Updating Description Logics using the AGM Theory

Giorgos Flouris, Dimitris Plexousakis, Grigoris Antoniou
Institute of Computer Science, FO.R.T.H.
P.O. Box 1385, GR 71110, Heraklion, Greece
{fgeo, dp, antoniou}@ics.forth.gr

Abstract
We consider the use of belief change techniques to address the problem of updating Knowledge Bases (KBs) based on Description Logics (DLs). We focus on the feasibility of the application of the AGM theory in DL KBs, evaluate the difficulties of the approach and determine the applicability of the method in certain families of DLs. For those DLs that are found compatible with the AGM model, we also describe a contraction operator that satisfies the AGM postulates. Finally, as an application of interest in the area of the Semantic Web, we study OWL, a W3C recommendation, and show that it is incompatible with the AGM model.

1 Introduction
The development of computer systems that can perform commonsense reasoning is a complex and multifaceted task. One of the crucial tasks that such a system should perform is to maintain a KB and update it dynamically to accommodate new information. This problem has been extensively studied for certain classes of representation languages in the area of belief change [Gärdenfors, 1992], the dominating approach being the work by Alchourron, Gärdenfors and Makinson (AGM) in [Alchourron et al., 1985].

Despite its apparent importance, the problem of KB updating has been generally overlooked [Lee and Meyer, 2004] in the literature relevant to DLs, an important family of logics [Baader et al., 2002]. We believe it is useful to apply the results of the belief change literature (more specifically the AGM theory) to this problem. This approach has been recently considered in [Flouris et al., 2004a; Kang and Lau, 2004; Lee and Meyer, 2004].

Unfortunately, the AGM model cannot be directly applied to DL KBs, as the assumptions originally made by AGM overrule DLs. We addressed this problem in [Flouris et al., 2004a; Flouris et al., 2004b], where we introduced a generalization of the AGM theory and determined the feasibility of the approach in a wide class of logics. In the current paper we specialize our results to DLs by formulating conditions under which it can be decided whether the generalized AGM theory is applicable in a DL or not. Furthermore, we apply our method to some specific DLs that have been considered in the literature, the most important one being OWL [Dean et al., 2004], a language with several applications in the Semantic Web.

2 Preliminaries

2.1 Description Logics
The term Description Logics [Baader et al., 2002] refers to a family of knowledge representation languages, heavily used in the Semantic Web. The basic blocks that are used to represent knowledge in DLs are classes (representing concepts), roles (representing binary relationships between concepts) and individuals (representing individual objects). These are used to form more complex expressions (terms) using certain operators. Knowledge is represented using axioms; an axiom represents a certain relationship (such as inclusion, membership and others) between terms using certain connectives.

The part of the DL KB dealing with concepts and roles is called the Tbox, while individuals are described in the Abox. The operators and connectives that a certain DL admits determine the type and complexity of the available axioms, which, in turn, determine the expressive power and the reasoning complexity of the DL.

The formal semantics of DLs is defined using interpretations I which consist of a non-empty set A and a function \( \mathcal{I} \) which maps every concept A to a set \( A^I \subseteq A \), every role R to a relation \( R^I \subseteq A^I \times A^I \) and every individual x to an object \( x^I \in A \). This mapping is extended to terms using the semantics of each operator; for example \( (A \sqcap B)^I = A^I \cap B^I \). The semantics of each connective determines whether an axiom is satisfied by a certain interpretation; for example the axiom \( A \sqcap B \sqsubseteq C \) is satisfied by the interpretation I iff \( (A \sqcap B)^I \subseteq C^I \). Analogously, a set of axioms is satisfied by an interpretation I iff all the axioms in the set are satisfied by I. A set of axioms X implies Y (X \( \vdash \) Y) iff all interpretations I that satisfy X also satisfy Y. A more detailed account on the semantics of the operators and connectives of DLs can be found in [Baader et al., 2002].

The importance of DLs stems from the fact that they play a primary role in the area of representation languages for ontologies [Baader et al., 2003]. For this reason, any research on DLs is expected to find immediate applications in the field of the Semantic Web. The aim of our research is to study the update-theoretic properties of DLs, an area of DL research which has been generally disregarded in the literature [Lee and Meyer, 2004]. This study may help in the automation of the task of ontology maintenance by indicating rational methods of ontology updating.
2.2 Belief Change and the AGM Postulates

The AGM theory [Alchourron et al., 1985] is undoubtedly the most influential work in the area of belief change. In this work, three fundamental operators of belief change were defined, as well as a set of rationality postulates (commonly referred to as the AGM postulates) that should apply to each belief change operation. The importance of the above model lies in the fact that AGM defined some widely accepted properties that any rational belief change operator should satisfy, thus setting the foundations for future research on the subject.

