}$. *If* $(\mathcal{A}''_1, \mathcal{R}_1)$ *is a pair returned by the algorithm Adaptation*$(\mathcal{AS}_1, k)$, *then there must be a pair* $(\mathcal{A}''_2, \mathcal{R}_2)$ *returned by Adaptation*$(\mathcal{AS}_2, k)$ *such that* $\langle \mathcal{T}, \mathcal{A}''_1 \rangle \equiv \langle \mathcal{T}, \mathcal{A}''_2 \rangle$ *and* $\mathcal{R}_2 = \sigma(\mathcal{R}_1)$ *for some substitution* σ *of variables.*

Theorem 2 is based on the following fact: let $\mathcal{G}_i =$ B-MW$(\langle \mathcal{T}, \mathcal{A}_i \rangle, k)$ $(i = 1, 2)$, then $\mathcal{G}_1$ and $\mathcal{G}_2$ are identical up to variable renaming in the case that k is sufficiently large. For Theorem 1 there is no requirement on the value of k.

In our algorithm, the revision graph $\mathcal{G}$ constructed by the procedure B-MW$(\langle \mathcal{T}, \mathcal{A} \rangle, k)$ is in fact a non-redundant k-depth-bounded model for the KB $\langle \mathcal{T}, \mathcal{A} \rangle$. Therefore, our revision process works on fine-grained representation of models and guarantees the minimal change principle in a fine-grained level. So, our algorithm also satisfies the property specified by **R5**.

The following theorem states that our algorithm is in exponential time.

Theorem 3. *For any adaptation setting* $\mathcal{AS} = (\mathcal{T}, \mathcal{A}, \mathcal{N})$, *assume the role depth of every concept occurring in* $\mathcal{AS}$ *is bounded by some integer* k, *then the algorithm Adaptation*$(\mathcal{AS}, k)$ *runs in time exponential with respect to the size of* $\mathcal{AS}$.

6 Discussion and Related Work

The idea of applying KB revision theory to adaptation in CBR was proposed by Lieber [17]. Based on a classical revision operator in propositional logic, a framework for adaptation was presented and it was demonstrated that the adaptation process should satisfy the AGM postulates. This idea was extended by Cojan and Lieber [7] to deal with adaptation based on the DL $\mathcal{ALC}$. Based on an extension of the classical tableau method used for deductive inferences in $\mathcal{ALC}$, an algorithm for adapting cases represented in $\mathcal{ALC}$ was proposed. It was shown that, except for the requirements on syntax-independence and minimality of change (i.e., **R4** and **R5** in our paper), all the other requirements specified by the AGM postulates (i.e., **R1**-**R3** in our paper) are satisfied by their algorithm.

From the point of view of KB revision in DLs, it is a great challenge to design revision operators or algorithms that satisfy the requirements specified by the AGM postulates. In the literature, there are two kinds of approaches, i.e., MBAs [14] and FBAs [4,16,20], for the instance-level KB revision problem in DLs. As we analyzed in Section 4, they either do not satisfy the requirements specified by **R4** and **R5**, or only work well for DLs of the *DL-Lite* family.

Our method is closer in spirit to the formula-based approaches, but it also inherits some ideas of model-based ones. On the one hand, in our algorithm, the revision graph $\mathcal{G}$ constructed by the procedure B-MW($\langle\mathcal{T},\mathcal{A}\rangle,k$) can be seen as a non-redundant, k-depth-bounded model for the KB $\langle\mathcal{T},\mathcal{A}\rangle$, and therefore our revision process essentially works on models. On the other hand, our revision process makes use of $(\mathcal{A}_\mathcal{G},\mathcal{N})$-repairs which inherits some ideas of FMAs based on justifications. As a result, our algorithm not only satisfies the requirements **R4** and **R5**, but also works for the DL $\mathcal{EL}_\perp$.

Given an adaptation setting, our algorithm will return a finite number of pairs $(\mathcal{A}''_i,\mathcal{R}_i)$ ($1 \leq i \leq n$), and it is left to the user to select the best solution case according to the sets $\mathcal{R}_i$. We can extend the algorithm to sort all the solution cases by priority. For example, if some $\mathcal{R}_i$ contains a role assertion $R(a,x)$ with x a variable, then the corresponding solution case $\mathcal{A}''_i$ will has a lower priority. Furthermore, we can define a selection function according to the user's selection criteria, and enable our algorithm to return only one best solution case.

7 Conclusion and Future Work

We studied the adaptation problem of CBR in the DL $\mathcal{EL}_\perp$. A formalism for adaptation based on $\mathcal{EL}_\perp$ was presented, and in this formalism the adaptation task was modeled as the instance-level KB revision problem in $\mathcal{EL}_\perp$. We illustrated that existing revision operators and algorithms in DLs did not work for the adaptation setting based on $\mathcal{EL}_\perp$ and then presented a new algorithm. We showed that our algorithm behaves well for $\mathcal{EL}_\perp$ in that it satisfies the requirements proposed in the literature for revision operators.