In our paper, we restrict our attention to the operation of contraction (denoted by ‘−’) which refers to the removal of a piece of information from a KB when this information is no longer believed. Contraction was chosen because, according to AGM, it is the most fundamental among the three belief change operators. Indeed, the theoretical importance of contraction has been accepted by most researchers, even though in practical applications revision (which refers to addition of information) is more often used.

In the AGM model, contraction is a binary operator between a KB K closed under logical consequence and a proposition x. According to AGM, a contraction operator should satisfy six postulates, which reflect common intuition about the removal of information. These postulates can be found in [Alchourron et al., 1985].

2.3 Rationality of Belief Change

We believe that the concept of “rationality” of a belief change operator is independent of the underlying knowledge representation scheme. Since the AGM theory formally encodes common intuition about the properties that a rational belief change operator should satisfy, we believe that it would be of interest to apply the AGM model to all logics and operators in order to determine the “rationality” of a given belief change operator in a given logic. This includes DLs, which are the focus of this paper.

Unfortunately though, the AGM theory itself does depend on the underlying language, since it is based on certain assumptions which place it in a certain context, making it applicable mainly in the context of Propositional and First-Order Logic. For this reason, the results of AGM cannot be directly applied to many interesting logics (including DLs).

Indeed, many AGM assumptions fail for DLs. For example, AGM assume the existence of certain operators in the logic such as ⊃, ∧ etc. This is not generally true for DLs; for example, the negation of an axiom x (say x="A∩B⊆C") cannot be defined in general. Furthermore, many DLs are not compact, which is another AGM assumption.

This problem was originally noticed and addressed in a very general context in [Flouris et al., 2004a], by reformulating the AGM postulates to make sense in a class of logics far wider than the one AGM originally considered. In that paper, we reformulated the postulates so that they can be applied to any logic that is definable as a pair <L,Cn>, where L is a set containing all the expressions of the logic and Cn is a consequence operator that satisfies the Tarskian axioms (iteration, inclusion, monotony).

Furthermore, we assumed that contraction is a binary operator applicable between two sets K, A, which are sets of expressions of the underlying logic (K,A⊆L). Following that, we reformulated the AGM postulates to make sense in this more general context. The generalized postulates that resulted from this work can be found in the following list, where the naming and numbering of each postulate corresponds to the original AGM naming and numbering:

(K-1) Closure: \( Cn(K-A)=K-A \)
(K-2) Inclusion: \( K-A \subseteq Cn(K) \)
(K-3) Vacuity: \( \text{If } A \not\subseteq Cn(K), \text{ then } K-A = Cn(K) \)
(K-4) Success: \( \text{If } A \not\subseteq Cn(\emptyset), \text{ then } A \not\subseteq Cn(K-A) \)
(K-5) Preservation: \( \text{If } Cn(A)=Cn(B), \text{ then } K-A = K-B \)
(K-6) Recovery: \( K \subseteq Cn((K-A) \cup A) \)

2.4 AGM-compliance

Upon attempt to apply the generalized AGM postulates to certain logics, we noticed that this was possible only in some of the logics in our generalized framework. In other words, several logics (not in the original AGM framework) cannot admit any contraction operator that satisfies the generalized AGM postulates. On the bright side, there are also logics outside the original AGM framework, in which a contraction operator that satisfies the generalized AGM postulates can be defined.

Following this observation, we introduced the notion of AGM-compliant logics. A logic was defined to be AGM-compliant iff a contraction operator that satisfies the six generalized AGM postulates can be defined in this logic. This class of logics was characterized using the following proposition [Flouris et al., 2004b]:

**Proposition 1** A logic <L,Cn> is AGM-compliant iff for all sets X, Y⊆L such that Cn(∅)⊆Cn(Y)⊆Cn(X) there is a set Z⊆L such that Cn(Z)⊆Cn(X) and Cn(Y∪Z)=Cn(X).

Using this result, any given logic can be checked for AGM-compliance, provided that it is definable as a pair <L,Cn>. In [Flouris et al., 2004b] we also showed that if the set Z required by proposition 1 actually exists, then it can be used to define an AGM-compliant contraction operator, by setting X=Y=Cn(Z).

The above results can be directly applied to DLs, since, for any given DL, we can take L to be the set of all possible axioms that can be formed in this DL (such as A⊆B∩∀R.C) and Cn(X) the set of all implications of a set of axioms X⊆L under the standard model-theoretic semantics of DLs [Baader et al., 2002].

In the current paper, we exploit this fact to determine the AGM-compliance of certain DLs, by formulating conditions that guarantee or bar AGM-compliance (sections 3 and 4 respectively). In section 5, we apply our results to certain DLs that have been considered in the literature, the most important one being OWL [Dean et al., 2004], a language with several applications in the Semantic Web, which has become a W3C recommendation. We conclude by discussing the merits as well as the limitations of our method and proposing interesting topics of future work on the subject (section 6).
3 Conditions for AGM-compliance

3.1 General Intuition and Initial Results

Throughout this section, we will assume a DL that allows for the top concept ⊤ and the connective ∈ (applicable to concept terms, at least), plus an arbitrary number of other connectives and/or operators. We will use the consequence operator Cn as given by the standard model-theoretic semantics of DLs [Baader et al., 2002].

The guiding intuition for our approach is the following observation: take two (singular) sets of DL axioms of the form X={A∈⊤}, Y={B∈⊤} for which it holds that Cn(Ø)⊆Cn(Y)⊆Cn(X). The set Z={A∈B} is a good candidate for the set required by proposition 1, since, obviously, Cn(Ø)⊆Cn(X) and Cn(Y∪Ø)=Cn(X). There is a catch though: proposition 1 requires that Cn(Z)⊆Cn(X); in the above naïve approach sometimes it so happens that Cn(Z)=Cn(X). To see this, take B=¬A∪∃R.A∪∀R.A for some role R (example provided by Thomas Studer, personal communication). To deal with this problem, the idea must be refined in order to guarantee that Cn(Z)⊆Cn(X). This refinement is described and proved in a more general setting in the following lemma:

Lemma 1 Consider the sets of axioms X={A∈∈ j∈J} and Y={B∈∈} for which it holds that Cn(Ø)⊆Cn(Y)⊆Cn(X). There is an interpretation I such that B∈Ø, then there is a set Z such that Cn(Z)⊆Cn(X) and Cn(Y∪Z)=Cn(X).

Proof: Set Z={A∈B | ∈ j∈J}. If we suppose that Z∈X, then, since X∈Y, we conclude that Z∈Y. However, by the hypothesis, there is an interpretation I such that B∈Ø; for this interpretation, Z is obviously satisfied, while Y is not, which is a contradiction. So Z∈X but X∈Z, i.e., Cn(Z)⊆Cn(X). The second relation, Cn(Y∪Z)=Cn(X), follows easily using the transitivity of ∈.

3.2 Generalizing to Arbitrary Axioms

Lemma 1 defines the set Z required by proposition 1, but only for sets of a special form. This might cause one to believe that it is of limited use; on the contrary, this lemma forms the backbone of our theory. Before stating that, we will show that the prerequisites of proposition 1 need only be checked for a subset of all the possible (X, Y) pairs:

Lemma 2 Consider a logic L∈Cn and two sets X, Y∈L, such that Cn(Ø)⊆Cn(Y)⊆Cn(X). If there are sets X′, Y′⊆L such that Cn(X′)=Cn(X), Cn(Ø)⊆Cn(Y′)⊆Cn(Y) and a set Z⊆L such that Cn(Z)⊆Cn(X′) and Cn(Y′∪Z)=Cn(X′) then Cn(Z)⊆Cn(X) and Cn(Y′∪Z)=Cn(X).

Proof: Obviously Cn(Z)⊆Cn(X′)=Cn(X).

Furthermore, since Cn(Y′)⊆Cn(Y) we conclude that Cn(Y′∪Z)=Cn(Y′∪Z)=Cn(X′)=Cn(X). Combining this with the facts that Cn(Y)⊆Cn(X) and Cn(Z)⊆Cn(X) we get Cn(Y′∪Z)=Cn(X) and the proof is complete.

Now consider a DL and any two sets of axioms X, Y such that Cn(Ø)⊆Cn(Y)⊆Cn(X). If X and Y are of the form required by lemma 1, then we are done; lemma 1 allows us to find a set Z that satisfies the requirements of proposition 1 for an AGM-compliant logic. If, on the other hand, X or Y are not of the desired form, lemma 2 shows the way; all we need is to find two sets X′, Y′ of the desired form such that Cn(X′)=Cn(X) and Cn(Ø)⊆Cn(Y′)⊆Cn(Y). Then, lemma 1 can be applied for X′, Y′ and the resulting set Z can be propagated to X, Y, using lemma 2. The important question is, are there sets X′, Y′ with the desired properties? The answer depends on the underlying DL.