For future work, we will extend our method to support adaptation based on $\mathcal{EL}^{++}$ [3]. Another work is to implement and optimize our algorithm and test its feasibility in practice. Finally, we will formalize the notion of minimality of change under the framework of adaptation.

Acknowledgments. The authors would like to thank the reviewers for their helpful comments and suggestions. This work was partially supported by the National Natural Science Foundation of China (Nos. 61363030, 61262030), the Natural Science Foundation of Guangxi Province (No.2012GXNSFBA053169) and the Science Foundation of Guangxi Key Laboratory of Trusted Software.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. J. Symb. Log. 50(2), 510–530 (1985)
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, Cambridge (2003)
3. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proc. of the 19th International Joint Conference on Artificial Intelligence, pp. 364–369. Morgan Kaufmann (2005)
4. Calvanese, D., Kharlamov, E., Nutt, W., Zheleznyakov, D.: Evolution of *DL-lite* knowledge bases. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 112–128. Springer, Heidelberg (2010)
5. Chang, L., Sattler, U., Gu, T.L.: Algorithm for adapting cases represented in a tractable description logic. arXiv: 1405.4180 (2014)
6. Chang, L., Sattler, U., Gu, T.L.: An ABox revision algorithm for the description logic $\mathcal{EL}_\perp$. In: Proc. of the 27th International Workshop on Description Logics (2014)
7. Cojan, J., Lieber, J.: An algorithm for adapting cases represented in an expressive description logic. In: Bichindaritz, I., Montani, S. (eds.) ICCBR 2010. LNCS, vol. 6176, pp. 51–65. Springer, Heidelberg (2010)
8. Consortium, T.G.O.: Gene Ontology: Tool for the unification of biology. Nature Genetics 25, 25–29 (2000)
9. d'Aquin, M., Lieber, J., Napoli, A.: Decentralized case-based reasoning for the semantic web. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 142–155. Springer, Heidelberg (2005)
10. Gómez-Albarrán, M., González Calero, P.A., Díaz-Agudo, B., Fernández-Conde, C.J.: Modelling the CBR life cycle using description logics. In: Althoff, K.-D., Bergmann, R., Karl Branting, L. (eds.) ICCBR 1999. LNCS (LNAI), vol. 1650, pp. 147–161. Springer, Heidelberg (1999)
11. Horrocks, I., Patel-Schneider, P.F., Harmelen, F.V.: From SHIQ and RDF to OWL: the making of a web ontology language. J. Web Semantics 1(1), 7–26 (2003)
12. Horrocks, I., Sattler, U.: A tableau decision procedure for $\mathcal{SHOIQ}$. J. Autom. Reasoning 39(3), 249–276 (2007)
13. Katsuno, H., Mendelzon, A.O.: Propositional knowledge base revision and minimal change. Artificial Intelligence 52(3), 263–294 (1991)
14. Kharlamov, E., Zheleznyakov, D., Calvanese, D.: Capturing model-based ontology evolution at the instance level: The case of DL-Lite. J. Comput. Syst. Sci. 79(6), 835–872 (2013)

15. Lehmann, K., Turhan, A.-Y.: A Framework for Semantic-Based Similarity Measures for $\mathcal{ELH}$-Concepts. In: del Cerro, L.F., Herzig, A., Mengin, J. (eds.) JELIA 2012. LNCS, vol. 7519, pp. 307–319. Springer, Heidelberg (2012)
16. Lenzerini, M., Savo, D.F.: On the evolution of the instance level of DL-Lite knowledge bases. In: Proc. of the 24th International Workshop on Description Logics (2011)
17. Lieber, J.: Application of the revision theory to adaptation in case-based reasoning: the conservative adaptation. In: Weber, R.O., Richter, M.M. (eds.) ICCBR 2007. LNCS (LNAI), vol. 4626, pp. 239–253. Springer, Heidelberg (2007)
18. Sánchez-Ruiz, A.A., Ontañón, S., González-Calero, P.A., Plaza, E.: Measuring similarity in description logics using refinement operators. In: Ram, A., Wiratunga, N. (eds.) ICCBR 2011. LNCS, vol. 6880, pp. 289–303. Springer, Heidelberg (2011)
19. Spackman, K.: Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. J. American Medical Informatics Assoc., Fall Symposium Special Issue (2000)
20. Wiener, C.H., Katz, Y., Parsia, B.: Belief base revision for expressive description logics. In: Proc. of the 4th International Workshop on OWL: Experiences and Directions (2006)