Let us deal with X′ first. One possible way to find X′ is to take each axiom x∈X and transform it independently into the desired form (A∈⊤). A complete list of the relevant transformations and their required operators is provided in table 1. The transformation is possible for axioms involving concepts, roles and even individuals. Thus, the results presented here apply also to DL KBs that contain a non-empty Abox. In table 1, A, B refer to concept terms, R, S refer to role terms and a, b refer to individuals. All operators subscripted by R (in the third column) apply to role terms; the other operators apply to concept terms or individuals. In the second column, the subscript R has been dropped for

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Equivalent axiom of the proper form</th>
<th>Required operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>A∈B</td>
<td>¬A∪B∈⊤</td>
<td>¬A∪B∈⊤</td>
</tr>
<tr>
<td>R∈S</td>
<td>∀(R∈¬S).⊥∃∈</td>
<td>∥, R,∈, R, R</td>
</tr>
<tr>
<td>A∈B</td>
<td>(¬A∪B) ∪ (B∪¬A)∈</td>
<td>¬A∪B∈⊤, T, R ∈, R</td>
</tr>
<tr>
<td>R∈S</td>
<td>∀(R∈¬S).⊥∈</td>
<td>∥, R,∈, R, R</td>
</tr>
<tr>
<td>A∈B</td>
<td>∀(A∈B) ∪ (¬R∈¬S).⊥∈</td>
<td>∥, R,∈, R, R</td>
</tr>
<tr>
<td>R∈S</td>
<td>∀(R∈¬S).⊥∈</td>
<td>∥, R,∈, R, R</td>
</tr>
<tr>
<td>R∈S</td>
<td>∀(R∈¬S).⊥∈</td>
<td>∥, R,∈, R, R</td>
</tr>
<tr>
<td>R∈S</td>
<td>∀(R∈¬S).⊥∈</td>
<td>∥, R,∈, R, R</td>
</tr>
<tr>
<td>R∈S</td>
<td>∀(R∈¬S).⊥∈</td>
<td>∥, R,∈, R, R</td>
</tr>
<tr>
<td>R∈S</td>
<td>∀(R∈¬S).⊥∈</td>
<td>∥, R,∈, R, R</td>
</tr>
<tr>
<td>R∈S</td>
<td>∀(R∈¬S).⊥∈</td>
<td>∥, R,∈, R, R</td>
</tr>
<tr>
<td>R∈S</td>
<td>∀(R∈¬S).⊥∈</td>
<td>∥, R,∈, R, R</td>
</tr>
<tr>
<td>R∈S</td>
<td>∀(R∈¬S).⊥∈</td>
<td>∥, R,∈, R, R</td>
</tr>
</tbody>
</table>

Table 1: Transforming DL axioms into the form A∈⊤
readability purposes; the domain of the respective operator is obvious in each case by the context. Likewise, connectives apply to concepts, roles or individuals, depending on the context. The symbol \( T_R \) refers to the top role, i.e., the role connecting every individual to every individual and the connective \( \sqsubseteq \) stands for non-proper-inclusion. The symbols \( \neg \) and \( \exists \) refer to full (rather than atomic) negation and full (rather than limited) existential quantification respectively.

Not all the transformations in table 1 are straightforward; some of them use transformations appearing earlier in the table; all of them can be shown using model-theoretic arguments. What table 1 shows is that all the axiom types commonly used in DLs can be equivalently rewritten in the form \( A \sqsubseteq T \); this allows any set \( X \) to be transformed into an equivalent set \( X' \) of the form required by lemma 1, provided that the DL allows for the operators necessary for the transformation.

But what about \( Y' \)? If \( Y \) is not of the desired form, then the same procedure cannot be used, because lemma 1 requires that the contracting expression is a unary set of the form \( \{ B \sqsubseteq T \} \) and that there is an interpretation \( I \) for which \( B = \emptyset \). But lemma 2 allows \( Y' \) to be a (non-tautological) consequence of \( Y \). One possible way to find \( Y' \) is as follows: first select any non-tautological axiom of \( Y \) (there will definitely be such an axiom, as \( Cn(\emptyset) \subseteq Cn(Y) \)); then apply the relevant transformation of table 1 to this axiom. Suppose that the resulting axiom is \( B \sqsubseteq T \). Set \( Y' = \{ \forall T_R.B \sqsubseteq T \} \). Since the originally selected axiom is non-tautological, so is \( B \sqsubseteq T \); thus, there is an interpretation such that \( B \neq T \). It is easy to show that for this interpretation it holds that \( \{ \forall T_R.B \} = \emptyset \). Moreover, it is obvious that \( Y' = \{ B \sqsubseteq T \} \vdash \{ \forall T_R.B \sqsubseteq T \} \Rightarrow Y' \) and \( Cn(Y') \neq Cn(\emptyset) \), so \( Y' \) is of the desired form. Again, the above transformations cannot be performed unless the underlying DL admits the necessary operators.

Combining the above thoughts with proposition 1, we conclude that, if the DL under question contains enough operators to allow us to perform the necessary transformations, then it is AGM-compliant. The required operators are the constant \( T \) and the (concept) connective \( \sqsubseteq \) for the basic case (lemma 1), the operators of table 1 for the transformation of \( X \) plus the operator \( \forall \) and the constant \( T_R \) for the transformation of \( Y \). Notice that there is a certain amount of redundancy in table 1; for example, if \( \neg \) and \( \sqcap \) are included, then \( \sqcup \) is not necessary. By eliminating this redundancy the following theorem can be proven:

**Theorem 1** Assume a DL that contains the constants \( T \), \( T_R \), the operators \( \neg \), \( \sqcap \), \( \forall \), \( \neg R \), \{\ldots\}, the connective \( \sqsubseteq \) (applicable to concepts) plus any of the connectives of table 1. Then this DL is AGM-compliant.

### 3.3 Discussion and Further Generalizations

Theorem 1 is important because it verifies that one particular family of DLs is AGM-compliant. However, its main importance lies in its numerous variations; theorem 1 should be viewed primarily as a “sample” theorem showing one possible application of our approach, because it actually uncovers a whole pattern that can be used to show several similar theorems.

This is true because there are several different ways to generalize theorem 1 to show similar positive (AGM-compliance) results on DLs. First of all, the transformations of table 1 may not be the only possible ones. A similar table, containing similar transformations, would possibly generate a different set of operators required for AGM-compliance. The same holds for the transformation of \( Y \).

Secondly, theorem 1 gives a **minimal** set of operators that are needed to guarantee AGM-compliance. Any additional operators do not bar AGM-compliance (notice however that any additional connectives might). Thus, all logics that contain more operators than the DL described in theorem 1 are AGM-compliant too.

Furthermore, some of the operators could be replaced by others; for example the combination \( \{\neg, \forall\} \) is equivalent to the combination \( \{\neg, \exists\} \), using the well-known equivalence: \( \forall R.A \equiv \neg \exists R.\neg A \). Similar facts hold for other operators as well. Moreover, the constants \( \sqcap \) and \( T_R \) could be removed from the minimal required set of operators, because they can be replaced by \( A \sqcap \neg A \) and \( R \sqcap \neg R \) respectively. Of course, this requires that there is at least one concept \( (A) \) and at least one role \( (R) \) in the namespace of the logic, but this is hardly an assumption. After all, what would DLs be without concepts or roles?

It should also be noticed that the operators required for AGM-compliance are depending on the necessary transformations and that we need one transformation per axiom type. Therefore, the required operators are depending on the variety of axiom types allowed in the logic; for example, if we are interested in DL KBs without an Abox, then the operator \{\ldots\} is not necessary, i.e., it could be removed from the minimal set of operators required for AGM-compliance. Similarly, certain logics disallow certain connectives or certain uses of connectives. Such restrictions might affect (i.e., reduce) the required minimal operator set (by allowing simpler transformations). On the other hand, if a DL admits any exotic connectives that are not considered in table 1, then we might need to add some operators to our minimal set in order to render the relevant transformations possible, if they are at all possible.

In theorem 1 we state that the DL under question must admit concept hierarchies (connective \( \sqsubseteq \)). This is a reasonable assumption, since most interesting DLs do satisfy it. However, it turns out that it is also an unnecessary one. To show that, we will use the concept of **equivalence of logics** (with respect to AGM-compliance) that appeared in [Flouris et al., 2004b]. In the same paper it was shown that equivalent logics have the same status as far as AGM-compliance is concerned.

Using model-theoretic arguments, it is easy to show the following equivalences: \( \{A \sqsubseteq T\} \iff \{A \sqsubseteq T\} \iff \{\neg \forall T_R.A \sqsubseteq T\} \iff \{\neg \forall T_R.A \sqsubseteq T\} \iff \{A \sqsubseteq T\} \iff \{\text{disj}(T, \neg A)\} \). These equivalences are all definable using the minimal set of operators of theorem 1. Now, using these transformations and a proposition in [Flouris et al., 2004b] (proposition 5), we can show that a
4 Conditions for non-AGM-compliance

Unfortunately, not all DLs are compatible with the AGM theory. In our previous work [Flouris et al., 2004b], a certain family of DLs was shown to be non-AGM-compliant. In this section, we show that our original result can be generalized to a far wider class containing many logics that allow for role axioms such as role equivalences and role hierarchies. Initially, we will show this simple lemma that is applicable in any logic:

**Lemma 3** Consider a logic <L, Cn> and a set X ⊆ L. Set Y = {x ∈ Cn(X) | Cn(⟨x⟩) ⊆ Cn(X)}. If Cn(∅) ⊆ Cn(Y) ⊆ Cn(X) then <L, Cn> is not AGM-compliant.

**Proof** Take any Z ⊆ L such that Cn(Z) ⊆ Cn(X). For every z ∈ Z it holds that Cn(⟨z⟩) ⊆ Cn(Z) ⊆ Cn(X), so z ∈ Y. Therefore Cn(Y ∪ Z) = Cn(Y) ⊆ Cn(X); thus, for the pair (X, Y) it holds that Cn(∅) ⊆ Cn(Y) ⊆ Cn(X) and there is no Z ⊆ L with the properties required by proposition 1, so <L, Cn> is not AGM-compliant. □

Lemma 3 states that, if a logic contains a belief which cannot be deduced by all its proper consequences combined, then this logic is not AGM-compliant. Unfortunately, this is the case for many DLs that allow for role hierarchies and/or role equivalences, but forbid the use of the operators ¬R, R. Indeed, suppose that a logic allows one to define the axiom R ⊆ S. This axiom has several implications, such as ∃R.A ∈ ∃S.A, (≤R) ∈ (≤S) and others (see [Baader et al., 2002] for details on the above operators). Set X = {R ∈ S} and Y = {x ∈ Cn(X) | Cn(⟨x⟩) ⊆ Cn(X)}, as in lemma 3. For several DLs, it so happens that Cn(∅) ⊆ Cn(Y) ⊆ Cn(X), so by lemma 3 such logics are not AGM-compliant:

**Theorem 2** Suppose a DL with the following properties:

- The namespace contains at least two role names (say R, S) and at least one concept name (say A)
- The logic admits at least one of the operators ∀, ∃, (≤a), (≥a), for at least some n
- The logic admits any (or none, or all) of the operators and constants ¬, ∨, ⊓, ⊔, ⊤, ⊥, where the symbol = stands for the inverse role operator
- The logic admits only the connective ⊆ applicable to both concepts and roles

Then this DL is not AGM-compliant.

**Sketch of Proof**

Set X = {R ∈ S}, Y = {x ∈ Cn(X) | Cn(⟨x⟩) ⊆ Cn(X)} and define two interpretations I, I', as follows:

\[ \Delta^I = \Delta'^I = \{a_1, a_2, b_1, b_2\} \]

\[ B^I = B'^I = \emptyset \quad \text{for all concepts B} \]

\[ R_0^I = R_0'^I = \emptyset \quad \text{for all roles} \quad \text{R other than} \quad \text{R, S} \]

\[ R^I = R^I' = \{a_1, b_1\} \]

\[ S^I = \{a_1, a_2\} \]

\[ S'^I = \{b_1, b_2\} \]

Notice that the two interpretations differ only in the interpretation of the role S.

Also notice that for these two interpretations and for every concept term C definable in the DLs considered by this theorem, it holds that C ⊆ C'; this can be shown using induction on the number of operators of C. Thus, any axiom
involving concept terms is satisfied by I if and only if it is satisfied by I'.

Furthermore, the absence of operators for roles (except possibly \(\tau\)) in the DL under question allows us to show that all axioms in \(Y\) that describe relationships between roles are actually tautological.

Regarding the interpretations I, I', notice that I satisfies X (obviously), so it satisfies Y (because \(X \equiv Y\)), thus any axiom \(y \in Y\) is satisfied by I. If y involves roles, then it is tautological, so it is satisfied by I'. If y involves concepts, then, by the above result, y is satisfied by I' since it is satisfied by I. Thus Y is satisfied by I'. On the other hand, X is obviously not satisfied by I'.

The above shows that there is an interpretation (I') satisfying Y but not X, so \(\text{Cn}(Y) \subseteq \text{Cn}(X)\). The existence of one of the operators \(\lor, \exists, (\leq_n), (\leq)\) (for some n) and of at least one concept (A) in the DL under question guarantees that Y is also non-tautological, because at least one of the non-tautological axioms \(\exists R.A \equiv \exists S.A\), \(\forall R.A \equiv \forall S.A\), \((\leq_n R) \equiv (\leq S), (\leq R) \equiv (\leq S)\), for some n, will be in Y. Thus \(\text{Cn}(\emptyset) \subseteq \text{Cn}(Y) \subseteq \text{Cn}(X)\).

These results, combined with lemma 3, give us the conclusion. □

The above negative result persists if the DL under question admits \(\equiv\) (applicable to both concepts and roles) instead of \(\equiv\), or if it admits both connectives; the proof is identical. The same result can be shown (using the same proof) if we add transitive roles (the axiom Trans(\_)) and/or qualified number restrictions. Furthermore, the DL under question remains non-AGM-compliant if only functional roles are admitted (R and S are already defined to be functional in the interpretations of the proof of theorem 2).

A further generalization of theorem 2 occurs by adding individuals, axioms with individuals and/or the operator \(\{\ldots\}\) in the DL under question. The proof for this result is similar to the proof of theorem 2, the only difference being that the two interpretations I, I' must be defined in a slightly different way. More specifically, we must add a new object, say c, in their domain (i.e., set \(\Delta' = \Delta' = \{a_1, a_2, b_1, b_2, c\}\)) and map all individuals to this new object (i.e., set \(x' = x = c\) for all individuals x), leaving the rest unchanged. With these new interpretations, all the steps of the proof can be repeated without changes to give us the desired result.

This analysis uncovers a rule of thumb regarding DLs: if theorem 1 cannot be applied, then there is a good chance that the prerequisites of lemma 3 will hold for a set of the form \(\{R \subseteq S\}\) or \(\{R \subseteq S\}\) (for two roles R, S), so the DL under question is not AGM-compliant. This gives us a simple test to determine whether a DL is AGM-compliant or not, which is applicable to many (but not all) commonly used DLs.

5 Applications of our Method

5.1 Web Ontology Language (OWL)

As an application of the results shown in the previous sections, we study the Web Ontology Language [Dean et al., 2004], also known as OWL. OWL is a knowledge representation language that is expected to play an important role in the future of the Semantic Web, as it has become a W3C recommendation. OWL comes in three "flavors", namely OWL Full, OWL DL and OWL Lite, with varying degree of expressive power and reasoning complexity. Only OWL DL and OWL Lite could be classified as DLs: OWL Full contains features not normally allowed in DLs. Unfortunately, all three flavors of OWL turn out to be non-AGM-compliant.

One of the reasons that OWL is not AGM-compliant has to do with the owl:imports construct. This construct is a special meta-logical annotation property forcing the parser to include another KB (ontology) in the current KB. In effect, a singular set X of the form X={owl:imports(O)} implies exactly what is implied by the ontology O. However, X is not equivalent to O, because there is no guarantee that the included ontology will remain static and unchanged. If O is modified, then X would be equivalent to the new version of O, but the new version of O would not necessarily be equivalent to O. This fact shows that the owl:imports(O) axiom cannot be replaced by the axioms of O, i.e., X and O are not equivalent. In fact, it holds that \(\text{Cn}(X) = X \cap \text{Cn}(O)\) and \(X \not\subseteq \text{Cn}(O)\).

The above analysis shows that lemma 3 is applicable for X={owl:imports(O)}, verifying that all three flavors of OWL (which contain the owl:imports axiom) are not AGM-compliant. This is true unless we could somehow guarantee that all ontologies would remain unchanged once they are created, which is a highly unrealistic assumption for the Semantic Web.

However, acknowledging the importance of OWL in the Semantic Web and the fact that owl:imports is not an essential feature of the language, we did not end our study at this point; instead, we addressed the question of what happens if owl:imports is removed from the language. In this respect, [Horrocks and Patel-Schneider, 2004] is of great use, as it identifies OWL DL and OWL Lite (without annotations) as equivalent to certain DLs (namely SHOIN'(D) and SHIF*(D) respectively); both these DLs can be shown to be non-AGM-compliant, using a proof similar to the proof of theorem 2. Thus, OWL DL and OWL Lite are not AGM-compliant, even without their meta-logical features.

5.2 DLs in the Literature and AGM-compliance

Theorems 1 and 2 can be applied to several DLs that have already been considered in the literature. In this subsection we are providing an indicative (but not necessarily complete) list of DLs for which a definite answer regarding AGM-compliance can be given. For a definition of the logics below, refer to [Baader and Sattler, 2001; Baader et al., 2002; Horrocks and Patel-Schneider, 2004; Lutz and Sattler, 2000].

The following DLs can be shown to be non-AGM-compliant: SH, SHI, SHIN, SHOIN(D), SHOIN*, SHOIN'(D), SHIQ, SHIF, SHIF(D), SHIF*. SHIF'(D). None of the three flavors of OWL is AGM-compliant if the owl:imports axiom is included; OWL DL and OWL Lite without their annotation features are not AGM-compliant.
either. If axioms involving roles (with $\subseteq$ or $\supseteq$) are allowed, then $FL_0$, $FL$ and $ALC[U][E][N][C][Q][F]$ are non-AGM-compliant. All these facts can be proven using theorem 2 or similar arguments.

On the other hand, AGM-compliance can be achieved by adding role operators to the AL family. The logic $ALC(\forall R, \sqsubseteq R, \neg R)$ (with empty ABox) can be shown to be AGM-compliant. Notice that $\exists R$ and $\{\ldots\}$ are not included in $ALC(\forall R, \sqsubseteq R, \neg R)$, but this is not a problem, as discussed in section 3.3. Similarly, $ALC^\circ$, $ALC^{\leftarrow}$, $ALC^\leftarrow\neg$, $ALC^{\rightarrow\neg}$ and $ALC^\leftarrow\neg\leftarrow$ (with empty ABox and no axioms involving role terms) are AGM-compliant. The above positive results persist if we add more operators to any of the above logics; for example all of $ALC[U][E][N][Q][F](\forall R, \sqsubseteq R, \neg R)$ (with no ABox) are AGM-compliant. All these facts are direct applications of some variant of theorem 1.

The above analysis shows the importance of role operators in guaranteeing AGM-compliance in DLs used in practice; we are currently working on determining the validity of this conjecture, but the answer will likely be positive. Unfortunately, role operators appear in very few logics in the literature; for this reason, we highly encourage research on DLs that contain these operators, because of their nice behavior with respect to updates.

6 Conclusion and Future Work

It is generally acknowledged that belief change is a very important problem in the field of Artificial Intelligence, with applications in several research areas, including the area of commonsense reasoning. Indeed, part of a computer system’s commonsense behavior is the ability to rationally update its KB and adapt itself to new information received.

The AGM theory [Alchourron et al., 1985] is a mature and widely accepted approach in the belief change literature with several applications in a variety of fields. We believe that a further application of this theory in DLs is very promising and will hopefully indicate rational methods for DL KB (and ontology) updating.

The feasibility of this approach was initially studied in our previous work [Flouris et al., 2004a; Flouris et al., 2004b], where it was shown that the application of the AGM theory to logics outside the theory’s original scope (such as DLs) is not always possible.

In the current paper, we specialized our original results to DLs by determining whether contracting a DL KB using the AGM model is possible for certain DLs and providing a roadmap allowing one to check AGM-compliance for DLs not covered by this work. We also described one possible AGM-compliant contraction operator for the DLs that were found to allow one. We are hoping that our work will help in uncovering the limitations of the AGM theory with respect to DLs, by verifying the applicability of the method in some cases and forcing us to consider alternative approaches in other cases.

One important application of our work lies in the field of the Semantic Web, where DLs play a primary role in the area of representation languages for ontologies [Baader et al., 2003]. Our research has the potential to find applications in ontology merging and update and, consequently, in the automation of the task of ontology maintenance on the Semantic Web. Unfortunately though, one of the most important knowledge representation languages in the Semantic Web, OWL, was shown to be non-AGM-compliant.

Our study was kept at a fairly abstract level; we did not focus on any specific DLs but dealt with the DL family as a whole, including DLs that have not yet been considered in the literature. This approach has the extra benefit of allowing our results to be of use to researchers who develop new DLs; if the focus is on developing a DL that can be rationally updated, then AGM-compliance should be considered as a desirable feature of the new DL, along with high expressive power, low reasoning complexity etc. In this respect, we highly encourage research on DLs that admit role operators ($\neg R, \sqsubseteq R, \sqsubseteq R, \neg R$), as such DLs provably exhibit nice behavior with respect to AGM-compliance.

The issue of applying the AGM theory to DLs is by no means concluded with this paper. We are currently trying to develop a complete characterization of AGM-compliant DLs by seeking more general formulations of theorems 1 and 2 that will engulf all their possible variations. We are hoping that these more general theorems will turn out to be complete characterizations of AGM-compliance for DLs.

Furthermore, we are planning to refine the proposed contraction operator for AGM-compliant DLs, to produce an operator that will be based on semantic rather than syntactic considerations, in addition to being AGM-compatible.

Notice that AGM-compliance simply guarantees the existence of a rational (in the AGM sense) contraction operator in the given DL; an AGM-compliant DL does not necessarily satisfy all results related to the AGM theory, unless these results are not depending on the AGM assumptions. Consequently, AGM-compliance is not the final word on the relation of the AGM theory to DLs.

However, AGM-compliance does constitute a necessary initial property that any rationally updatable DL should possess. For this reason, we believe that it is an important property that should be considered when determining the practical usefulness of a certain DL (whether currently under development or already proposed in the literature). It is part of our future work to determine how the property of AGM-compliance for DLs is related to other results related to the AGM theory, such as the various representation theorems, the supplementary postulates, the revision, update and erasure operators and their respective postulates, the Levi and Harper identities etc; see [Gärdenfors, 1992; Katsuno and Mendelzon, 1992] for details on the above results.

Acknowledgments

The authors would like to thank Thomas Studer for his example in section 3.1 which resolved a long-standing issue.
References


